



Research paper

Sequential Multi-objective Genetic Algorithm

Leila Falahiazar¹, Vahid Seydi^{2*} and Mitra Mirzarezaee¹

1. Department of Electrical and Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.

2. Department of Computer Engineering, South Tehran Branch, Islamic Azad University Tehran, Iran.

Article Info

Article History:

Received 28 April 2020

Revised 30 December 2020

Accepted 01 March 2021

DOI:10.22044/jadm.2021.9598.2092

Keywords:

Multi-objective Evolutionary Algorithms, Non-dominated Sorting Genetic Algorithm II, Sequential Multi-objective Algorithm, Benchmark Functions, T-test.

*Corresponding author:
v_seydi@azad.ac.ir (V. Seydi).

Abstract

Many real-world issues have multiple conflicting objectives, and optimization of the contradictory objectives is very difficult. In the recent years, the Multi-objective Evolutionary Algorithms (MOEAs) have shown a great performance in order to optimize such problems. Thus the development of MOEAs will always lead to the advancement of science. The Non-dominated Sorting Genetic Algorithm II (NSGAI) is considered as one of the most used evolutionary algorithms, and many MOEAs such as the Sequential Multi-Objective Algorithm (SEQ-MOGA) have emerged to resolve the NSGAI problems. SEQ-MOGA presents a new survival selection that arranges the individuals systematically, and the chromosomes can cover the entire Pareto Front region. In this work, the Archive Sequential Multi-Objective Algorithm (ASMOGA) is proposed in order to develop and improve SEQ-MOGA. ASMOGA uses the archive technique in order to save the history of the search procedure so that the maintenance of the diversity in the decision space is adequately satisfied. In order to demonstrate the performance of ASMOGA, it is used and compared with several state-of-the-art MOEAs for optimizing the benchmark functions and designing the I-Beam problem. The optimization results are evaluated by the performance metrics such as the hyper-volume, generational distance, spacing, and t-test (a statistical test). Based on the results obtained, the superiority of the proposed algorithm is clearly identified.

1. Introduction

Finding the optimal solutions for a problem with several conflicting objects (goals) is a difficult and complex issue [1-5]. For example, the design of centrifuge engines, economic problems, and many engineering designs are considered as a multi-objective problem. In the last decade, the use of Multi-objective Evolutionary Algorithms (MOEAs) has been enormous to optimize the real-world problems [6], which indicates the robustness and applicability of MOEAs. In many different fields in the world of science, or most cases, we are faced with multi-objective situations [7]. It is time-consuming to create a balance between the conflicting objectives through the trial-and-error methods and numerous

experiments [8]. Therefore, the MOEA development will provide higher-performance solutions for the world of science. An MOEA optimizes a Pareto-optimal solution set. The performance of MOEA is: (i) a set of optimized solution converge to the Pareto Front, and (ii) optimize the solution covering the entire Pareto Front region, and the diversity is satisfied [9], and the performance of an MOEA is evaluated by the two mentioned reasons. The most important and commonly used algorithm for solving scientific problems is the Non-Dominated Sorting Genetic Algorithm II (NSGAI) [10]. Many researchers have solved the problems and upgraded this algorithm [1, 11]. NSGAI first corrected all the

problems with the predecessor algorithms [12, 13]. It is based on elitism and the Genetic Algorithm (GA) [14]; the only feature of this algorithm is to provide the crowding distance (a survival selection operator), and the perturbation in NSGAI is in a large number of generations, individuals trend to the boundary solutions. In MOEAs, other evolutionary optimization methods such as Particle Swarm Optimization (PSO) instead of GA gradually were used [13, 15, 16]. In fact, these algorithms provide different methods for exploration (or mutation) and exploitation (or crossover) operators only. However, the survival selection is significant in an MOEA to converge close to the Pareto Front, which must carefully select individuals for the next generation from the offspring and parent community.

The accuracy in choosing the next generation should be made in such a way that the diversity in the next generation is preserved well, and consequently, the exploration (mutation) and exploitation (cross-over) operations are not interrupted [17]. Many MOEAs such as NSGAI, over a few generations, obtain solutions, and these solutions (individuals) are non-dominated with each other. After that, this subset converges slowly to the Pareto Front. Indeed, the subject mentioned above is a problem in MOEAs. When an MOEA is carried out with several generations, all individuals occur in the non-dominant class or higher ranked class, and convergence takes place slowly from this point onwards. This problem is solved with Sequential Multi-Objective Algorithm (SEQ-MOGA) [18]. SEQ-MOGA presents a new selection operator that is called the Sequential Distance (SD); we will explain SD later. In this paper, the Archive Sequential Multi-Objective Algorithm (ASMOGA) is proposed. We try to use the archive concept in ASMOGA in order to improve the efficiency of SEQ-MOGA. The archive concept was first used by Knowles [19] in a multi-objective evolutionary algorithm. In the recent years, the archive concept in the literature has been very much considered to improve the performance of MOEAs [3, 20-25], and good results have been obtained by using it. The use of the archive causes the maintenance of diversity in the decision space, and the history of the search procedure is saved. The following subjects are clarified in the rest of this paper. We will briefly discuss about the well-known and state-of-the-art algorithms such as the non-dominant sorting particle swarm optimizer (NSPSO), Adaptive

Weighted Particle Swarm Optimization (AWPSO), Fuzzy Multi-objective Particle Swarm Optimization (FOMOPSO), Non-dominated Sorting Genetic Algorithm II (NSGAI), and SEQ-MOGA (the basis of our proposed algorithm). The proposed algorithm, ASMOGA, will be fully expressed in Section 2. In Section 3, ASMOGA is compared with the other algorithms in order to optimize the test functions such as the ZDT1, ZDT2, ZDT3, and ZDT4 functions [20], and the I-Beam as a real problem [11, 26]. The performance metrics and the t-test (as a statistical test) are used to examine and to compare the efficiency of ASMOGA with other well-known algorithms. Finally, the conclusions of this work are drawn in Section 4.

1.2. Related Literature Review

In NSGAI [10], the children are first produced by the standard bimodal cross-over operator and the polynomial mutation operators. Then children and the parents are merged in a set, and this set is ranked according to the concept of dominance and divided into different sets. A new population is selected from the ranking sets. However, if the size of the last set that wants to be added to the end of the new population is greater than the size of the rest of the new population, then by the Crowding Distance (CD) operator, the individuals of this set will be assigned the fitness value, and then according to the fitness assigned to each chromosome, the set is sorted in a descending order.

Finally, this sorted set is added to the end of the new population, respectively. AWPSO [16] is a multi-objective version of Particle Swarm Optimization (PSO) [27] that has the entire process of pure PSO but uses a dynamic formula to determine the fitness in a multi-objective solution space; for the next generation, it does not have enough diversity in the optimized solutions, and there is no innovation in the concept of domination. The Non-dominated Sorting Particle Swarm Optimizer (NSPSO) [28] uses all concepts of PSO, and all steps of NSGAI with the difference that the personal best of each individual is parent, and the offspring is a new movement of the parent. Also for assigning the fitness value, the Niche Count [29] and CD are used, and both of them are not significantly different to improve the diversity. The structure of the improved non-dominant sorting algorithm [30] uses a new diversity operator to select the new population. In

the Pareto archived evolution strategy [19], the evolutionary strategy (1 + 1) -ES has been combined with the archive concept to improve the convergence.

The Fuzzy Multi-objective Particle Swarm Optimization (FOMOPSO) [17] is an auto-tuning fuzzy controller, and has the process of PSO, and uses CD. It saves the non-dominated solution to the archive in each iteration for maintenance of the convergence. Sequential Multi-Objective Algorithm (SEQ-MOGA) [18] presents a new selection operator that is called the Sequential Distance (SD). In the next section, we will review a multi-objective problem, SEQ-MOGA, and the SD operator steps. After that, we will describe the proposed algorithm.

1.3. A Multi-objective Optimization Problem

A multi-objective optimization problem (MOP) has N variables and k objective functions; it can be defined as follows [8]:

\vec{X} is a vector with N components:
 $\vec{X} = \{X_1, X_2, \dots, X_N\}$, and \vec{X} is as a solution that we want to obtain the optimum values for the objective functions $f^i(\vec{X})$, $i = 1, 2, \dots, k$ under the following constraint $g^i(\vec{X})$ functions:

$$g^i(\vec{X}) \leq 0, \quad (1)$$

$$X_j^{\min} \leq X_j \leq X_j^{\max}, j = 1, 2, \dots, N$$

In MOPs, we wish to balance some entirely conflicting objects. MOPs can be solved using Evolutionary Algorithms (EAs), which means that EAs provide a bunch of solutions as optimized solutions for MOPs.

1.4. Sequential Multi-objective Genetic Algorithm

The Sequential Multi-Objective Genetic Algorithm (SEQ-MOGA) [18] is outlined in Algorithm 1. In step (7), the Sequential Distance (SD) is used for selecting the new population. In fact, after some generation in many MOEAs, all the optimization chromosomes are converged to the first class, and they are placed in the non-dominated set. Therefore, in all MOEAs, the determination of fitness has a great impact on the convergence of the algorithm.

The steps of SEQ-MOGA are explained in detail in Algorithm 1. In step (1), an initial population (P_{t_0}) of the individuals or chromosomes must be created, and these chromosomes present the

solutions of an MOP. In step (2), the off-springs are generated (C_t) via the mutation and the cross-over operators.

Algorithm 1. SEQ-MOGA.

-
- 1) Initialize a random population (P_{t_0}).
 - 2) Form new off-springs (C_t) through parents using mutation and crossover.
 - 3) Mix (P_t) with (C_t).
 - 4) Separate ($P_t + C_t$) into non-dominated sets.
 - 5) Add sub-sets to the new population until the last sub-set size is equal or more than the rest of the new population.
 - 6) If the rest of the new population is equal to the last subset size:
 go to step (8).
 else
 go on to the next step.
 - 7) Sort the last subset using the sequential distance algorithm. Add the sorted sub-set to the new population.
 - 8) If the end conditions are satisfied):
 go on.
 Else,
 go back to step (2).
 - 9) End.
-

In step (3), the parents and children are mixed and form a mixed population. In step (4), the individuals in the mixed population ($P_t + C_t$) are hierarchically categorized into the dominant subsets based on the concepts of dominance [8]. Each sub-set is called Pareto front (PF). In steps (5) and (6), the individuals are selected from the highest dominant sub-sets to constitute the new population. This process continues until a sub-set has a size greater than the remaining of the new population (in Figure 1, the size of the PF_3 is greater than the remaining of the new population). In step (7), the sequential distance (SD) algorithm is used to select the remaining of the new population from the individuals in the sub-set PF_3 . In SEQ-MOGA, the SD operator assigns the fitness to the individuals of a last set. In NSGAI, the crowding distance operator is used instead of the SD operator, and the steps of SEQ-MOGA are similar to NSGAI. However, in NSGAI, after a few generations, the two problems appear; 1) the convergence slows down, and 2) the individuals tend to the maximum and minimum values in the objective functions. These two problems have been solved in SEQ-MOGA by providing a SD operator. SD plays the role of a survival selection, and determines who to choose and stay for the next generation.

1.5. Sequential Distance (SD) Algorithm

In SEQ-MOGA [18], all the individuals are firstly divided into sub-sets or classes based on the non-dominated concepts (for example, Class 1 dominates all the other members of the population, and so on). These sub-sets are added to the new population, respectively and so long as a set is larger than the rest of the new population (see Algorithm 1). At this stage, SD determines which members of the last set will be selected for the next generation by assigning the fitness value. SD algorithm is described in Algorithm 2. In fact, SD tries to select the individuals that stay in a sequential and regular distance.

In step (1), SD sorts the sub-set according to one of the objective functions (f_m). For preserving the diversity, in step (2), SD assigns the most fitness to individuals that have f_m^{\min} and f_m^{\max} the zero value is allocated to the minimum and maximum of f_m , and this means that these two individuals get the best fitness (the best fitness value in SEQ-MOGA is zero). After that, the rest of the individuals in this sub-set will get their fitness in the following steps (3 to 5). In step (3), d_m^{mean} is calculated. d_m^{mean} means that the best diversity is obtained when all the individuals are placed among the maximum and the minimum for each objective function at a sequential distance, and this distance is d_m^{mean} .

Thus, we need to find the distance between the upper and lower values of each objective function (f_m), then divide this distance into the number of individuals. Thus d_m^{mean} is the ideal distance for each individual with its next and last neighbours in f_m .

As a result, the best position for an individual of its neighbours must be d_m^{mean} . Therefore, any individual that has a d_m^{mean} distance from its neighbours has a greater selecting chance for the next generation. In the light of the above considerations (in step 5), if the individual's neighbours have distances less than d_m^{mean} then a negative fitness is assigned to it (a bad diversity); but if an individual's neighbours have distances equal or more than d_m^{mean} then a zero or positive fitness is assigned to it (a good diversity).

Algorithm 2. SD algorithm.

-
- N is the number of individuals in this subset $n = 1, 2, \dots, N$.
 - M is the number of objectives.
 $m = 1, 2, \dots, M$.
 - 1. Sort the individuals based on an objective function f_m .
 - 2. The zero value is allocated to minimum and maximum of f_m .
 $fitness(1) = 0, fitness(n) = 0$
 - 3. Calculate the distance between f_m^{\min} and f_m^{\max} . Then divide it by the number of individuals.
For ($m = 1, 2, \dots, M$), do
$$d_m^{\text{mean}} = \frac{f_m^{\max} - f_m^{\min}}{N - 1}$$
 - 4. Calculate the distance of each individual from its neighbors.
For ($i = 2, \dots, N-1$), do
{
for ($m = 1, 2, \dots, M$), do
$$d_m^{\text{distance}}(i) = \frac{(|f_m^i(n) - f_m^i(n-1)| + |f_m^i(n+1) - f_m^i(n)|)}{2}$$

}
 - 5. Calculate the fitness of the individuals.
$$fitness(i) = \sum_{m=1}^M (d_m^{\text{distance}}(i) - d_m^{\text{mean}})$$
 - 6. Sort the array in this manner: select the individuals that have a positive or zero fitness value, and sort ascending them and put them in front of the array (Note that the individuals that have the maximum and the minimum objective functions are transmitted to the beginning of this array); then select the individuals that have a negative fitness, sort descending them, and put them in the rest of the array. Finally, this array will be added to the rest of the new population.
-

In Algorithm 2 and in step (5), the expression $d_m^{\text{distance}}(i) - d_m^{\text{mean}}$ is interpreted into three modes:

1. $d_m^{\text{distance}}(i) > d_m^{\text{mean}}$: the individual is at a good distance than its neighbours (far from its neighbours). Thus a positive fitness is assigned to it because we have a good diversity in objective space.
2. $d_m^{\text{distance}}(i) < d_m^{\text{mean}}$: the individual is at a worth distance from its neighbours (close to its neighbours). Thus a negative and worse fitness is assigned to it because we have a bad diversity in the objective space.
3. $d_m^{\text{distance}}(i) = d_m^{\text{mean}}$: the individual is exactly at a better distance from its neighbours (the distance from its neighbours is equal to d_m^{mean}). Thus a zero-fitness value is assigned to it because

we have the best diversity in the objective space.

In step (6), the sub-set sorts in the following manner, and it adds to the rest of the new population. At first, the individuals that have a zero fitness are added as the supremum points to the beginning of this array to maintain the diversity of the Pareto Front, and then the individuals that have a positive fitness are added in an ascending order to the array, and then the individuals that have a negative fitness are added in a descending order to the continuation of the array. Thus a better selecting chance in the next generation is for the individuals that follow the order of a sequential distance or d_m^{mean} of all the objective functions in the population. On the other hand, we have a good diversity using the SD operator.

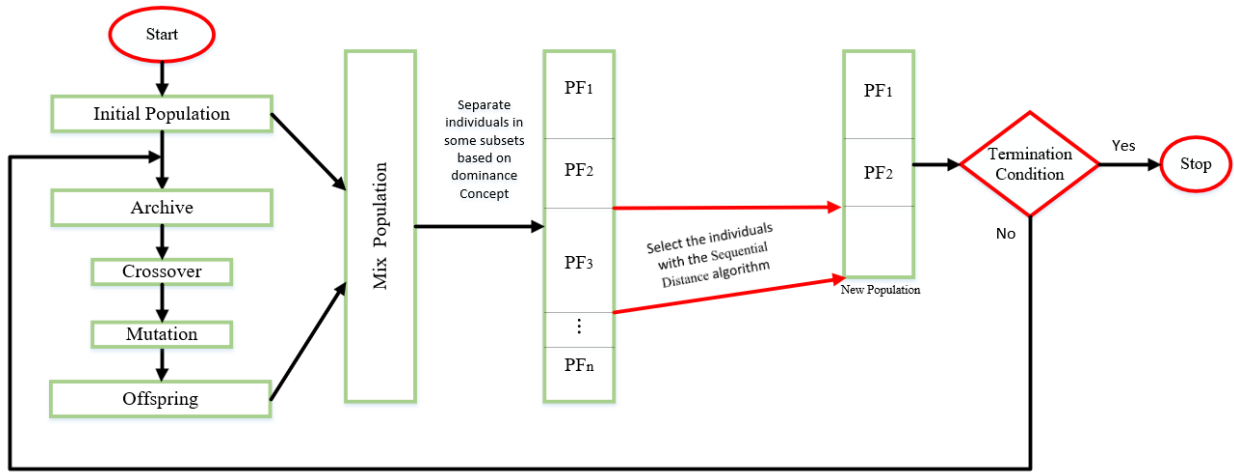


Figure 1. Proposed ASMOGA procedure.

The archive is a reserve non-dominated population. Initially, this archive is empty, and at each generation, the non-dominated solutions are selected, and added to the archive, and the archive is updated.

As you can see in Algorithm 3, the differences in SEQ-MOGA are only in steps (1) and (8). In step (8), new off-springs are compared with those in the archive, and if a child can dominate one of the members of the archive, the new child will be placed instead of a dominated individual inside the archive. As a result, we will have a memory of the best individuals in each generation using the archive. Subsequently, ASMOGA will obtain the Pareto Front area, and converge to it quickly.

2. Archive Sequential Multi-objective Algorithm (ASMOGA)

The local search approaches such as local Hill-Climbing, Tabu Search, Simulated Annealing, and archive concept are used in multi-objective to develop MOEAs [19], and these developing in MOEAs have better results than the population-based search methods. In this work, in order to increase the performance of Sequential Multi-Objective Algorithm (SEQ-MOGA) [18], we added the concept of the archive to the steps of SEQ-MOGA, and we propose the Archive Sequential Multi-Objective Algorithm (ASMOGA), which results in the simulation section showing that ASMOGA is more powerful than the other MOEAs. ASMOGA is presented in Algorithm 3. The steps of the ASMOGA process are presented in detail in Figure 1.

Algorithm 3. ASMOGA.

- 1) Initialize the random population (P_{t_0}) and the archive population (A_{t_0}).
- 2) Form new off-springs (C_t) through parents using mutation and crossover.
- 3) Mix (P_t) with (C_t).
- 4) Separate ($P_t + C_t$) into non-dominated sets.
- 5) Add sub sets to the new population until the last sub-set size is equal or more than the rest of the new population.
- 6) If the rest of the new population is equal to the last subset size:
 go to step (8).
 else,
 go on to the next step.
- 7) Sort the last sub-set using the SD algorithm. Add the sorted sub-set to the new population.
- 8) Update the archive population (A_t).
- 9) If the end conditions are satisfied,
 go on.
 else,
 go back to step (2).
- 10) End.

In step (7), the SD algorithm such a survival

selection operator is used, and SD is described in the previous section. The ending condition for this algorithm (like other MOEAs) is the generation counter. In the following, we will use the Archive Sequential Multi-Objective Algorithm (ASMOGA) to optimize a real problem and the test functions.

3. Simulations

The parameters and assigned values of Table 1 were used in the simulations stages of the ZDT1 to ZDT4 test functions and the design of an I-Beam.

3.1. Optimization of Test Functions

In this section, the Archive Sequential Multi-Objective Algorithm (ASMOGA) is compared with NSGAII, AWPSO, NSPSO, FOMOPSO, and

SEQ-MOGA. The ZDT test functions are presented in Table 2. In Figure 2, the solutions obtained from ASMOGA for optimizing ZDT1, ZDT2, ZDT3, and ZDT4 [8, 20] are shown in a single run. In all functions, the generation number is 200, and the population size is 100. According to the previous research works [8, 10], the chromosome dimension or length in ZDT1, ZDT2, and ZDT3 are usually 30 but for ZDT4 is 10. In this paper, the Rcross is the cross-over probability, and Rmute is the mutation probability; the Rcross and Rmute assigned values are 0.5. In the experimental optimization of the ZDT functions and design of I-Beam, the linear cross-over [31] and the uniform mutation [32] are used.

Table 1. Parameters and assigned values in the simulations.

Population size	100
Generation number	200
Independent runs	30 times
Chromosome dimension in ZDT1, ZDT2, and ZDT3	30
Chromosome dimension in ZDT4	10
Chromosome dimension or variables for design of an I-Beam	4
Rcross	0.5
Rmute	0.5
Worst objective function values in ZDT1, ZDT2, and ZDT3	$Max(f_1) = 1, Max(f_2) = 10$
Worst objective function values in ZDT4	$Max(f_1) = 1, Max(f_2) = 384.3$
Worst objective function values for design of an I-Beam	$Max(f_1) = 1000, Max(f_2) = 1$

Table 2. ZDT test functions [8].

$ZDT\ 1 = \begin{cases} f_1 = x_1 \\ f_2 = g(x)h(f_1, g(x)) \\ g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i \\ h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} \\ \{x_1, x_2, \dots, x_n\} \in [0, 1] \end{cases}$	$ZDT\ 3 = \begin{cases} f_1 = x_1 \\ f_2 = g(x)h(f_1, g(x)) \\ g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i \\ h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} - \left(\frac{f_1}{g}\right) \sin(10\pi f_1) \\ \{x_1, x_2, \dots, x_n\} \in [0, 1] \end{cases}$
$ZDT\ 2 = \begin{cases} f_1 = x_1 \\ f_2 = g(x)h(f_1, g(x)) \\ g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i \\ h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^2 \\ \{x_1, x_2, \dots, x_n\} \in [0, 1] \end{cases}$	$ZDT\ 4 = \begin{cases} f_1 = x_1 \\ f_2 = g(x)h(f_1, g(x)) \\ g(x) = 1 + 10(n-1) \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i)) \\ h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} \\ x_1 \in [0, 1], \{x_2, \dots, x_n\} \in [-5, 5] \end{cases}$

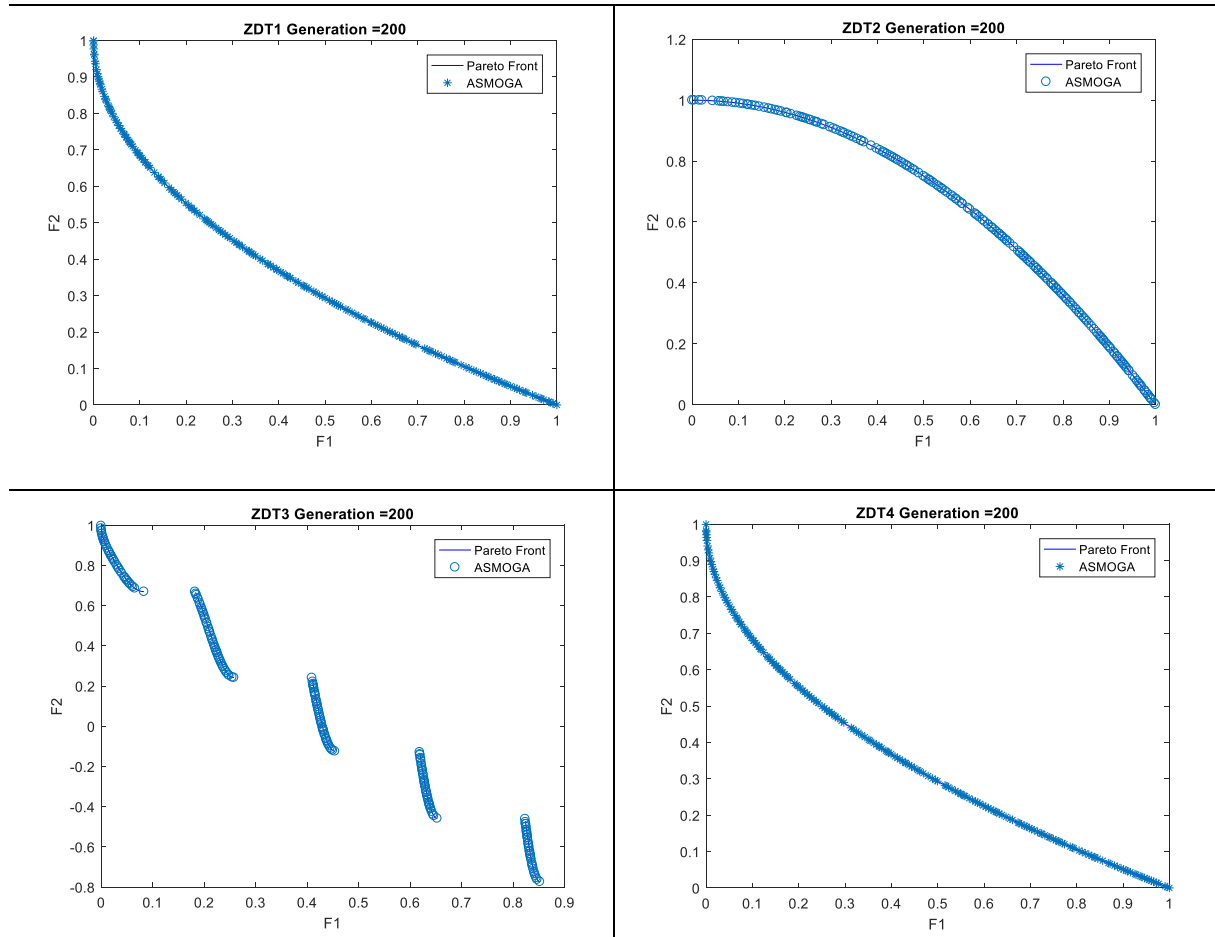


Figure 2. Optimization of the ZDT1 to ZDT4 functions performed using ASMOGA in a single run.

The general form of the linear cross-over is as follows:

$$C_1 = \lambda_1 P_1 + \lambda_2 P_2 \quad (2)$$

$$C_2 = \lambda_1 P_2 + \lambda_2 P_1 \quad (3)$$

C_1 and C_2 are the children of P_1 and P_2 , respectively. λ_1 and λ_2 are the combination coefficients that are randomly obtained by the following formula:

$$\lambda_1 + \lambda_2 \leq 2, \quad \lambda_1 > 0, \lambda_2 > 0 \quad (4)$$

The reason for using the linear cross-over is that this kind of cross-over covers the entire search space. Thus the exploration of the search space is perfectly achieved. Consequently, in the uniform mutation operator, only one gene is randomly selected in each chromosome, and then the gene is mutated within the specified range.

In order to compare the proposed algorithm with other methods, we used two performance metrics [8]. These metrics are Generational Distance (GD) [8, 33] and spread (Δ) [10], which are described below. The GD and spread metrics are used when the Pareto Front (P^*) is known. The average values and standard deviation from 30

independent runs of the test functions are considered in Tables 3 and 4. Table 3 shows the GD metric, and Table 4 shows the spread metric. In GD, the distance between the optimization solutions (Q) and the known solutions of Pareto Front (P^*) is measured. GD is defined as follows:

$$GD = \frac{(\sum_{i=1}^{|Q|} d_i^m)^{1/m}}{|Q|} \quad (5)$$

The number of objective functions is m , $|Q|$ is the number of non-dominated solutions obtained from MOEA, and d is the euclidean distance between Q and the nearest solution from P^* . The best value for GD is zero; this means that if the solutions obtained from the optimization algorithm are closer to the Pareto Front, then GD must tend to zero.

The spread metric [8, 33] measures the distance of the solutions or, in other words, determines the diversity of the non-dominated solutions. The ideal value for the spread is zero ($\Delta = 0$). The spread metric is formulated as follows:

$$f_2(\vec{X}) = \frac{PL^3}{4E(x_3(x_1 - 2x_4)^3 + 2x_2x_4(4x_4^2 + 3x_1(x_1 - 2x_4)))} \quad (8)$$

$$\Delta = \frac{\sum_{m=1}^M d_m^e + \sum_{i=1}^{|Q|} |d_i - \bar{d}|}{\sum_{m=1}^M d_m^e + |Q|\bar{d}} \quad (6)$$

where d_i is the distance between the optimization solutions with the neighbors in the Q set, \bar{d} is the mean value for all d_i , and d_m^e is the distance between the upper limit and the lower limit of the m th objective functions for P^* and Q . When the best distribution of Q is obtained (uniform distribution), $\Delta = 0$. As one can see in Tables 3 and 4, ASMOGA performs a better optimization in comparison with the other methods, and only the spread metric in ZDT3 has a negligible difference with SEQ-MOGA.

3.2. I-Beam Optimization

I-Beam is a very important structure for the building strength. In some papers, the strength of this structure has been increased [34-38], and in some of the references, the optimal design of this structure has been considered [11, 26, 39, 40]. In this work, we will optimize the design of the I-Beam. The physical characteristics in the design of the I-Beam are shown in Figure 3.

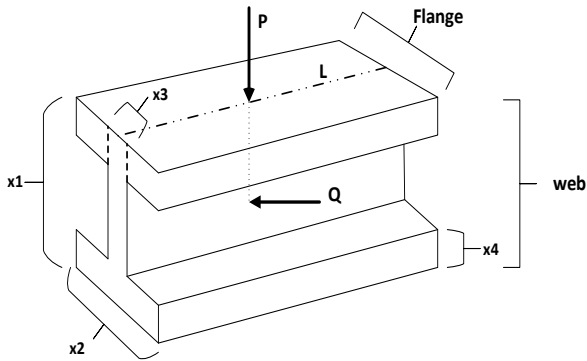


Figure 3. Physical structure of the I-Beam; the vertical (P) and horizontal (Q) loads are shown.

This engineering problem has two conflict objective functions. The first objective function is to reduce the amount of material (f_1), and the second objective function is to increase the strength of the I-Beam (f_2). On the other hand, if the material constituting (f_1) is minimized, then the strength and the bearing pressure is reduced (f_2). f_1 is in conflict with f_2 . The formula of f_1

and f_2 is as follows [11, 26]:

$$f_1(\vec{X}) = 2x_2x_4 + x_3(x_1 - 2x_4), \quad (7)$$

Also the constraints can be described as follows:

$$\frac{M_Y}{Z_Y} + \frac{M_Z}{Z_Z} \leq \sigma_b \quad (9)$$

In Equation (9), M_Y and M_Z are the maximum bending values in the Y and Z directions, respectively. σ_b is the permissible bending stress for the material. This equation can be re-written as follows:

$$\frac{180000x_1}{x_3(x_1 - 2x_4)^3 + 2x_2x_4[4x_4^2 + 3x_1(x_1 - 2x_4)]} + \frac{15000x_2}{(x_1 - 2x_4)x_3^3 + 2x_4x_2^3} \leq 16 \quad (10)$$

The geometric constraints in the I-Beam are as follows:

$$10 \leq x_1 \leq 80, 10 \leq x_2 \leq 50, \quad (11)$$

$$0.9 \leq x_3 \leq 5, 0.9 \leq x_4 \leq 5$$

The length of the I-Beam is $L = 200$ cm. $P = 600$ kN and $Q = 50$ kN are the external loads. The Young's modulus of elasticity are $\sigma_b = 16 \text{ kN/cm}^2$ and $E = 2 \times 10^4 \text{ kN/cm}^2$. In ASMOGA and other compared algorithms, for optimizing the I-Beam, the length of the representation chromosome is 4. This chromosome consists of four genes or four variables are used to design the I-Beam (Figure 4). In Figure 4, X_1 is the first variable that shows the I-Beam's height, the second variable or X_2 shows the I-Beam width, X_3 shows the web width, and X_4 represents the flange width.

X_1	X_2	X_3	X_4
-------	-------	-------	-------

Figure 4. A chromosome that is used to design an I-Beam and is used by the optimization algorithms.

X_i is the constructors of each objective function f_1 and f_2 for $i = 1, 2, 3, 4$. We used the t-test (a statistical test) in order to compare the proposed algorithm with the other methods. The t-test requires at least 30 data [41, 42]. Thus we run 30 times each algorithm separately, and used two performance metrics [8]. At this point, for the I-Beam to be a real problem and the exact Pareto Front not clearly exist, we must use the Hypervolume (HV) [40, 43] and spacing

performance metrics [44] to compare the performance of different algorithms, and two preceding performance metrics, GD, and spread are not applicable. HV is a perfect performance metric for comparing MOEAs. The HV formula is given below:

$$HV = volume(\bigcup_{i=1}^{|Q|} v_i) \quad (12)$$

Where v_i is the worst value of the i -th objective function, these worst values along with the obtained non-dominated solutions are used to form the corners of a multi-dimensional space. The worst values used in this work to calculate HV are given in Table 1. A larger value of HV shows the efficiency of the optimization algorithm. The spacing metric is calculated by the following formula:

$$S = \sqrt{\frac{1}{|Q|} \sum_{i=1}^{|Q|} (d_i - \bar{d})^2} \quad (13)$$

where i is the number of non-dominated solutions that are obtained from the optimization algorithm

and $d_i = \min_{z \in Q \wedge z \neq i} \sum_{m=1}^M |f_m^i - f_m^z|$, and

$\bar{d} = \sum_{i=1}^{|Q|} d_i / |Q|$, and M is the number of objective

functions. The ideal value for this metric is zero. The proposed algorithm, ASMOGA, in this work is compared with the other five mentioned algorithms previously by these two metrics for optimizing the I-Beam. As one can see, in Table 5, the lowest value of the spacing metric is obtained by ASMOGA. Also in Table 6, the largest value is obtained by ASMOGA (the proposed algorithm). Also a statistical test called the t-test [41, 42] was used to compare the results of the algorithms. In all simulations for optimization of the I-Beam, the constant values in Table 1 were used. Each algorithm was repeated in 30 independent runs, and then the results of these 30-times runs were given for the t-test. The t-test at least requires 30 statistical data to work correctly, and it is not necessary that the data has a normal or abnormal distribution. It is the main advantage of the t-test.

Thus in this research work, we run each algorithm for 30 times. The t-test was used to determine the significant difference between the two algorithms [45]. In the simulation of this research work, we used the ttest2 function in the MATLAB program. The results of the t-test for the hyper-volume and spacing metrics are presented in Tables 7 and 8 for optimizing the I-Beam. In the t-test, the

significant difference between the two algorithms appeared when the p-value was upper than 0.05, and H became 1. In Tables 7 and 8, ASMOGA shows the better p-values in comparison with all the other five algorithms, and the other algorithms were compared with each other (the blue highlighted cell). In Figure 5, the result of a single run of ASMOGA for the I-Beam optimization is shown.

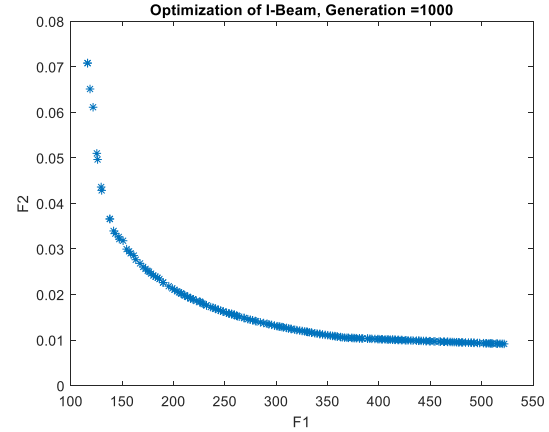


Figure 5. Result of a single run obtained by ASMOGA for the I-Beam optimization. The generation number is 1000.

The generation number is 1000, and all variables of the mutation and cross-over functions, and the performance metrics are represented in Table 1. In this figure, one can obviously see the convergence of the solutions to the Pareto Front.

In Figure 6, the result of a single run for all six compared algorithms is shown, the generation number is 200, and ASMOGA indicates the best Pareto Front for the I-Beam optimization.

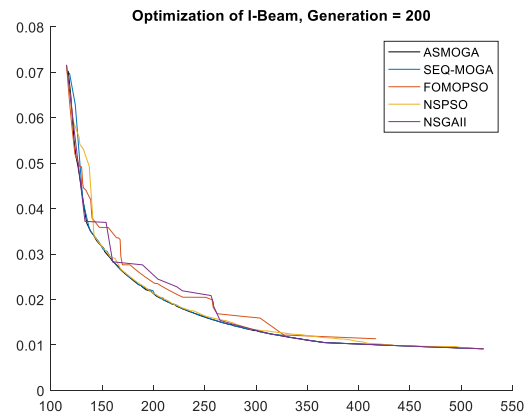


Figure 6. Optimization results of the I-Beam. problem after a single run for six compared algorithms.

In other papers, Hajela and Shih [26], Huang and Ying-Kui Gu [39], and Annamdas and Rao [11], they have worked on f_1 optimization, because these methods are based on single-objective optimization, and they have provided a

solution as an optimal solution.

Based on the results in Table 9, the superiority of ASMOGA (the proposed method in this article) is

clarified. You can see the best result in function f_1 according to Figure 5 and Table 9.

Table 3. The mean and variance values of the Generational Distance metric (GD) for 30 independent runs.

	ZDT1		ZDT2		ZDT3		ZDT4	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
NSGAII	8.94e-4	0	8.24e-4	0	4.34e-02	4.20e-5	2.92e-2	4.67e-2
NSPSO	7.53e-4	4.18e-5	8.05e-4	3.05e-5	3.4e-03	2.54e-4	7.82e-4	.91e-5
AWPSO	1.01e-4	2.61e-9	1.210e-4	1.4e-9	5.206e-4	2.85e-9	5.34e-4	3.92e-8
SEQ-MOGA	1.26e-04	1.07e-09	5.06e-05	7.98e-12	9.24e-05	1.72e-10	1.19e-04	1.34e-09
FOMOPSO	2.37e-03	2.43e-05	3.11e-04	1.03e-06	1.50e-03	2.85e-07	2.38 e-2	1.302 -03
ASMOGA	9.68e-05	1.80e-10	3.46e-05	9.81e-13	6.01e-05	3.11e-11	9.73e-05	1.23e-10

Table 4. The mean and variance values of the spread metric for 30 independent runs.

	ZDT1		ZDT2		ZDT3		ZDT4	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
NSGAII	0.463	4.16e-2	0.435	2.46e-2	0.576	5.08e-3	0.655	1.98e-1
NSPSO	0.767	3.00e-2	0.758	2.77e-2	0.869	5.81e-2	0.768	3.57e-2
AWPSO	0.759	1.71e-3	0.758	4.92e-3	0.891	1.10e-2	0.680	3.00e-3
SEQ-MOGA	0.389	5.78e-04	0.38	9.69e-04	0.57	3.90e-04	0.354	6.00e-04
FOMOPSO	0.7242	3.08 -03	0.7873	6.01e03	0.7942	1.46 e-03	1.1871	3.87 e-2
ASMOGA	0.387	3.81e-04	0.37	3.68e-04	0.58	2.59e-04	0.353	3.02e-04

Table 5. The mean and variance of the spacing metric for 30 independent runs for the I-Beam optimization.

	ASMOGA	SEQ-MOGA	FOMOPSO	NSGAII	NSPSO	AWPSO
Mean	0.8952	1.7913	16.1103	1.6337	20.1084	3.6954
Variance	0.0018	0.0206	269.1585	0.9694	267.8379	7.1109

Table 6. The mean and variance of the hyper-volume metric for 30 independent runs for the I-Beam optimization.

	ASMOGA	SEQ-MOGA	FOMOPSO	NSGAII	NSPSO	AWPSO
Mean	871.8148	870.6849	857.6305	860.8421	854.9336	237.8956
Variance	0.2928	2.1292	59.4158	49.7240	365.5411	5.8077e+05

Table 7. The t-test of the spacing for the I-Beam optimization for 30 independent runs.

	ASMOGA	SEQ-MOGA	FOMOPSO	NSGAII	NSPSO	AWPSO
ASMOGA	H0 (P-value = 1)	H1 P-value = 3.9371e-39	H1 P-value = 4.2087e-06	H1 P-value = 1.2840e-04	H1 P-value = 2.6365e-08	H1 P-value = 3.4971e-07
SEQ-MOGA	H1 P-value = 3.9371e-39	H0 (P-value = 1)	H1 P-value = 1.2373e-05	H0 P-value = 0.3894	H1 P-value = 8.3100e-08	H1 P-value = 2.4799e-04
FOMOPSO	H1 P-value = 4.2087e-06	H1 P-value = 1.2373e-05	H0 (P-value = 1)	H1 P-value = 1.0571e-05	H0 P-value = 0.3486	H1 P-value = 1.3425e-04
NSGAII	H1 P-value = 1.2840e-04	H0 P-value = 0.3894	H1 P-value = 1.0571e-05	H0 (P-value = 1)	H1 P-value = 7.0855e-08	H1 P-value = 1.9895e-04
NSPSO	H1 P-value = 2.6365e-08	H1 P-value = 8.3100e-08	H0 P-value = 0.3486	H1 P-value = 7.0855e-08	H0 (P-value = 1)	H1 P-value = 1.1975e-06
AWPSO	H1 P-value = 3.4971e-07	H1 P-value = 2.4799e-04	H1 P-value = 1.3425e-04	H1 P-value = 1.9895e-04	H1 P-value = 1.1975e-06	H0 (P-value = 1)

Table 8. The t-test of the hyper-volume metric for the I-Beam optimization for 30 independent runs.

	ASMOGA	SEQ-MOGA	FOMOPSO	NSGAI	NSPSO	AWPSO
ASMOGA	H0 (P-value = 1)	H1 P-value = 1.9628e-04	H1 P-value = 2.5395e-14	H1 P-value = 9.0123e-12	H1 P-value = 1.0207e-05	H1 P-value = 2.7319e-05
SEQ-MOGA	H1 P-value = 1.9628e-04	H0 (P-value = 1)	H1 P-value = 8.5974e-13	H1 P-value = 4.4558e-10	H1 P-value = 3.3305e-05	H1 P-value = 2.8107e-05
FOMOPSO	H1 P-value = 2.5395e-14	H1 P-value = 8.5974e-13	H0 (P-value = 1)	H0 P-value = 0.0976	H1 P-value = 0.4765	H1 P-value = 3.9000e-05
NSGAI	H1 P-value = 9.0123e-12	H1 P-value = 4.4558e-10	H0 P-value = 0.0976	H0 (P-value = 1)	H0 P-value = 0.1177	H1 P-value = 3.5992e-05
NSPSO	H1 P-value = 1.0207e-05	H1 P-value = 3.3305e-05	H1 P-value = 0.4765	H0 P-value = 0.1177	H0 (P-value = 1)	H1 P-value = 4.1878e-05
AWPSO	H1 P-value = 2.7319e-05	H1 P-value = 2.8107e-05	H1 P-value = 3.9000e-05	H1 P-value = 3.5992e-05	H1 P-value = 4.1878e-05	H0 (P-value = 1)

Table 9. Results of ASMOGA (the proposed method in this paper) compared with other literature for the I-Beam optimization. The generation number is 1000.

	f_1	f_2	Design variables $[x_1, x_2, x_3, x_4]$
Hajela and Shih (1990)	206.14	0.0205	[80, 39.79, 0.9, 1.72]
Huanga and Ying-Kui Gu (2006)	276.4525	0.0143	[80.0000, 50.0000, 0.9000, 2.0820]
Annamdas and Rao (2009)	132.5374	0.0375	[80, 34.53, 0.9, 0.9]
ASMOGA	116.333	0.070813	[60.001240, 35.016105, 0.9000778, 0.9134597]

The blue highlight in all tables shows the superiority of an algorithm in the optimization compared to the others.

4. Conclusions

In this paper, the new Archive Sequential Multi-objective Algorithm (ASMOGA) based on the Sequential Multi-objective Algorithm (SEQ-MOGA) was presented. In ASMOGA, the concept of archive is combined with SEQ-MOGA in order to increase the efficiency of a local search. Then the standard test functions such as ZDT1, ZDT2, ZDT3, and ZDT4 and a real problem (the design of I-Beam) were used to compare ASMOGA with the other well-known multi-objective optimization methods such as NSGAI, NSPSO, and AWPSO. In this comparison, the performance metrics such as the generational distance, spread, hyper-volume, spacing, and t-test (a statistical test) were used. However, the superiority of ASMOGA was found to be better than the other studied approaches here. For future work, we have decided to use ASMOGA in order to optimize the complex scientific problems.

References

[1] B. Huang, B. Buckley, and T.-M. Kechadi, "Multi-objective feature selection by using NSGA-II for customer churn prediction in telecommunications," *Expert Systems with Applications*, vol. 37, no. 5, pp. 3638-3646, 2010.

[2] M. R. Nikoo, I. Varjavand, R. Kerachian, M. D. Pirooz, and A. Karimi, "Multi-objective optimum design of double-layer perforated-wall breakwaters: Application of NSGA-II and bargaining models," *Applied Ocean Research*, vol. 47, pp. 47-52, 2014.

[3] A. Nourbakhsh, H. Safikhani, and S. Derakhshan, "The comparison of multi-objective particle swarm optimization and NSGA II algorithm: applications in centrifugal pumps," *Engineering Optimization*, vol. 43, no. 10, pp. 1095-1113, 2011.

[4] G. Sun, G. Li, Z. Gong, G. He, and Q. Li, "Radial basis functional model for multi-objective sheet metal forming optimization," *Engineering Optimization*, vol. 43, no. 12, pp. 1351-1366, 2011.

[5] J. Zeng, X. Zhang, and X. Guan, "Path Planning for General Aircrafts Under Complex Scenarios Using an Improved NSGA-II Algorithm," *Journal of Computational Information Systems*, vol. 9, no. 16, pp. 6545-6553, 2013.

[6] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007.

[7] F. Sabahi, "Fuzzy Adaptive Granulation Multi-Objective Multi-microgrid Energy Management," *Journal of AI and Data Mining*, vol. 8, no. 4, pp. 481-489, 2020.

[8] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001.

[9] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE transactions on*

Evolutionary Computation, vol. 3, no. 4, pp. 257-271, 1999.

[10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182-197, 2002.

[11] K. K. Annamdas and S. S. Rao, "Multi-objective optimization of engineering systems using game theory and particle swarm optimization," *Engineering optimization*, vol. 41, no. 8, pp. 737-752, 2009.

[12] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary computation*, vol. 2, no. 3, pp. 221-248, 1994.

[13] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *evolutionary computation, IEEE transactions on*, vol. 3, no. 4, pp. 257-271, 1999.

[14] R. B. Agrawal, K. Deb, and R. Agrawal, "Simulated binary crossover for continuous search space," *Complex systems*, vol. 9, no. 2, pp. 115-148, 1995.

[15] X. Li, "A non-dominated sorting particle swarm optimizer for multiobjective optimization," in *Genetic and Evolutionary Computation—GECCO 2003*, 2003: Springer, pp. 37-48.

[16] M. Mahfouf, M. Chen, and D. A. Linkens, "Adaptive weighted particle swarm optimisation for multi-objective optimal design of alloy steels," in *Parallel problem solving from nature-ppsn viii*, 2004: Springer, pp. 762-771.

[17] J. A. Rangel-González et al., "Fuzzy Multi-objective Particle Swarm Optimization Solving the Three-Objective Portfolio Optimization Problem," *International Journal of Fuzzy Systems*, pp. 1-9, 2020.

[18] L. Falahiazar and H. Shah-Hosseini, "The Sequential Multi-Objective Genetic Algorithm: A novel multi-objective genetic algorithm " presented at the International Conference on New Research Achievements in Electrical and Computer Engineering, International Federation of Inventors Association in Iran, 2016.

[19] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evolutionary computation*, vol. 8, no. 2, pp. 149-172, 2000.

[20] S.-K. S. Fan, J.-M. Chang, and Y.-C. Chuang, "A new multi-objective particle swarm optimizer using empirical movement and diversified search strategies," *Engineering Optimization*, vol. 47, no. 6, pp. 750-770, 2015.

[21] S.-T. Hsieh, S.-Y. Chiu, and S.-J. Yen, "An improved multi-objective genetic algorithm for solving multi-objective problems," *Applied Mathematics & Information Sciences*, vol. 7, no. 5, p. 1933, 2013.

[22] M. Kim, T. Hiroyasu, M. Miki, and S. Watanabe, "SPEA2+: Improving the performance of the strength Pareto evolutionary algorithm 2," in *International Conference on Parallel Problem Solving from Nature*, 2004: Springer, pp. 742-751.

[23] Y. Xiang, Y. Zhou, and H. Liu, "An elitism based multi-objective artificial bee colony algorithm," *European Journal of Operational Research*, vol. 245, no. 1, pp. 168-193, 2015.

[24] Z. Xu, "A Novel Hybrid Algorithm for Constrained Multi-objective Optimization," *International Journal of Hybrid Information Technology*, vol. 7, no. 3, pp. 265-274, 2014.

[25] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," *TIK-report*, vol. 103, 2001.

[26] P. Hajela and C.-J. Shih, "Multiobjective optimum design in mixed integer and discrete design variable problems," *AIAA journal*, vol. 28, no. 4, pp. 670-675, 1990.

[27] R. Eberhart and J. Kennedy, "Particle swarm optimization, proceeding of IEEE International Conference on Neural Network," *Perth, Australia*, pp. 1942-1948, 1995.

[28] X. Li, "A non-dominated sorting particle swarm optimizer for multiobjective optimization," in *Genetic and Evolutionary Computation Conference*, 2003: Springer, pp. 37-48.

[29] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, 1987: Hillsdale, NJ: Lawrence Erlbaum, pp. 41-49.

[30] V. S. Ghomsheh, M. A. Khanehsar, and M. Teshnehlal, "Improving the non-dominate sorting genetic algorithm for multi-objective optimization," in *Computational Intelligence and Security Workshops, 2007. CISW 2007. International Conference on*, 2007: IEEE, pp. 89-92.

[31] R. Cheng and M. Gen, "Genetic algorithms for multi-row machine layout problem," *Engineering Design and Automation*, pp. 876-881, 1996.

[32] Z. Michalewicz, C. Z. Janikow, and J. B. Krawczyk, "A modified genetic algorithm for optimal control problems," *Computers & Mathematics with Applications*, vol. 23, no. 12, pp. 83-94, 1992.

[33] D. A. Van Veldhuizen, "Multiobjective evolutionary algorithms: classifications, analyses, and new innovations," AIR FORCE INST OF TECH WRIGHT-PATTERSONAFB OH SCHOOL OF ENGINEERING, 1999.

[34] Y. Chen, X. Chen, and C. Wang, "Experimental and finite element analysis research on I-beam under web crippling," *Materials and Structures*, vol. 49, no. 1-2, pp. 421-437, 2016.

- [35] N. D. Hai, H. Mutsuyoshi, S. Asamoto, and T. Matsui, "Structural behavior of hybrid FRP composite I-beam," *Construction and Building Materials*, vol. 24, no. 6, pp. 956-969, 2010.
- [36] Q.-Y. Song, A. Heidarpour, X.-L. Zhao, and L.-H. Han, "Performance of double-angle bolted steel I-beam to hollow square column connections under static and cyclic loadings," *International Journal of Structural Stability and Dynamics*, p. 1450098, 2014.
- [37] Q.-Y. Song, A. Heidarpour, X.-L. Zhao, and L.-H. Han, "Post-earthquake fire behavior of welded steel I-beam to hollow column connections: An experimental investigation," *Thin-Walled Structures*, vol. 98, pp. 143-153, 2016.
- [38] W. Wang, T.-M. Chan, and H. Shao, "Numerical investigation on I-beam to CHS column connections equipped with NiTi shape memory alloy and steel tendons under cyclic loads," in *Structures*, 2015, vol. 4: Elsevier, pp. 114-124.
- [39] Hong-Zhong Huang and X. D. Ying-Kui Gu, "An interactive fuzzy multi-objective optimization method for engineering design," *Engineering Applications of Artificial Intelligence*, Elsevier, 2006.
- [40] D. Veldhuizen, "Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. 1999," *School of Engineering of the Air Force Institute of Technology, Dayton, Ohio*, 1999.
- [41] K. Bösecke, "Value creation in mergers and acquisitions—theoretical paradigms and past research," in *Value Creation in Mergers, Acquisitions, and Alliances*. Dissertation Jacobs University Bremen, 2009: Springer, 2009, p. 92.
- [42] T. Lumley, P. Diehr, S. Emerson, and L. Chen, "The importance of the normality assumption in large public health data sets," *Annual review of public health*, vol. 23, no. 1, pp. 151-169, 2002.
- [43] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—a comparative case study," in *International conference on parallel problem solving from nature*, 1998: Springer, pp. 292-301.
- [44] J. R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," Air Force Inst of Tech Wright-Patterson AFB OH, 1995.
- [45] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization," *Journal of Heuristics*, vol. 15, no. 6, p. 617, 2009.

ارائه یک الگوریتم چندهدفه‌ی ترتیبی آرشیوی

لیلا فلاحی آذر^۱، وحید صیدی^{۲*} و میترا میرزازایی^۱

^۱ دانشگاه آزاد اسلامی، واحد علوم و تحقیقات، گروه مهندسی برق و کامپیوتر، تهران، ایران.

^۲ گروه مهندسی کامپیوتر، واحد تهران جنوب، دانشگاه آزاد اسلامی، تهران، ایران.

ارسال ۲۰۲۰/۰۴/۲۸؛ بازنگری ۲۰۲۰/۱۲/۳۰؛ پذیرش ۲۰۲۱/۰۳/۰۱

چکیده:

بسیاری از مسائل دنیای واقعی از جمله مسائل چندهدفه هستند. در سالهای اخیر برای بهینه‌سازی مسائل چندهدفه، الگوریتم‌های چندهدفه تکاملی نتایج بسیار خوبی را نشان داده‌اند. بنابراین ارائه الگوریتم‌های کارا تر و بهتر همواره سبب توسعه و گسترش علم خواهد شد. در این مقاله یک الگوریتم تکاملی چند هدفه جدید، با نام الگوریتم چندهدفه‌ی ترتیبی آرشیوی ۱ (ASMOGA) ارائه شده است. در بسیاری از مقالات سعی شده است تا مشکلات الگوریتم ژنتیک مرتب‌سازی نامغلوب ۲ (NSGAII) به عنوان یکی از کاربردترین الگوریتم‌های تکاملی، برطرف شود و الگوریتم‌های بسیاری از جمله الگوریتم چندهدفه‌ی ترتیبی ۳ (SEQ-MOGA) ارائه شده است. در SEQ-MOGA روشی جدید برای انتخاب نسل بعدی ارائه شده است تا افراد به صورت منظم و پشت‌سرهم مرتب شوند و بتوانند تمامی ناحیه‌ی پیشانی پاسخ را پوشش دهند. در این تحقیق، جهت بهبود SEQ-MOGA، الگوریتم چندهدفه‌ی ترتیبی آرشیوی (ASMOGA) ارائه شده است. ASMOGA از روش آرشیو کروموزوم‌ها برای ذخیره تاریخچه جستجو استفاده می‌کند تا تنوع در فضای تصمیم‌گیری از بین نرود. از ASMOGA برای بهینه‌سازی توابع تست ZDT و طراحی I-Beam استفاده شده است و نتایج بدست آمده با چند الگوریتم شناخته شده مورد مقایسه قرار گرفته است. نتایج بهینه‌سازی با معیارهایی مانند ابرحجم، فاصله‌ی نسلی، فاصله بندی و t-test (یک آزمون آماری) مورد ارزیابی قرار گرفته است و برتری الگوریتم پیشنهادی به طور واضح مشخص است.

کلمات کلیدی: الگوریتم‌های تکاملی چند هدفه (MOEA)، الگوریتم ژنتیک مرتب‌سازی نامغلوب (NSGAII)، الگوریتم چندهدفه‌ی ترتیبی (SEQ-MOGA)، الگوریتم چندهدفه‌ی ترتیبی آرشیوی (ASMOGA)، توابع تست استاندارد، t-test.

¹ . Archive Sequential Multi-Objective Algorithm

² . Non-dominated Sorting Genetic Algorithm II (NSGAII)

³ . Sequential Multi-Objective Algorithm