Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

Solving constrained engineering design problems with multi-objective artificial algae algorithm

Çok amaçlı yapay alg algoritması ile kısıtlı mühendislik tasarım problemlerinin çözülmesi

Ahmet ÖZKIŞ[1*], *Ahmet BABALIK*[2]

[1]Department of Computer Forensics Engineering, Faculty of Engineering, Necmettin Erbakan University, Konya, Turkey.
ahmetozkis@gmail.com

[2]Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Konya Technical University, Konya, Turkey.
ababalik@ktun.edu.tr

Received/Geliş Tarihi: 10.02.2022 Revision/Düzeltme Tarihi: 28.06.2022 doi: 10.5505/pajes.2022.88646
Accepted/Kabul Tarihi: 30.06.2022 Research Article/Araştırma Makalesi

Abstract

Engineering design problems fall into the category of problems that are very difficult to optimize. Nature-inspired metaheuristic techniques can be beneficial to solve such problems. In this study, a total of 14 different problems, 7 of which are benchmark problems and 7 of which are engineering design problems, were optimized using the recently proposed multi-objective artificial algae algorithm, MOAAA for short. For the performance test of the MOAAA, 4 different metrics named HV, SPREAD, EPSILON and IGD were used. Performance comparison was made with NSGA-II, PAES, MOCell, IBEA and MOVS algorithms which are well known in the literature. The Friedman test was applied to the metrics obtained for all algorithms and the average ranks of each algorithm were calculated. The results show that MOAAA has better performance than other algorithms in 3 of 4 metrics. In addition, the Wilcoxon's test reveals that the results obtained by the MOAAA are significant in the 95% confidence level.

Keywords: Artificial algae algorithm, Multi-objective constrained optimization, Metaheuristic algorithms, Multi-objective engineering design problems.

Öz

Mühendislik tasarım problemleri, optimize edilmesi oldukça zor olan problemler sınıfına girer. Doğadan ilham alan metasezgisel teknikler, bu tür problemleri çözmek için faydalı olabilmektedir. Bu çalışmada, yakın zamanda önerilen çok amaçlı yapay alg algoritması (MOAAA) kullanılarak 7 tanesi benchmark problemi, 7 tanesi mühendislik tasarım problemi olmak üzere toplam 14 farklı problem optimize edilmiştir. MOAAA'nın performans testi için, HV, SPREAD, EPSILON ve IGD olarak isimlendirilen 4 farklı metrik kullanılmıştır. Performans karşılaştırması literatürde iyi bilinen NSGA-II, PAES, MOCell, IBEA ve MOVS algoritmaları ile yapılmıştır. Tüm algoritmalar için elde edilen metriklere Friedman testi uygulanmış ve her algoritmanın ortalama başarı sırası hesaplanmıştır. Sonuçlar, MOAAA'nın 4 performans metriğinden 3'ünde diğer algoritmalardan daha iyi performansa sahip olduğunu göstermektedir. Ayrıca Wilcoxon testi, MOAAA ile elde edilen sonuçların %95 güven düzeyinde anlamlı olduğunu ortaya koymaktadır.

Anahtar kelimeler: Yapay alg algoritması, Çok amaçlı kısıtlı optimizasyon, Metasezgisel algoritmalar, Çok amaçlı mühendislik tasarım problemleri.

1 Introduction

Optimization problems with multiple objectives are called multi-objective optimization problems (MOOPs), and simultaneous optimization of these objectives is called multi-objective optimization (MOO). Real-world problems are generally in the type of NP-hard MOOPs. NP-hard means that it cannot be proven that there is a solution in polynomial time, or that the algorithms that can solve it efficiently are not known [1]; examples of these problems are found in engineering design, product and process design, land-use planning, management science, economics etc. In MOOPs, objective functions are generally inversely proportional. In other words, obtaining a satisfactory solution for an objective function result in a poor solution for the other objective function. Thus, it is not possible to obtain a global best solution as in single-objective problems, but instead, a solution set consisting of the best solutions is obtained.

Studies since the 1950s have shown that classical mathematical methods encounter various problems while solving MOOPs. These problems can be described as the failure of classical mathematical methods in problems where the Pareto front (PF) is concave or discrete, where difficulties are caused by the structure of classical mathematical methods and where the computational cost is high [2]. Thus, researchers favor multi-objective evolutionary algorithms (MOEAs) to solve MOOPs. MOEAs have a population consisting of multiple solutions, and all solutions work in cooperation using intelligent strategies. Obtaining successful results independent of the problem structure and much lower computational costs have encouraged researchers to develop new multi-objective evolutionary or heuristic algorithms. Day by day, new algorithms are added to well-known existing algorithms such as NSGA-II [3], MOPSO [4], SPEA2 [5], PAES [6], PESA [7], MOEA/D [8] and IBEA [9]. In parallel "survival of the fittest" theory of Darwin, the creation of numerous algorithms and the survival of the best of them is a natural process. In addition, "No-Free Lunch" theorem [10], which states the performance of optimization algorithms is similar on average when considering all possible types of problems, supports the creation of new algorithms. Some single-objective algorithms and corresponding multi-objective versions are given in Table 1, [11]:

*Corresponding author/Yazışılan Yazar

Table 1. Single-objective metaheuristic algorithms and multi-objective versions.

Single-objective metaheuristic algorithms	Multi-objective versions
Particle Swarm Optimization (PSO)	MOCLPSO [12], OMOPSO [13], SMPSO [14], MOPSO+LS [15], PO-MOPSO [16], MPSO/D [17], CEMOPSO [18]
Artificial Bee Colony (ABC)	VEABC [19]; MOABC [20]; HMOABC [21], A-MOABC/PD, A-MOABC/NS, S-MOABC/NS [22]
Ant Colony Optimization (ACO)	MOACOM [23], ACOAMO [24], SACO [25]
Grey Wolf Optimization (GWO)	MOGWO [26]
Ant Lion Optimization (ALO)	MOALO [27]
Cuckoo Search Algorithm (CSA)	MOCS [1]
Moth Flame optimization (MFO)	NS-MFO [11]
Water Cycle Algorithm (WCA)	MOWCA [28]
Simulated Annealing (SA)	OSA [29]
Artificial Immune Algorithm (AIA)	MOAIS [30], NNIA [31], HQIA [32]
Biogeography-based optimization(BBO)	MO-BBO [33]
Invasive Weed Optimization (IWA)	NSIWO [34]
Firefly algorithm (FA)	MOFA [35]
Bat Algorithm (BA)	MOBA [36]
Teaching-learning Based Optimization (TBLO)	MOTLBO [37]
Shuffled Frog Leaping Algorithm (SFLA)	MOSFLA [38], MOSG [39-41]
Vortex Search Algorithm (VS)	MOVS [42]
Artificial Algae Algorithm (AAA)	MOAAA [43]

Researchers continually propose new multi-objective metaheuristic algorithms. One of these is the recently proposed multi-objective artificial algae algorithm (MOAAA). MOAAA was recently tested on continuous and unconstrained multi-objective optimization problems, and it performed successfully [43]. This study aimed to test the performance of the MOAAA on constrained benchmark problems and engineering design problems.

Engineering design problems are one of the most important real-world problems, many of which are constrained problems [44]. Since optimal solutions of engineering design problems are difficult to find, metaheuristic algorithms are used in most studies [40],[45],[46].

The test set consists of 14 problems, two unconstrained and twelve constrained. The MOAAA was compared with NSGA-II, MOCell, MOVS, IBEA and PAES algorithms. The results obtained showed that the MOAAA performed better than the comparison algorithms on the used problem set.

The study is organized as follows. In section 2, MOOPs, the concept of Pareto and performance metrics are presented. In section 3, AAA is summarized. The MOAAA is explained in detail in section 4. Section 5 shows the results and performance analysis of the algorithms used on the problems. Finally, section 6 details conclusion and recommendations for further studies.

1.1 The major addition of the research

The MOAAA is a recently proposed technique for the solution of MOOPs. The MOAAA was first tested on unconstrained MOOPs and produced successful results. The performance of the MOAAA is aimed to investigate mainly on constrained MOOPs in this study. For this purpose, MOAAA has been run on 14 different MOOPs including engineering design problems and benchmark problems. The solutions obtained were evaluated according to 4 different performance metrics:

i. HV,
ii. SPREAD,
iii. EPSILON and
iv. IGD. The obtained metric results show that the MOAAA is generally superior to the comparison algorithms.

2 Multi-objective optimization, the pareto theorem and performance metrics

2.1 Multi-objective optimization problems

MOOPs are mathematically defined as follows. [28, 47]:

$$\begin{aligned} &Optimize\ F(X) = [f_1(X), f_2(X), \dots, f_M(X)]^T, \\ &subject\ to\ g_i(X) \le 0, i = 1,2, \dots, I \\ &h_j(X) = 0, j = 1,2, \dots, J \\ &X = [x_1, x_2, \dots, x_D] \\ &L_d \le x_d \le U_d, d = 1,2, \dots, D \end{aligned} \quad (1)$$

Where M is the number of functions, I is the number of inequality constraints, J is the number of equality constraints, and D is the number of decision variables. Also, g_i assigns the i^{th} inequality constraint, h_j assigns the j^{th} equality constraint, X is a candidate agents in the search area, L_d is lower bound and U_d is upper bound of the d^{th} decision variable.

2.2 Pareto theorem

In the MOOPs, since functions are generally inversely proportional to each other, numerous solutions that produce different values for different objective functions are formed. Researchers generally use the Pareto Theorem to determine the best ones by comparing these solutions. Where Q is the ideal set of solutions

-Those inside the search space that do not violate the problem constraints-and $A, B \in Q$, the Pareto Theorem consists of the following 4 rules [47]:

1. Pareto-dominance: If solution A is not poorer than solution B for any objective and is better at least in one objective, it dominates solution B and it is denoted as $A \prec B$. For a minimization problem, the mathematical notation of this rule is given in Equation (2).

$$\begin{aligned} &\forall i \in \{1,2,\dots,M\} f_i(A) \le f_i(B) \\ &\wedge \exists j \in \{1,2,\dots,M\} f_j(A) < f_j(B) \end{aligned} \quad (2)$$

If solutions A and B produce better values than each other in any objective, these are named as non-dominated solutions.

2. Pareto-optimal (PO): If no element in Q dominates solution A, it means that A is a PO solution.

$If\ \neg\exists \in Q: C \prec A,\ \ A$ is a PO solution.

3. Pareto-optimal-set (PS): A set of position vectors in the search area of PO solutions in Q.

$$\mathrm{PS} = \{\mathrm{A} \in Q | \neg\exists \mathrm{C} \in Q: \mathrm{C} \prec \mathrm{A}\} \quad (3)$$

4. Pareto-optimal-front (PF): A set of position vectors in the objective space of solutions in PS.

$$PF = \{F(A) | A \in PS\} \quad (4)$$

2.3 Performance metrics

In MOO, Pareto-optimal-fronts generated by the algorithms (Pareto front-estimated,PF_e) are expected to ideally estimate the true Pareto-optimal-fronts (Pareto-front-true, PF_t). Prediction success depends on two criteria:

i. Convergence: Convergence of PF_e to PF_t,
ii. Diversity: Distribution of PF_e over PF_t.

The quality of the algorithms was determined by comparing the PF_e solutions they produced. When the PF_e solutions produced by two different algorithms are similar, it is not possible to distinguish the better one by observation. Therefore, researchers have developed mathematical performance metrics that calculate convergence and diversity of the PF_e solutions. Some of these metrics calculate either convergence or diversity, while others calculate both. The inverted generational distance (IGD), Hypervolume (HV), EPSILON and SPREAD metrics used in this study are explained below:

- **HV** [47],[48]: Reference point W is found by taking the worst value in PF_t for every objective function. The volume HV_e is calculated by combining the closed space between reference point W and every solution in PF_e. Similarly, HV_t is calculated for PF_t and normalized HV (denoted as HV in the study) is obtained from the term HV_e/HV_t. The term HV should be close to 1, which means that the success of PF_e in estimating PF_t has increased. The HV metric is used to calculate convergence and diversity performance of a PF_e,
- **SPREAD** [48]: It evaluates the quality of distribution of the objective function vectors in PF_e. It is calculated by using the distance ($d(A, PF_e)$) of the vectors in PF_e to each other and to extreme solution vectors $(e_1, e_2, \dots, e_M)$ in PF_t. It is used to compute the diversity performance of a PF_e,
- **EPSILON**[48]: It evaluates the minimum distance which is needed for converting every solution in PF_e with a view to it is able to dominate the PF_t of the problem,
- **IGD** [48, 49]: It is used to measure the average from PF_t to PF_e.

The mathematical formulas of the performance metrics are given below [47],[48].

	Evaluated Criterion	Mathematical Formula	
HV	convergence, diversity	$\mathrm{HV}_e = \text{volume}\left(\bigcup_{i=1}^{\lvert PF_e\rvert} v_i\right)$, $\mathrm{HV}_t = \text{volume}\left(\bigcup_{i=1}^{\lvert PF_t\rvert} v_i\right)$, $\mathrm{HV} = \frac{\mathrm{HV}_e}{\mathrm{HV}_t}$	(5)
SPREAD	diversity	$\Delta = \frac{\sum_{m=1}^{M} d(e_m, PF_e) + \sum_{X\in PF_e} \lvert d(X, PF_e) - \bar{d}\rvert}{\sum_{m=1}^{M} d(e_m, PF_e) + \lvert PF_e\rvert \cdot \bar{d}}$ $d(X, PF_e) = \min_{Y\in PF_e, Y\neq X} \lVert F(X) - F(Y) \rVert^2$, $\bar{d} = \frac{1}{\lvert PF_t\rvert} \sum_{X\in PF_t} d(X, PF_e)$.	(6)
EPSILON	convergence	$E(PF_e, PF_t) = \inf_{\epsilon\in\mathbb{R}^+}\{\forall \vec{p} \in PF_t, \exists \vec{a} \in PF_e : \vec{a} \prec_\epsilon \vec{p}\}$,	(7)
IGD	convergence, diversity	$\mathrm{IGD}(PF_e, PF_t) = \frac{\sum_{v\in PF_t} d(v, PF_e)}{\lvert PF_t\rvert}$,	(8)

3 Artificial algae algorithm (AAA)

The AAA was inspired by the behaviors of the real algae, such as turning towards the source of light, growth by photosynthesis, reproduction by mitosis after reaching a sufficient size and adaptation to medium for survival. AAA was initially applied to solve single-objective problems and achieved quite successful results. In Figure 1, the main steps of the AAA are given; detailed information about the AAA can be obtained from [50].

Algorithm-1. Artificial Algae Algorithm

Initialization

1. *Define algorithm parameters (shear force (Δ) = 2, adaptation probability (Ap) = 0.5, loss of energy (e)=0.3)*
2. *Randomly distribute the algal colony cells(X_i) in search space, calculate the values of objective function ($f(X_i)$) and assign the initial size as 1. (G_i = 1), i = 1, 2,…,N*
3. *Update the sizes of algal colony cells $G_i = calculateGreatness(G_i, f(X_i))$*

Cycling

4. *Helical movement: Randomly choose 3 algal cells (k, l and m) and update positions*
$$X_{im}^{t+1} = X_{im}^t + (X_{jm}^t - X_{im}^t)(\Delta - \tau_i)p$$
$$X_{ik}^{t+1} = X_{ik}^t + (X_{jk}^t - X_{ik}^t)(\Delta - \tau_i)\cos\alpha$$
$$X_{il}^{t+1} = X_{il}^t + (X_{jl}^t - X_{il}^t)(\Delta - \tau_i)\sin\beta$$
X_i, X_j: algal colony cells; t: time
$$\alpha, \beta \in [0,2\pi];\ p \in [-1,1];\ \tau_i = 2\pi\left(\sqrt[3]{\frac{3G_i}{4\pi}}\right)^2$$
5. *Evolutionary process: Replicate a random cell chosen from the biggest algal colony to the smallest colony*
$$smallest_r^t = biggest_r^t$$
6. *Adaptation: Assimilate the starving algal colony into the biggest algal colony*
if *rand<A_p*
$$starving^{t+1} = starving^t + (biggest^t - starving^t) \times rand$$
end if

Termination

7. *Find the current best solution*

Figure 1. Main steps of the AAA.

4 Multi-objective artificial algae algorithm (MOAAA)

New multi-objective algorithms are proposed by rearranging metaheuristic algorithms that succeeded in the field of single-objective optimization using suitable strategies (Pareto-based, decomposition-based etc.). Unfortunately, implementation of these strategies is not sufficient for the success of the new algorithms because, while single-objective algorithms are intended to find a single point that produces the best solution, multi-objective algorithms are intended to find the best set of solutions (Pareto-front). Therefore, the ability of single-objective algorithms to distribute solutions needs to be improved. The arrangements for using the artificial algae algorithms in solving multi-objective problems are explained below.

4.1 Non-domination rank & Crowding-distance strategies

Non-domination rank (NDR): Non-dominated solutions were mentioned while explaining Pareto-dominance in section 2.2. When non-domination rank (NDR) strategy is used with NSGA-II [3], each set consisting of non-dominated solutions is indicated by a different number. The first front (FR1) in the population contains the best solution and is called the Pareto-front (PF). Selecting solutions with a small front number as the parent solution, or transferring them to the next generation, contributes to the convergence performance of the algorithm.

Crowding-distance (CRD): The NDR information is not sufficient to choose a solution from two coexisting solutions in the same front. Therefore, Deb et al. proposed the CRD strategy to determine which choice of solution will have a valuable contribution to the diversity. The CRD is calculated as follows:

i. The values obtained for each objective function by each solution on the same front are sorted in ascending order,
ii. The CRD values of the outermost solutions are assigned as infinite,
iii. The CRD values of the solutions in between are calculated for each objective function by normalizing the difference between the two closest neighboring solutions. The CRD value of a non-extreme solution, with the total of M objective functions, is calculated as in Eq. 9.

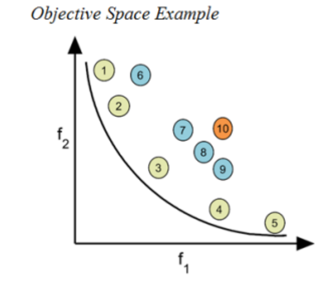
$$CRD_i = \sum_{m=1}^{M} \frac{f_m^{i+1} - f_m^{i-1}}{f_m^{\max} - f_m^{\min}} \tag{9}$$

In MOAAA, NDR and CRD are used in two cases:

i. When the parent algal colony cells are selected using binary tournament selection,
ii. When the best N of the 2N solutions-the main population (N) and the child solutions (N)-is transferred to the next generation at the end of each iteration.

4.2 Calculation of quality ranking (QR)

Real algae grow by photosynthesizing as they approach a light source. The algae that are closer to the light source will grow more because the rate of photosynthesis will increase. This is modeled in AAA as follows: the algal colony is initialized with the same sizes (Greatness) (Algorithm-1 Step 2), the sizes are increased at every iteration according to the values of Greatness and the objective function (Algorithm-1, Step 3). The values of the objective function are normalized in the calculateGreatness(G_i, $f(X_i)$) function. A scaler value representing the quality of the solutions is needed for the normalization, so using the objective vector in MOO is not applicable. Calculation of quality ranking (QR), which calculates the quality rankings of the solutions according to NDR and CRD values, is proposed to overcome this problem. In this calculation, QR values of the extreme solutions in the first front are assigned as 1, and other solutions are sorted in descending order according to CRD values and their QR values are increased by 1. The same procedure is repeated for all remaining solutions, where the QR value of the extreme solutions in the second front is 1 more than the highest QR value in the first front. Figure 2 gives an example of how the QR values are calculated.



Solution No	NDR	CRD	QR
1	1	inf	1
5	1	inf	1
3	1	0.9	2
4	1	0.7	3
2	1	0.65	4
6	2	inf	5
9	2	inf	5
7	2	0.61	6
8	2	0.12	7
10	3	inf	8

Figure 2. Calculation of QR value.

4.3 Polynomial mutation (PM)

In the original AAA, the evolutionary process and adaptation allow the failed algal colony cells to be influenced by the most successful colony cells to go to better locations. While the strategies comparing the failed solutions to the most successful one have a positive contribution to the convergence process, they have a negative effect on diversity performance in multi-objective optimization algorithms. Therefore, polynomial mutation [51], which contributes to the diversity, is added in MOAAA instead of the evolutionary process and adaptation. Main steps of the MOAAA is given in Figure 3.

Algorithm-2. Multi-objective Artificial Algae Algorithm

Initialization

1. *Define algorithm parameters (shear force (Δ) = 2, adaptation probability (Ap) = 0.5, loss of energy (e)=0.3)*
2. *Randomly distribute the algal colony cells(X_i) in search space, calculate the values of the objective function vectors ($F(X_i)$) and assign the initial size as 1. (G_i = 1), i = 1, 2,...,N*
3. *Update the sizes of algal colony cells G_i = calculateGreatness(G_i, $f(X_i)$)*
4. *Apply NDR & CRD strategies to algal colony cells*

Cycling

5. *Calculate G, QR, e and τ_i values of algal colony cells*
6. *Choose main (X_i) and neighbor solution (X_j) from the population using the binary tournament selection*
7. *Helical movement: Randomly choose 3 algae cells (k, l, m) and update positions*

$$offSpring_s = X_i^t$$
$$offSpring_{sm} = X_{im}^t + \left(X_{jm}^t - X_{im}^t\right)(\Delta - \tau_i)p$$
$$offSpring_{sk} = X_{ik}^t + \left(X_{jk}^t - X_{ik}^t\right)(\Delta - \tau_i)\cos\alpha$$
$$offSpring_{sl} = X_{il}^t + \left(X_{jl}^t - X_{il}^t\right)(\Delta - \tau_i)\sin\beta$$
X_i, X_j : algal colony cells; t: time
$$\alpha, \beta \in [0, 2\pi];\ p \in [-1,1];\ \tau_i = 2\pi\left(\sqrt[3]{\frac{3G_i}{4\pi}}\right)^2;\ s = 1, 2,...N$$

8. *Apply polynomial mutation to the produced $offSpring_s$ solution and calculate values of the objective function vector ($F(offSpring_s)$)*
9. *Merge algal colony population and offspring population (N+N = 2N)*
10. *Apply NDR & CRD to merged population and assign the best N solution as the algal colony population for the next iteration*

Termination

Find the PF solutions

Figure 3. Main steps of the MOAAA.

5 Results and performance analysis

5.1 The test set

The test set consists of 14 different multi-objective optimization problems: 7 benchmarks and 7 engineering designs. While twelve of these problems have various constraints, 2 of them are unconstrained problems. The number of objective, constraint and decision variables of the problem set are given in Table 2. The mathematical formulas of the problems are given in [1],[11],[27],[28],[52].

Table 2. Benchmarks and engineering desing problems.

	Problem	Dec. var. No	Obj. No	Constr. No
Benchmarks	Binh2	2	2	2
	ConstrEx	2	2	2
	Osyczka2	6	2	6
	Srinivas	2	2	2
	Tanaka	2	2	2
	KITA	4	2	3
	Water	3	5	7
Engineering design	Four-bar truss design	4	2	-
	Disk brake design	4	2	5
	Gear Train	4	2	-
	Spring	3	2	8
	Cantilever beam design	2	2	2
	Welded Beam	4	2	4
	Speed Reducer	7	2	10

5.2 Experimental environment

This study was carried out in the jMetal 4.5 environment, which is a multi-objective optimization software package coded in Java. While NSGA-II, PAES MOCell and IBEA algorithms used in the study were available in the jMetal package, MOVS and MOAAA were coded by the authors. Engineering design problems and the KITA problem were also coded by the authors and added to the package. Since the size of the problems was small (between 2 to 7 dimesions), the maximum function evaluation numbers (maxFES) of the algorithms were kept relatively low. All operations were repeated 50 times for 4000 maxFES. A parameter analysis study was also carried out in order to determine the optimal values of the K and le parameters used in the MOAAA for the problem set in this study. In the original AAA algorithm, parameter K was used as 2 and parameter le as 0.3. In this study, the K parameter was kept constant at 2, the le parameter was increased by 0.1 between 0.1 and 1, and 10 different MOAAA versions were run on the problem set with 50 repetitions. The results of the obtained solution sets in EPSILON metric were ranked by Friedman test. The obtained results showed that the parameter le obtained the most successful results for the value of 0.3. Secondly, the parameter le was kept constant at 0.3 and the parameter K was tested by increasing it from 1 to 5. Obtained results showed that K parameter gives the most successful results for 2 values. The population was taken as 100 for all algorithms; other parameters are set to default values as given in Table 3.

5.3 Experimental results

IGD, SPREAD, EPSILON and HV quality indicators were used to compare the performances of the algorithms.

Table 3. Parameter values of algorithms used in runs.

Algorithm	Parameters
NSGA-II	mutationProbability_ = 1/D crossoverProbability_ = 0.9 mutationDistributionIndex_ = 20.0 crossoverDistributionIndex_ = 20.0
PAES	archiveSize_ = 100 biSections_ = 5 mutationProbability_ = 1/D mutationDistributionIndex_ = 20
MOCell	archiveSize_ = 100 feedback_ = 20 mutationProbability_ = 1/D crossoverProbability_ = 0.9 mutationDistributionIndex_ = 20.0 crossoverDistributionIndex_ = 20.0
IBEA	archiveSize_ = 100 mutationProbability_ = 1/D crossoverProbability_ = 0.9 mutationDistributionIndex_ = 20.0 crossoverDistributionIndex_ = 20.0
MOVS	No specific parameter
MOAAA	mutationProbability_ = 1/D mutationDistributionIndex_ = 20.0 K = 2 le = 0.3

The Pareto-front-true (PFt) solutions for the problems were obtained to calculate these indicators. PFt solutions of the problems in the jMetal package were taken from the software website [53]. PFt solutions of the problems coded by the authors were obtained by merging the Pareto-front-estimated (PFe) solutions that were generated by solving each problem 50 times for all algorithms and separating the non-dominated solutions (maximum 500 solutions) within these. The values obtained by the algorithms for the four performance metrics are given in Tables 4 to 7. The values are expected to be high for HV and low for the others. For readability, the two best results are highlighted in dark gray (the best) and light gray (the second best).

The HV metrics in Table 4 show that the MOAAA takes first or second place in 8 (6+2) of the 14 problems. NSGA-II, MOCell, MOVS and IBEA take first or second place in 9 (3+6), 7 (3+4), 2 (1+1) and 2 (1+1) problems respectively.

The SPREAD metrics in Table 5 show that the MOCell is the most successful algorithm by taking first or second place in 13 (9+4) problems. The MOAAA, NSGA-II and PAES takes first or second place in 10 (4+6), 4 (1+3) and 1 (0+1) problems respectively.

The EPSILON metrics in Table 6 show that the MOAAA is the most successful algorithm by taking first or second place in 10 (6+4) problems. NSGA-II, MOCell, MOVS and IBEA take first or second place in 5 (3+2), 5 (3+2), 5 (1+4) and 3 (1+2) problems respectively.

The IGD metrics in Table 7 show that, the MOAAA is the most successful algorithm by taking first or second place in 9 (7 + 2) problems. NSGA-II, MOCell and MOVS takes first or second place in 8 (3+5), 5 (3+2) and 5 (1+4) problems respectively.

Table 4. HV metrics of the algorithms.

	NSGA-II		PAES		MOCell		IBEA		MOVS		MOAAA	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
Binh2	8.10e-01	2.8e-04	8.06e-01	7.9e-03	8.11e-01	8.4e-05	8.07e-01	1.9e-03	8.09e-01	3.5e-04	8.10e-01	2.2e-04
ConstrEx	7.73e-01	4.2e-04	7.30e-01	5.7e-02	7.71e-01	3.1e-03	7.57e-01	4.5e-03	7.70e-01	1.9e-03	7.70e-01	1.0e-03
Osyczka2	6.90e-01	4.6e-02	3.74e-01	1.1e-01	5.86e-01	1.4e-01	6.81e-01	5.0e-02	6.89e-01	1.7e-02	7.09e-01	1.3e-02
Srinivas	5.32e-01	3.3e-04	5.29e-01	8.9e-04	5.34e-01	1.1e-04	5.34e-01	1.5e-04	5.32e-01	4.2e-04	5.32e-01	3.2e-04
Tanaka	3.01e-01	1.5e-03	2.86e-01	1.3e-02	2.99e-01	4.3e-03	2.92e-01	2.5e-03	2.97e-01	3.3e-03	2.96e-01	2.3e-03
KITA	6.42e-01	1.8e-03	6.15e-01	2.3e-02	6.41e-01	2.6e-03	6.36e-01	2.1e-03	6.33e-01	3.4e-03	6.36e-01	2.4e-03
Water	3.93e-01	9.6e-03	2.81e-01	8.7e-02	4.01e-01	8.1e-03	2.70e-01	4.1e-02	4.08e-01	6.6e-03	3.97e-01	9.3e-03
FourBarTruss	7.11e-01	4.0e-04	6.80e-01	3.2e-02	7.09e-01	9.2e-03	7.13e-01	2.4e-03	7.11e-01	3.1e-04	7.12e-01	3.6e-04
DiscBrake	8.74e-01	9.3e-04	8.71e-01	1.3e-03	8.70e-01	6.0e-03	8.69e-01	4.0e-03	8.74e-01	1.0e-03	8.75e-01	8.9e-04
WeldedBeam	9.02e-01	1.4e-02	8.67e-01	4.8e-02	8.75e-01	3.5e-02	8.75e-01	1.3e-02	8.95e-01	9.2e-03	9.04e-01	4.9e-03
CantileverBeam	8.71e-01	1.8e-04	8.70e-01	2.2e-03	8.72e-01	2.9e-05	7.88e-01	3.2e-02	8.70e-01	1.9e-04	8.71e-01	1.9e-04
SpeedReducer	9.64e-01	2.7e-03	8.43e-01	8.3e-02	9.42e-01	3.8e-02	9.62e-01	2.4e-03	9.63e-01	2.5e-03	9.67e-01	6.7e-04
Spring	7.44e-01	9.6e-03	7.02e-01	2.8e-02	7.33e-01	1.8e-02	6.80e-01	4.3e-02	7.32e-01	1.8e-02	7.48e-01	5.0e-03
GearTrain	9.02e-01	6.3e-03	7.86e-01	1.5e-01	8.96e-01	1.7e-02	8.99e-01	8.3e-03	9.04e-01	1.4e-03	9.05e-01	2.5e-04
B+S	(3+6)		(0+0)		(3+4)		(1+1)		(1+1)		(6+2)	

B: Best. S: Second best.

Table 5. SPREAD metrics of the algorithms.

	NSGA-II		PAES		MOCell		IBEA		MOVS		MOAAA	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
Binh2	4.01e-01	3.2e-02	6.70e-01	5.5e-02	1.64e-01	1.1e-02	5.45e-01	1.2e-02	4.70e-01	3.8e-02	3.45e-01	3.0e-02
ConstrEx	4.68e-01	4.1e-02	8.73e-01	9.7e-02	3.84e-01	7.2e-02	1.12e+00	3.8e-02	5.78e-01	4.8e-02	4.86e-01	3.8e-02
Osyczka2	1.01e+00	9.9e-02	1.17e+00	1.2e-01	8.65e-01	1.0e-01	1.06e+00	5.6e-02	1.01e+00	8.6e-02	9.80e-01	1.0e-01
Srinivas	4.06e-01	3.0e-02	6.17e-01	4.5e-02	1.14e-01	1.4e-02	3.73e-01	2.4e-02	4.39e-01	4.1e-02	3.37e-01	2.8e-02
Tanaka	1.12e+00	6.3e-02	1.52e+00	6.1e-02	9.52e-01	6.7e-02	1.44e+00	6.8e-02	1.22e+00	5.6e-02	8.25e-01	5.2e-02
KITA	6.78e-01	1.6e-01	1.20e+00	9.8e-02	5.98e-01	1.2e-01	1.18e+00	6.4e-02	1.00e+00	9.2e-02	7.32e-01	1.5e-01
Water	5.65e-01	4.1e-02	6.38e-01	5.9e-02	5.56e-01	3.5e-02	7.33e-01	1.1e-01	5.76e-01	4.5e-02	5.31e-01	4.0e-02
FourBarTruss	3.87e-01	3.5e-02	7.81e-01	7.8e-02	2.27e-01	9.4e-02	4.47e-01	3.5e-02	4.69e-01	3.1e-02	4.00e-01	3.7e-02
DiscBrake	5.25e-01	8.1e-02	7.60e-01	5.4e-02	5.24e-01	6.7e-02	7.50e-01	3.2e-02	5.77e-01	4.0e-02	4.60e-01	4.3e-02
WeldedBeam	6.07e-01	7.2e-02	8.82e-01	5.3e-02	6.24e-01	1.5e-01	9.64e-01	3.8e-02	6.87e-01	3.8e-02	6.10e-01	5.5e-02
CantileverBeam	3.94e-01	3.0e-02	6.54e-01	5.1e-02	1.26e-01	1.8e-02	6.80e-01	4.1e-02	5.14e-01	3.6e-02	3.86e-01	2.9e-02
SpeedReducer	7.52e-01	1.0e-01	1.03e+00	7.6e-02	7.40e-01	7.7e-02	9.30e-01	3.8e-02	7.54e-01	5.8e-02	4.77e-01	5.0e-02
Spring	1.26e+00	6.0e-02	1.43e+00	7.0e-02	6.21e-01	7.5e-02	1.17e+00	1.2e-01	1.33e+00	1.9e-01	7.84e-01	1.4e-01
GearTrain	1.44e+00	5.5e-02	1.33e+00	1.3e-01	7.81e-01	3.9e-02	1.41e+00	8.9e-02	1.50e+00	2.9e-02	1.50e+00	2.9e-02
B+S	(1+3)		(0+1)		(9+4)		(0+0)		(0+0)		(4+6)	

B: Best. S: Second best.

Table 6. EPSILON metrics of the algorithms.

	NSGA-II		PAES		MOCell		IBEA		MOVS		MOAAA	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
Binh2	9.72e-01	1.8e-01	2.17e+00	2.9e+00	4.72e-01	5.0e-02	1.15e+00	3.4e-01	1.02e+00	1.7e-01	8.58e-01	1.4e-01
ConstrEx	1.59e-02	2.5e-03	8.59e-02	7.4e-02	2.13e-02	1.0e-02	4.34e-01	9.8e-02	2.11e-02	6.4e-03	1.98e-02	2.5e-03
Osyczka2	2.66e+01	1.7e+01	1.03e+02	4.1e+01	5.19e+01	4.0e+01	3.56e+01	1.8e+01	2.60e+01	9.3e+00	1.48e+01	6.3e+00
Srinivas	3.37e+00	5.9e-01	4.91e+00	1.3e+00	1.80e+00	2.6e-01	2.10e+00	2.2e-01	3.13e+00	5.6e-01	3.06e+00	6.2e-01
Tanaka	2.14e-02	4.5e-03	7.46e-02	7.6e-02	3.40e-02	1.7e-02	4.14e-02	6.6e-03	3.89e-02	1.4e-02	2.90e-02	7.4e-03
KITA	4.76e-02	9.1e-03	1.60e-01	6.9e-02	5.36e-02	1.3e-02	7.25e-02	8.8e-03	8.56e-02	2.1e-02	5.79e-02	7.1e-03
Water	7.92e+04	1.8e+04	1.16e+06	9.6e+05	5.81e+04	1.2e+04	1.17e+06	3.1e+05	6.39e+04	1.5e+04	7.49e+04	1.6e+04
FourBarTruss	1.04e+00	1.2e+00	5.81e+01	5.6e+01	1.67e+01	2.6e+01	4.10e+00	1.0e+01	1.13e-03	1.8e-04	1.14e-03	2.2e-04
DiscBrake	6.42e-02	1.3e-02	9.08e-02	3.0e-02	1.53e-01	1.1e-01	2.88e-01	1.0e-01	6.85e-02	1.7e-02	6.09e-02	1.1e-02
WeldedBeam	7.95e-01	8.1e-01	2.47e+00	2.1e+00	2.22e+00	1.5e+00	5.83e-01	4.5e-01	1.27e+00	6.1e-01	4.96e-01	4.4e-01
CantileverBeam	5.32e-04	4.5e-04	6.70e-03	2.3e-02	5.32e-04	4.9e-04	3.10e-04	9.1e-05	6.95e-04	7.2e-04	5.57e-04	5.2e-04
SpeedReducer	2.53e+01	1.8e+01	1.14e+02	4.9e+01	5.96e+01	6.8e+01	3.13e+01	2.4e+01	3.66e+01	1.6e+01	1.44e+01	2.6e+00
Spring	3.51e+03	4.3e+03	1.69e+04	1.1e+04	8.00e+03	7.8e+03	2.35e+04	1.2e+04	2.32e+03	3.4e+03	7.25e+02	6.3e+02
GearTrain	7.46e-01	8.4e-01	5.95e+00	6.2e+00	1.35e+00	1.3e+00	7.87e-01	9.8e-01	1.83e-02	2.6e-02	3.50e-03	5.5e-03
B+S	(3+2)		(0+0)		(3+2)		(1+2)		(1+4)		(6+4)	

B: Best. S: Second best.

Table 7. IGD metrics of the algorithms.

	NSGA-II		PAES		MOCell		IBEA		MOVS		MOAAA	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
Binh2	1.49e-04	7.6e-06	7.13e-04	7.7e-04	1.11e-04	2.5e-06	1.56e-03	3.5e-04	1.58e-04	8.0e-06	1.41e-04	6.8e-06
ConstrEx	2.02e-04	3.5e-05	2.37e-03	2.7e-03	3.31e-04	3.3e-04	4.87e-03	1.1e-03	2.60e-04	8.5e-05	2.63e-04	5.6e-05
Osyczka2	3.45e-03	2.1e-03	1.07e-02	2.8e-03	6.65e-03	3.0e-03	5.42e-03	1.9e-03	2.86e-03	1.5e-03	1.59e-03	8.4e-04
Srinivas	1.45e-04	7.9e-06	2.00e-04	2.7e-05	1.03e-04	3.4e-06	1.26e-04	7.0e-06	1.48e-04	1.0e-05	1.35e-04	7.9e-06
Tanaka	1.02e-03	1.8e-04	3.23e-03	2.7e-03	1.36e-03	5.7e-04	5.46e-03	4.2e-04	1.52e-03	4.2e-04	1.22e-03	1.9e-04
KITA	5.56e-04	8.6e-05	2.12e-03	1.8e-03	6.28e-04	1.6e-04	9.38e-04	1.3e-04	9.19e-04	2.1e-04	7.08e-04	7.8e-05
Water	2.48e-03	1.2e-04	1.04e-02	3.7e-03	2.33e-03	1.3e-04	1.50e-02	1.5e-03	2.30e-03	1.2e-04	2.48e-03	2.0e-04
FourBarTruss	2.86e-04	1.9e-05	4.56e-03	3.8e-03	1.10e-03	1.6e-03	5.06e-04	5.9e-04	2.99e-04	1.2e-05	2.83e-04	1.9e-05
DiscBrake	8.37e-04	5.6e-04	2.42e-03	9.4e-04	1.94e-03	1.2e-03	4.97e-03	4.3e-04	8.50e-04	7.0e-04	3.80e-04	5.3e-05
WeldedBeam	1.86e-03	1.7e-03	4.53e-03	3.0e-03	4.44e-03	2.5e-03	1.02e-02	2.1e-03	2.82e-03	1.4e-03	1.27e-03	9.6e-04
CantileverBeam	2.99e-04	1.7e-05	6.42e-04	1.0e-03	2.24e-04	4.6e-06	1.15e-02	1.6e-03	3.28e-04	1.8e-05	3.02e-04	1.9e-05
SpeedReducer	3.30e-03	2.9e-03	7.19e-03	3.3e-03	8.20e-03	2.8e-03	1.07e-02	1.8e-04	8.84e-04	6.9e-04	2.72e-04	5.4e-05
Spring	3.48e-03	1.7e-03	9.51e-03	4.7e-03	5.20e-03	3.1e-03	1.24e-02	5.2e-03	4.28e-03	1.9e-03	2.50e-03	1.2e-03
GearTrain	1.32e-02	6.4e-03	3.53e-02	2.1e-02	1.87e-02	8.0e-03	3.46e-02	5.4e-03	9.51e-03	2.6e-03	9.34e-03	2.7e-03
B+İ	(3+5)		(0+0)		(3+2)		(0+1)		(1+4)		(7+2)	

B: Best. S: Second best.

In Table 8, the Friedman test [54] results, which compare the average rankings of the algorithms, are given for each performance metric. In the Friedman test, it is desirable that the average ranking is high for HV and low for others. According to the results, the MOAAA has the best ranking for all metrics except the SPREAD metric. The MOAAA has the second-best ranking for the SPREAD metric. The NSGA-II has the second-best ranking of all the metrics, except for SPREAD. The MOCell has the best ranking for SPREAD.

Table 8. The average rankings of the algorithms for metrics.

	HV	SPREAD	EPSILON	IGD
NSGA-II	4.64	2.85	2.64	2.35
PAES	1.35	5.42	5.64	5.28
MOCell	3.85	1.42	3.42	3.21
IBEA	2.64	4.78	4.14	5.28
MOVS	3.71	4.35	3.21	2.99
MOAAA	4.78	2.14	1.92	1.85

The Wilcoxon's rank sum test was applied at a 95% confidence level to show whether the metric values obtained by the MOAAA are statistically different to the other algorithms or not. If the p (probability) value is smaller than 0.05, it is denoted by "+" and shows that MOAAA statistically differs from the other algorithms. The Wilcoxon rank sum test results in Tables 9 to 12 show that results produced by the MOAAA are statistically different compared to the other algorithms. Estimated Pareto fronts (PFe) obtained by the algorithms are compared with the real Pareto fronts (PFt) in Figure 4. When the figures are examined, it is observed that the proposed MOAAA generally estimates the PFt more successfully than the other algorithms. The box plot showing the results of the problems executed 50 times for each metric is given in Figure 5. When the box plot is examined, it is seen that MOAAA is a successful algorithm that produces robust results in solving the multi-objective optimization problems in the test set.

Table 9. Wilcoxon's rank sum test results for HV metric.

MOAAA vs	NSGA-II		PAES		MOCell		IBEA		MOVS	
	p-value	Sign	p-value	Sign	p-value	Sign	p-value	Sign	p-value	Sign
Binh2	1.17e-07	+	8.46e-18	+	7.07e-18	+	4.16e-12	+	4.73e-17	+
CantileverBeam	4.18e-01	-	2.33e-17	+	7.07e-18	+	7.07e-18	+	1.64e-14	+
ConstrEx	7.97e-18	+	4.00e-08	+	1.08e-06	+	7.50e-18	+	1.57e-01	-
DiscBrake	1.94e-04	+	1.13e-16	+	4.36e-09	+	1.05e-15	+	2.38e-07	+
FourBarTruss	2.02e-16	+	7.07e-18	+	1.12e-01	-	1.52e-11	+	2.65e-13	+
GearTrain	6.32e-01	-	8.47e-17	+	2.00e-02	+	1.43e-17	+	4.10e-05	+
SpeedReducer	4.73e-17	+	2.14e-16	+	6.69e-16	+	7.07e-18	+	2.62e-17	+
KITA	2.14e-16	+	9.54e-15	+	2.27e-13	+	7.59e-01	-	3.18e-05	+
Osyczka2	2.81e-05	+	7.07e-18	+	5.84e-12	+	1.58e-09	+	5.14e-09	+
Spring	1.20e-01	-	2.78e-17	+	1.77e-07	+	8.46e-18	+	5.56e-12	+
Srinivas	2.44e-11	+	7.07e-18	+	7.07e-18	+	7.07e-18	+	4.16e-12	+
Tanaka	7.92e-16	+	4.49e-08	+	9.33e-08	+	2.02e-11	+	1.30e-01	-
Water	9.59e-02	-	6.32e-16	+	3.58e-03	+	7.07e-18	+	1.12e-09	+
WeldedBeam	2.01e-01	-	3.16e-10	+	1.45e-09	+	4.46e-17	+	3.85e-08	+

Table 10. Wilcoxon's rank sum test results for SPREAD metric.

MOAAA vs	NSGA-II		PAES		MOCell		IBEA		MOVS	
	p-value	Sign	p-value	Sign	p-value	Sign	p-value	Sign	p-value	Sign
Binh2	7.80e-12	+	7.07e-18	+	7.07e-18	+	7.07e-18	+	8.46e-18	+
CantileverBeam	2.21e-01	-	7.07e-18	+	7.07e-18	+	7.07e-18	+	1.14e-17	+
ConstrEx	2.40e-02	+	7.07e-18	+	2.02e-10	+	7.07e-18	+	1.43e-13	+
DiscBrake	1.08e-08	+	7.07e-18	+	3.70e-08	+	7.07e-18	+	2.14e-16	+
FourBarTruss	7.93e-02	-	7.07e-18	+	2.79e-13	+	3.56e-08	+	2.39e-13	+
GearTrain	2.14e-06	+	7.61e-10	+	7.05e-18	+	1.11e-04	+	6.97e-01	-
SpeedReducer	6.72e-17	+	7.07e-18	+	9.54e-18	+	7.07e-18	+	7.07e-18	+
KITA	6.32e-02	-	1.21e-17	+	5.46e-06	+	7.50e-18	+	1.21e-12	+
Osyczka2	9.32e-02	-	7.09e-12	+	3.31e-07	+	1.30e-05	+	2.08e-01	-
Spring	5.64e-17	+	7.07e-18	+	1.23e-10	+	9.38e-16	+	8.00e-17	+
Srinivas	4.69e-15	+	7.07e-18	+	7.07e-18	+	1.32e-08	+	1.13e-16	+
Tanaka	7.07e-18	+	7.07e-18	+	5.88e-14	+	7.07e-18	+	7.07e-18	+
Water	1.70e-04	+	2.03e-14	+	2.56e-03	+	1.60e-16	+	3.21e-06	+
WeldedBeam	7.33e-01	-	7.07e-18	+	9.37e-01	-	7.07e-18	+	1.03e-10	+

Table 11. Wilcoxon's rank sum test results for EPSILON metric.

MOAAA vs	NSGA-II		PAES		MOCell		IBEA		MOVS	
	p-value	Sign	p-value	Sign	p-value	Sign	p-value	Sign	p-value	Sign
Binh2	6.69e-04	+	4.70e-11	+	7.07e-18	+	1.30e-05	+	3.67e-06	+
CantileverBeam	8.77e-01	-	2.56e-07	+	6.62e-01	-	7.47e-02	-	4.93e-01	-
ConstrEx	3.56e-11	+	1.76e-13	+	2.24e-01	-	7.07e-18	+	8.01e-01	-
DiscBrake	1.84e-01	-	3.39e-11	+	9.14e-09	+	7.07e-18	+	2.40e-02	+
FourBarTruss	7.06e-18	+	7.06e-18	+	7.06e-18	+	7.06e-18	+	6.97e-01	-
GearTrain	8.30e-06	+	1.18e-16	+	8.04e-10	+	8.58e-14	+	2.73e-05	+
SpeedReducer	1.34e-09	+	1.73e-17	+	9.64e-03	+	3.29e-02	+	1.54e-17	+
KITA	1.26e-08	+	7.07e-18	+	1.13e-02	+	7.44e-12	+	2.79e-13	+
Osyczka2	8.98e-08	+	1.01e-17	+	4.44e-15	+	1.73e-14	+	7.29e-10	+
Spring	2.70e-08	+	1.45e-17	+	6.88e-14	+	1.63e-17	+	9.68e-07	+
Srinivas	4.09e-03	+	5.29e-14	+	8.99e-17	+	2.69e-16	+	3.26e-01	-
Tanaka	5.59e-09	+	1.55e-12	+	6.87e-01	-	3.42e-12	+	2.07e-05	+
Water	2.90e-01	-	7.07e-18	+	3.69e-07	+	7.07e-18	+	6.52e-04	+
WeldedBeam	7.25e-02	-	2.53e-10	+	3.90e-11	+	3.23e-01	-	1.34e-09	+

Table 12. Wilcoxon's rank sum test results for IGD metric.

MOAAA vs	NSGA-II		PAES		MOCell		IBEA		MOVS	
	p-value	Sign	p-value	Sign	p-value	Sign	p-value	Sign	p-value	Sign
Binh2	6.62e-08	+	7.07e-18	+	7.07e-18	+	7.07e-18	+	9.54e-15	+
CantileverBeam	6.47e-01	-	2.07e-17	+	7.07e-18	+	7.07e-18	+	6.68e-10	+
ConstrEx	1.99e-12	+	1.58e-13	+	6.72e-02	-	7.07e-18	+	2.63e-01	-
DiscBrake	3.59e-12	+	7.07e-18	+	7.07e-18	+	7.07e-18	+	1.55e-15	+
FourBarTruss	2.57e-01	-	7.07e-18	+	1.67e-01	-	9.88e-10	+	2.76e-07	+
GearTrain	3.28e-04	+	2.79e-13	+	3.86e-12	+	7.02e-18	+	9.07e-01	-
SpeedReducer	1.91e-16	+	7.07e-18	+	7.07e-18	+	7.07e-18	+	2.95e-17	+
KITA	3.59e-12	+	2.02e-16	+	1.87e-06	+	3.19e-15	+	7.80e-12	+
Osyczka2	4.86e-08	+	7.97e-18	+	1.74e-15	+	2.56e-15	+	2.30e-08	+
Spring	3.16e-04	+	1.39e-15	+	9.52e-09	+	4.73e-17	+	1.49e-08	+
Srinivas	3.84e-09	+	7.07e-18	+	7.07e-18	+	4.12e-10	+	2.21e-10	+
Tanaka	4.67e-08	+	1.13e-16	+	8.77e-01	-	7.07e-18	+	1.08e-06	+
Water	4.54e-01	-	7.07e-18	+	5.11e-06	+	7.07e-18	+	7.42e-08	+
WeldedBeam	1.41e-01	-	9.81e-11	+	1.03e-10	+	7.50e-18	+	3.12e-09	+

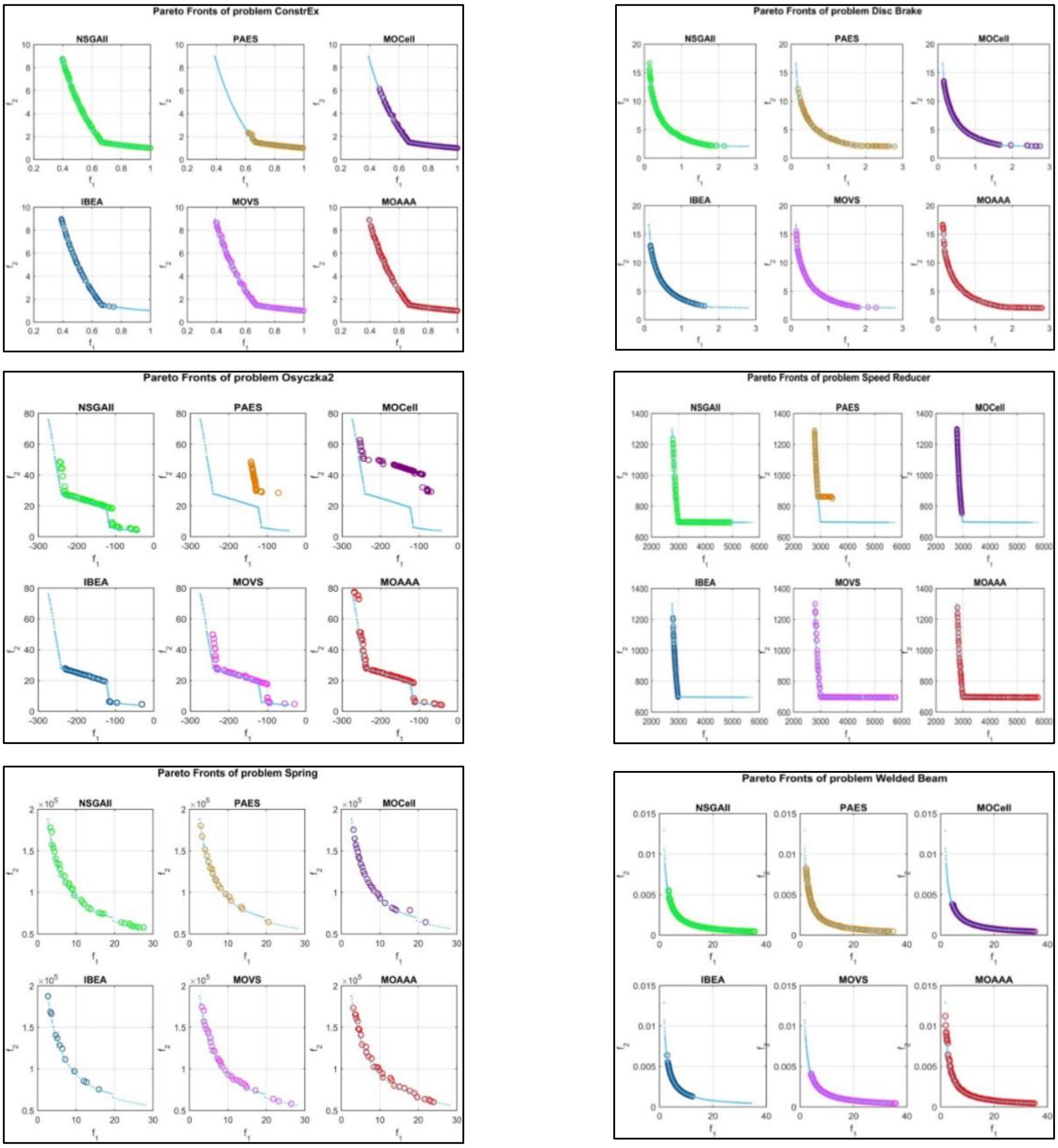


Figure 4. The Pareto fronts produced by algorihtms for the ConstrEx, Disc brake, Osyczka2, Speed reducer, Spring, Welded beam problems.

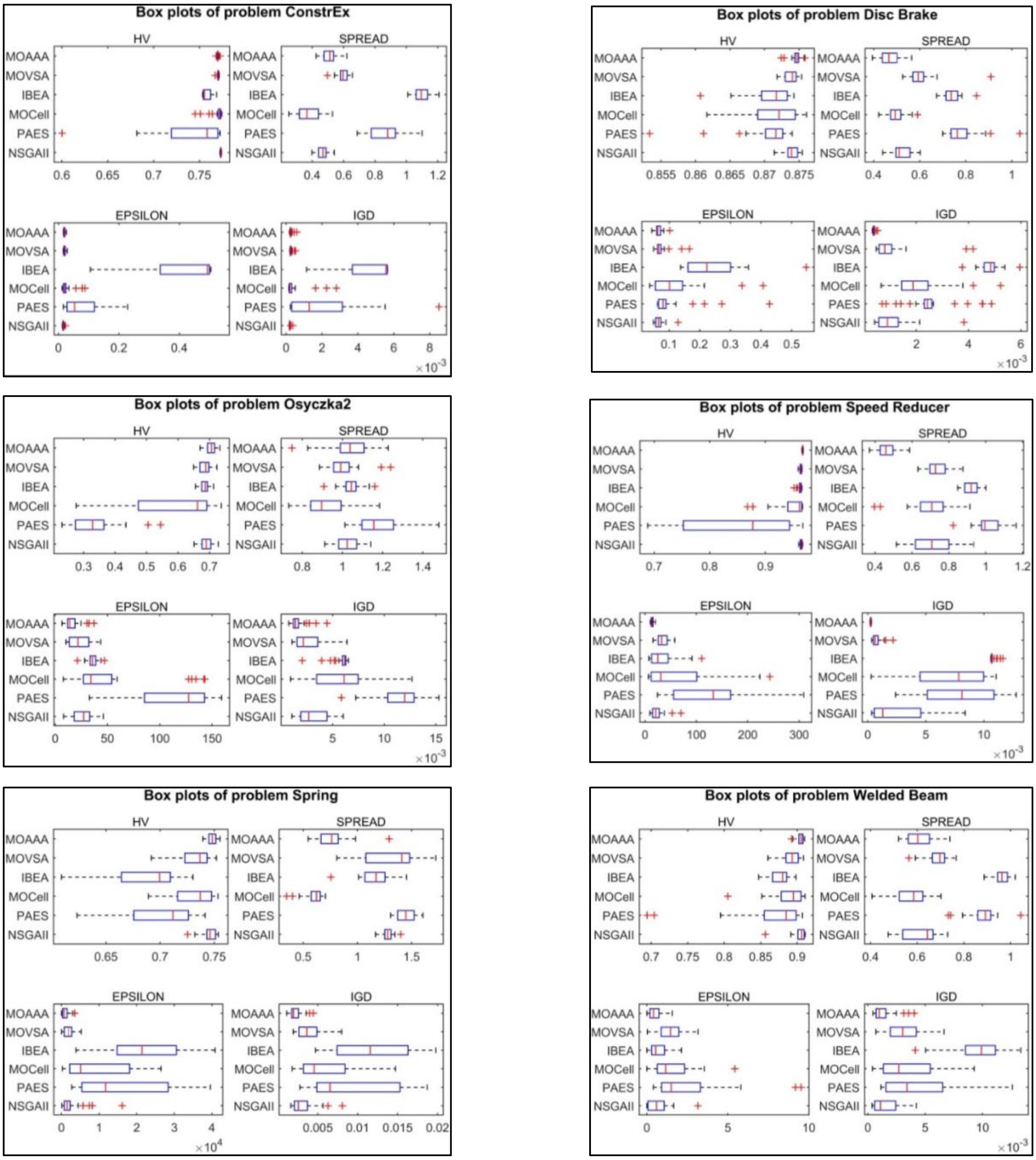


Figure 5. The box plots produced by algorihtms for ConstrEx, Disc brake, Osyczka2, Speed reducer, Spring, Welded beam problems.

6 Conclusions and recommendations

In this study, the performance of the MOAAA, a recently proposed multi-objective optimization algorithm, has been tested for constrained benchmarks and engineering design problems. The test set consisted of 14 well-known problems. The values obtained for HV, SPREAD, EPSILON and IGD from the MOAAA test set are compared with the well-known NSGA-II, PAES, MOCell, IBEA algorithms and the recently proposed MOVS algorithms. When the Friedman test, which compares the average rankings of the algorithms was applied, it was observed that MOAAA had the best ranking of all metrics, except for SPREAD. Furthermore, when the Pareto fronts and the boxplots were analyzed, it is seen that MOAAA was a consistent and stable algorithm that successfully estimated the Pareto fronts. Finally, the Wilcoxon rank sum test showed that MOAAA is a unique algorithm that produces statistically significant results different to the compared algorithms. The results show that MOAAA is an alternative method that generates successful results in solving real-world multi-objective problems.

In further studies, researchers could suggest modifications to enhance the distribution performance-SPREAD-of the MOAAA, or use MOAAA in solving discrete, dynamic or hybrid multi-objective real-world problems.

7 Author contribution statement

In this study, ÖZKIŞ focused on forming the idea, conducting experimental studies and evaluating the results; BABALIK, on the other hand, contributed to the review of the literature, spelling and checking the article in terms of content.

8 Ethics committee approval and conflict of interest statement

There is no need for an ethics committee approval in the prepared article. There is no conflict of interest with any person/institution in the prepared article.

9 References

[1] Yang XS, Deb S. "Multiobjective cuckoo search for design optimization". *Computers & Operations Research*, 40(6), 1616-1624, 2013.

[2] Yu CL, Lu Y, Chu J. "Multi-objective optimization with combination of particle swarm and extremal optimization for constrained engineering design". *WSEAS Transactions on Systems and Control*, 4(7), 129-138, 2012.

[3] Deb K, Pratap A, Agarwal S,Meyarivan T. "A fast and elitist multiobjective genetic algorithm: NSGA-II". *IEEE Transactions on Evolutionary Computation*, 6(2), 182-197, 2002.

[4] Coello CAC, Pulido GT, Lechuga MS. "Handling multiple objectives with particle swarm optimization". *IEEE Transactions on Evolutionary Computation*, 8(3), 256-279, 2004.

[5] Zitzler E, Laumanns M, Thiele L. "SPEA2: Improving the strength Pareto evolutionary algorithm". TIK-Report. Zurich, Switzerland, 103, 2001.

[6] Knowles J, Corne D. "The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation". *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99,* Washington, DC, USA, 6-9 July 1999.

[7] Corne DW, Knowles JD, Oates MJ. "*The Pareto envelope-based selection algorithm for multiobjective optimization"*. *Parallel Problem Solving from Nature PPSN VI: 6th International Conference,* Paris, France, 18-20 September 2000.

[8] Li H, Zhang Q. "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II". *IEEE Transactions on Evolutionary Computation*, 13(2), 284-302, 2009.

[9] Zitzler E, Künzli S. "*Indicator-based selection in multiobjective search"*. *Parallel Problem Solving from Nature PPSN VIII: 8th International Conference,* Birmingham, UK, 18-22 September 2004.

[10] Wolpert DH, Macready WG. "No free lunch theorems for optimization". *IEEE Transactions on Evolutionary Computation*, 1(1), 67-82, 1997.

[11] Savsani V, Tawhid MA. "Non-dominated sorting moth flame optimization (NS-MFO) for multi-objective problems". *Engineering Applications of Artificial Intelligence*, 63, 20-32, 2017.

[12] Huang VL, Suganthan PN, Liang JJ. "Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems". *International Journal of Intelligent Systems*, 21(2), 209-226, 2006.

[13] Sierra MR, Coello CC. "Improving PSO-based multi-objective optimization using crowding, mutation and e-dominance". *Third International Conference on Evolutionary Multi-Criterion Optimization, Guanajuato,* Mexico, 9-11 March 2005.

[14] Nebro AJ, Durillo JJ, Garcia-Nieto J, Coello CC, Luna F,Alba E. "Smpso: A new pso-based metaheuristic for multi-objective optimization". *Computational İntelligence in Miulti-Criteria Decision-Making (MCDM).* Nashville, TN, USA, 30 March 2009-02 April 2009.

[15] Moslehi G, Mahnam M. "A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search". *International Journal of Production Economics*, 129(1), 14-22, 2011.

[16] Wang Y, Yang Y. "Particle swarm optimization with preference order ranking for multi-objective optimization". *Information Sciences*, 179(12), 1944-1959, 2009.

[17] Dai C, Wang Y, Ye M. "A new multi-objective particle swarm optimization algorithm based on decomposition". *Information Sciences*, 325, 541-557, 2015.

[18] Zhang Y, Gong DW, Geng N. "Multi-objective optimization problems using cooperative evolvement particle swarm optimizer". *Journal of Computational and Theoretical Nanoscience*, 10(3), 655-663, 2013.

[19] Omkar S, Senthilnath J, Khandelwal R, Naik GN, Gopalakrishnan S. "Artificial Bee Colony (ABC) for multi-objective design optimization of composite structures". *Applied Soft Computing*, 11(1), 489-499, 2011.

[20] Akbari R, Hedayatzadeh R, Ziarati K,Hassanizadeh B. "A multi-objective artificial bee colony algorithm". *Swarm and Evolutionary Computation*, 2, 39-52, 2012.

[21] Zhang H, Zhu Y, Zou W,Yan X. "A hybrid multi-objective artificial bee colony algorithm for burdening optimization of copper strip production". *Applied Mathematical Modelling*, 36(6), 2578-2591, 2012.

[22] Akay B. "Synchronous and asynchronous Pareto-based multi-objective artificial bee colony algorithms". *Journal of Global Optimization*, 57, 415-445, 2013.

[23] Gravel M, Price WL, Gagné C. "Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic". *European Journal of Operational Research*, 143(1), 218-229, 2002.

[24] McMullen PR. "An ant colony optimization approach to addressing a JIT sequencing problem with multiple objectives". *Artificial Intelligence in Engineering*, 1 5(3), 309-317, 2001.

[25] T'kindt V, Monmarché N, Tercinet F, Laügt D. "An ant colony optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem". *European Journal of Operational Research*, 142(2), 250-257, 2002.

[26] Mirjalili S, Saremi S, Mirjalili SM, Coelho LdS. "Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization". *Expert Systems with Applications*, 47, 106-119, 2016.

[27] Mirjalili S, Jangir P, Saremi S. "Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems". *Applied Intelligence*, 46(1), 79-95, 2017.

[28] Sadollah A, Eskandar H, Kim JH. "Water cycle algorithm for solving constrained multi-objective optimization problems". *Applied Soft Computing*, 27, 279-298, 2015.

[29] Suman B, Hoda N, Jha S. "Orthogonal simulated annealing for multiobjective optimization". *Computers & Chemical Engineering*, 34(10), 1618-1631, 2010.

[30] Aydin I, Karakose M, Akin E. "A multi-objective artificial immune algorithm for parameter optimization in support vector machine". *Applied soft computing*, 11(1), 120-129, 2011.

[31] Gong M, Jiao L, Du H,Bo L. "Multiobjective immune algorithm with nondominated neighbor-based selection". *Evolutionary Computation*, 16(2), 225-255, 2008.

[32] Gao J, Wang J. "A hybrid quantum-inspired immune algorithm for multiobjective optimization". *Applied Mathematics and Computation*, 217(9), 4754-4770, 2011.

[33] Jamuna K, Swarup K. "Multi-objective biogeography based optimization for optimal PMU placement". *Applied Soft Computing*, 12(5), 1503-1510, 2012.

[34] Nikoofard AH, Hajimirsadeghi H, Rahimi-Kian A,Lucas C. "Multiobjective invasive weed optimization: Application to analysis of Pareto improvement models in electricity markets". *Applied Soft Computing*, 12(1), 100-112, 2012.

[35] Yang XS. "Multiobjective firefly algorithm for continuous optimization". *Engineering with Computers*, 29(2), 175-184, 2013.

[36] Yang XS. "Bat algorithm for multi-objective optimisation". *International Journal of Bio-Inspired Computation*, 3(5), 267-274, 2011.

[37] Krishnanand K, Panigrahi BK, Rout PK, Mohapatra A. "Application of multi-objective teaching-learning-based algorithm to an economic load dispatch problem with incommensurable objectives*". International Conference on Swarm, Evolutionary, and Memetic Computing*, Andhra Pradesh, India, 19-21 December 2011.

[38] Arshi SS, Zolfaghari A, Mirvakili SM. "A multi-objective shuffled frog leaping algorithm for in-core fuel management optimization". *Computer Physics Communications*, 185(10), 2622-2628, 2014.

[39] Karakoyun M, Gülcü Ş, Kodaz H. "D-MOSG: Discrete multi-objective shuffled gray wolf optimizer for multi-level image thresholding". *Engineering Science and Technology, an International Journal*, 24(6), 1455-1466, 2021.

[40] Karakoyun M, Kodaz H. "Çok amaçlı mühendislik tasarımı ve kısıtlı problemler için hibrit birçok amaçlı optimizasyon algoritması". *Mühendislik Bilimleri ve Tasarım Dergisi*, 9(4), 1200-1211, 2021.

[41] Karakoyun M, Ozkis A, Kodaz H. "A new algorithm based on gray wolf optimizer and shuffled frog leaping algorithm to solve the multi-objective optimization problems". *Applied Soft Computing*, 96, 1-26, 2020.

[42] Özkış A, Babalık A. "A novel metaheuristic for multi-objective optimization problems: The multi-objective vortex search algorithm". *Information Sciences*, 402, 124-148, 2017.

[43] Babalik A, Ozkis A, Uymaz SA, Kiran MS. "A multi-objective artificial algae algorithm". *Applied Soft Computing*, 68, 377-395, 2018.

[44] Augusto OB, Bennis F, Caro SJPO. "Multiobjective engineering design optimization problems: a sensitivity analysis approach". *Pesquisa Operacional*, 32, 575-596, 2012.

[45] Tawhid MA, Savsani VJAI. "A novel multi-objective optimization algorithm based on artificial algae for multi-objective engineering design problems". *Applied Intelligence*, 48, 3762-3781, 2018.

[46] Tawhid MA, Savsani VJNC, Applications. "Multi-objective sine-cosine algorithm (MO-SCA) for multi-objective engineering design problems". *Neural Computing and Applications*, 31, 915-929, 2019.

[47] Deb K. *Multi-Objective Optimization Using Evolutionary Algorithms.* 1st ed. Chichester, UK, Wiley, 2001.

[48] Durillo JJ, Nebro AJ. "jMetal: A Java framework for multi-objective optimization". *Advances in Engineering Software*, 42(10), 760-771, 2011.

[49] Zhang Q, Zhou A, Zhao S, Suganthan PN, Liu W,Tiwari S. "Multiobjective optimization test instances for the CEC 2009 special session and competition". University of Essex, Colchester, UK and Nanyang technological University, Singapore, Special Session On Performance Assessment Of Multi-Objective Optimization Algorithms, Technical Report, 264, 2008.

[50] Uymaz SA, Tezel G, Yel E. "Artificial algae algorithm (AAA) for nonlinear global optimization". *Applied Soft Computing*, 31, 153-171, 2015.

[51] Agrawal RB, Deb K, Agrawal R. "Simulated binary crossover for continuous search space". *Complex Systems*, 9(2), 115-148, 1995.

[52] Tawhid MA, Savsani V. "∊-constraint heat transfer search (∊-HTS) algorithm for solving multi-objective engineering design problems". *Journal of Computational Design and Engineering*, 5(1), 104-119, 2018.

[53] Kashan MH, Nahavandi N, Kashan AH. "DisABC: a new artificial bee colony algorithm for binary optimization". *Applied Soft Computing*, 12(1), 342-352, 2012.

[54] Derrac J, García S, Molina D,Herrera F. "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms". *Swarm and Evolutionary Computation*, 1(1), 3-18, 2011.