Scientia Iranica A (2019) 26(3), 1249–1265

Sharif University of Technology
Scientia Iranica
Transactions A: Civil Engineering
http://scientiairanica.sharif.edu

Research Note

The application of multi-objective charged system search algorithm for optimization problems

A. Ranjbar[a], **S. Talatahari**[b,*], **and F. Hakimpour**[a]

a. *Department of GIS Engineering, Faculty of Surveying Engineering, Tehran University, Tehran, Iran.*
b. *Department of Civil Engineering, University of Tabriz, Tabriz, Iran.*

Received 11 July 2017; accepted 16 February 2019

KEYWORDS
Charged system search;
Meta-heuristics;
Multi-objective optimization;
Pareto optimal;
Multi-objective charged system search.

Abstract. The charged system search algorithm is a relatively new optimization algorithm developed based on some principles from physics and mechanics. This paper presents an approach in which Pareto dominance is incorporated into the charged system search in order to allow this algorithm to handle problems with some multi-objective functions; the proposed algorithm will be called Multi-Objective Charged System Search (MOCSS). Well-known mathematical and engineering benchmarks were used to evaluate the proposed algorithm, and the results were compared with those of other new approaches. The results of implementing an algorithm on some test problems show that the proposed algorithm outperforms other algorithms in terms of generational distance, maximum spread, spacing, coverage of two sets, and hypervolume indicator. Results of well-known mathematical examples indicate that an approach is highly competitive and can be considered as a viable alternative to solving multi-objective optimization problems. These results encourage the application of the proposed method to more complex and real-world multi-objective optimization problems. The proposed method can deal with highly nonlinear problems with complex constraints and diverse Pareto optimal sets.

1. Introduction

Many realistic problems contain simultaneous optimization of several objectives that may conflict with each other and with other nonlinear constraints, if available [1-7]. These types of problems are known as Multi-Objective Problems (MOPs). Multi-objective optimization (also known as vector optimization, multi-criteria/attribute optimization, multi-objective programming, or Pareto optimization) is defined as the process of finding a decision vector to optimize a set of objective functions that satisfies some certain constraints [8,9], while the aim of single-objective optimization is to optimize just one objective function. In contrast with a single-objective optimization, multi-objective problems are more difficult and complex [10,11]. Some reasons for this are as follows:

1. In single-objective optimization, the fitness of solutions is reachable easily due to the existence of just one objective function, while no single unique solution can be determined as the best for multi-objective optimization; instead, a set of non-dominated solutions should be found in order to obtain a good approximation of the true Pareto fronts [3,12-14], leading to a trade-off among the objectives [15,16];
2. Algorithms that work well for single-objective problems usually cannot directly be used for multi-

[*]. *Corresponding author.*
E-mail address: talatahari@tabrizu.ac.ir (S. Talatahari)

doi: 10.24200/sci.2018.20184

objective ones, and it is necessary to consider some special conditions. As a simple way, one can combine multiple objectives into a single objective using some weighted sum method [17];

3. Even if a multi-objective algorithm can find solutions on a Pareto front, there is no guarantee that distribution of multiple Pareto points becomes uniform, and this may reduce the applicability of the results [3,14,17].

Therefore, developing an efficient multi-objective algorithm for solving multi-objective optimization problems seems inevitable.

Nowadays, Multi-Objective Evolutionary Algorithms (MOEAs) have shown an acceptable performance in many benchmarks and real-world problems with their origins in engineering, scientific, and industrial areas [1]. The main reason for the popularity of evolutionary algorithms for solving multi-objective optimization is their population-based nature and ability to find multiple optima simultaneously. In 1985, Schaffer was probably the first to use Vector Evaluated Genetic Algorithms (VEGA) to solve multi-objective optimization without using any composite aggregation and by combining all objectives into a single objective [18]. After that, a wide variety of MOEAs have been suggested such as Micro-Genetic Algorithm (Micro-GA) [19], Non-dominated Sorting Genetic Algorithm (NSGA) [20], new variant of NSGA or NSGA-II [21], Strength Pareto Evolutionary Algorithm (SPEA) [22], SPEA2 [23], Pareto Archive Evolution Strategy (PAES) [24], Pareto Differential Evolution Approach (PDEA) [25], MOEA/D: A Multi-Objective Evolutionary Algorithm based on Decomposition [26], NSGAII based on Differential Evolution (NSGAII-DE [27]), a hybrid multi-objective particle swarm optimization and decision-making procedure for optimal design of truss structures [28], the third version of Generalized Differential Evolution (GDE3) [29], Multi-Objective Differential Evolution-the Ranking-based Mutation Operator (MODE-RMO) [30], Multi-Objective Particle Swarm Optimization (MOPSO) [31], Differential Evolution for Multi-objective Optimization (DEMO) [32], a novel hybrid charged system search and particle swarm optimization method for multi-objective optimization [33], Multi-Objective Differential Evolution (MODE)[34], multi-objective bees algorithms (Bees) [35], and Non-dominated Rank Genetic Algorithm (NRGA) [36]. In recent years, some other types of algorithms have also been developed such as Multi-Objective Cuckoo Search (MOCS) [37], Multi-Objective Firefly Algorithm (MOFA) [38], a new multi-swarm multi-objective optimization method for structural design [39], a swarm-based memetic evolutionary algorithm for multi-objective optimization of large structures [40], Multi-Objective Flower Pollination Algorithms (MOFPA) [17], and multi-objective optimization method based on sensitivity analysis [14].

Regardless of the number of these methods and their differences, they share some defections as follows:

I. The final distribution of Pareto points is not often well-spread; therefore, maximum information on the Pareto cannot often be obtained [13,41];

II. The finding results often require heavy computation and are time consuming.

Therefore, the development of a new multi-objective optimization method to resolve some of these drawbacks seems necessary.

Kaveh and Laknejadi (2011) [33] used a hybrid charged system search and particle swarm multi-objective optimization, where the answers space was divided based to some spaces in order to find a uniform Pareto point. A multi-objective charged system search was developed by Kaveh and Massoudi [42]. These algorithms are extended to the single-objective Charged System Search (CSS), as introduced by Kaveh and Talatahari [43,44].

In the present paper, another variant of Multi-Objective Charged System Search (MOCSS) is presented, where the idea of the non-dominated method is used.

The rest of the paper is organized as follows. Section 2 describes the basic characteristics of the standard CSS. In Section 3, the multi-objective CSS algorithm will be presented in detail. The fundamental concept of the utilized constraint-handling method for MOCSS in detail is described in Section 4. The benchmark function, multi-objective performance metrics, and computational results are presented in Section 5. Validation of the MOCSS by some engineering design problems will be presented in Section 6. Finally, some relevant issues, future works, and conclusions are drawn in Section 7.

2. A brief review on standard charged system search

In physics, the electric field around an electric charge is the space surrounding it and applies a force to other electrically charged objects. The Coulomb law determines the electric field surrounding a point charge. Its value is proportional with the product of two charged particles and inversely square of the separation distance between the particles directed along the line. Based on Gauss's law, the magnitude of the electric field at a point inside a charged sphere can be determined (proportional with the separation distance between the particles). By using these principles, the standard Charged System Search (CSS) defines a number of solution candidates or Charged Particles (CPs) that act as a charged sphere and can apply electrical forces to the other CPs. The resultant force acts on each CP creating acceleration, according to Newton's second law. Finally, by utilizing the Newtonian mechanics, the position of each CP is determined at any time based on its previous position, velocity, and acceleration in the search space [43]. The main formula of electrical physics for calculating the electrical force between two CPs is as follows:

$$\mathbf{F}_j = q_j \sum_{i,i\neq j} (\frac{q_i}{a^3} r_{ij}.i_1 + \frac{q_i}{{r_{ij}}^2}.i_2) p_{ij} (\mathbf{X}_i - \mathbf{X}_j),$$

$$\begin{cases} j = 1,2,...,n \\ i_1 = 1, i_2 = 0 \Leftrightarrow r_{ij} < a \\ i_1 = 0, i_2 = 1 \Leftrightarrow r_{ij} \geq a \end{cases} \tag{1}$$

where:

a	The radius of the charged sphere;
n	The total number of CPs;
$\mathbf{F}_j$	The resultant force acting on the jth CP;
q_i	The magnitude of the charge;
r_{ij}	The separation distance between two charged particles;
p_{ij}	The probability of two charged particles;
$\mathbf{X}_i$ and $\mathbf{X}_j$	The positions of the ith and jth CPs.

The initial position of CPs is obtained through Eq. (2) in the search space:

$$x_{i,j} = x_{i,\min} + rand \times (x_{i,\max} - x_{i,\min}),$$
$$i = 1,2,...,n, \tag{2}$$

where $x_{i,j}$ determines the initial value of the ith variable for the jth CP; $x_{i,\min}$ and $x_{i,\max}$ are the minimum and the maximum allowable values for the ith variable; *rand* is a random number at the interval [0,1]; and n is the number of variables. The magnitude of the charge is calculated by the quality of solutions as follows:

$$q_i = \frac{fit(i) - fit(worst)}{fit(best) - fit(worst)}, \quad i = 1,2,...,n, \tag{3}$$

where $fit(best)$ and $fit(worst)$ are the best and the worst fitness of all particles so far; $fit(i)$ represents the objective function value or the fitness of agent i; n is the total number of CPs. In Eq. (4), a force is attractive as long as all good CPs can attract bad CPs and only some of bad agents attract good agents due to appropriate exploitation and exploration abilities as follows:

$$p_{ij} = \begin{cases} 1 & \frac{fit(i)-fit(best)}{fit(j)-fit(i)} > rand \;\text{ or }\; fit(j) > fit(i) \\ 0 & \text{else} \end{cases} \tag{4}$$

The separation distance, r_{ij}, between two charged particles is also defined as follows:

$$r_{ij} = \frac{\|\mathbf{X}_i - \mathbf{X}_j\|}{\left\|\frac{(\mathbf{X}_i + \mathbf{X}_j)}{2} - \mathbf{X}_{best}\right\| + \varepsilon}, \tag{5}$$

where $\mathbf{X}_i$ and $\mathbf{X}_j$ are the positions of the ith and jth CPs, $\mathbf{X}_{best}$ is the position of the best current CP, and ε is a small positive number to avoid singularities. The radius of the charged sphere (a) is considered as follows:

$$a = 0.1 \times \max(\{x_{i,\max} - x_{i,\min} | i = 1,2,...,n\}). \tag{6}$$

Further, the main formula of the CSS uses Newton's laws (with some modifications) for calculating the new position and velocity of each CP as follows:

$$\mathbf{X}_{j,new} = 0.5 rand_{j1}.(1 + \frac{iter}{iter_{\max}}) . \sum_{i,i\neq j} (\frac{q_i}{a^3} r_{ij}.i_1 + \frac{q_i}{{r_{ij}}^2}.i_2) p_{ij} (\mathbf{X}_i - \mathbf{X}_j) + 0.5 rand_{j2}.(1 - \frac{iter}{iter_{\max}}).\mathbf{V}_{j,old} + \mathbf{X}_{j,old}, \tag{7}$$

$$\mathbf{V}_{j,new} = \mathbf{X}_{j,new} - \mathbf{X}_{j,old}, \tag{8}$$

where *iter* is the actual number of iterations, and $iter_{\max}$ is the maximum number of iterations.

3. Multi-objective charged system search

The proposal optimization algorithm, so-called Multi-Objective Charged System Search (MOCSS), is used for solving multi-objective problems by combining CSS algorithm with Non-dominated Sorting (NS) for good convergence and high diversity of Pareto front [45], respectively. NS sorts the solutions on the basis of non-domination and, then, forms the levels of Pareto fronts. For selecting the numbers of the best solution, the solutions with the highest Pareto front rank are chosen; if required, the other solutions are selected for the next Pareto front. This process is repeated until the Crowding Distance (CD) condition is satisfied (according to Figure 1). This article utilized the mutation function from GA in order to prevent early convergence. The following pseudo-codes summarized the MOCSS algorithm:

Level 1: Initialization

Step 1: Initialize specification of the optimization problem and algorithm parameters;

Step 2: Initialize the first positions of charged particles and their associated velocities;

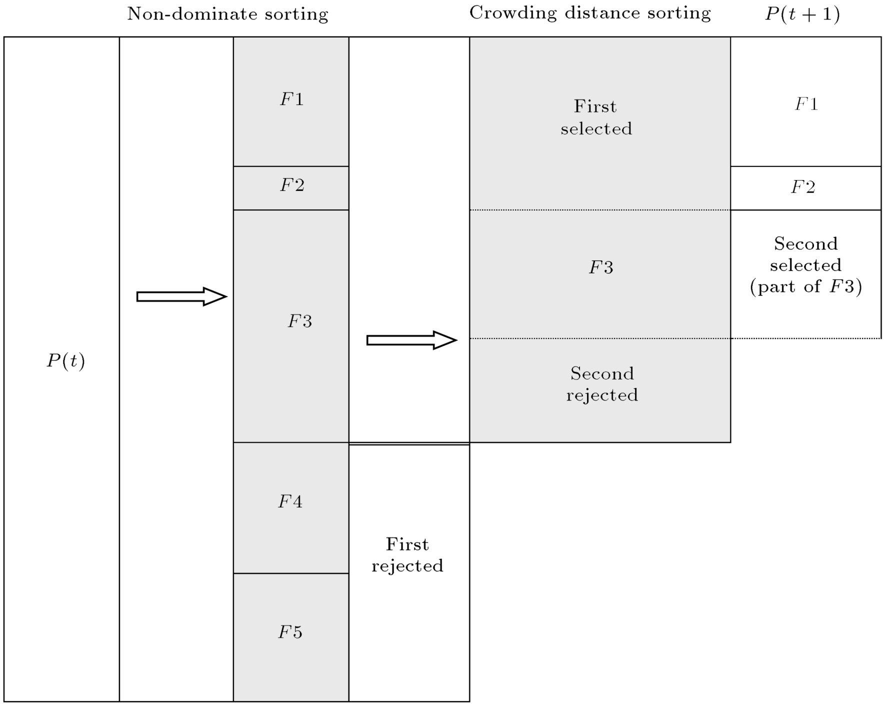


Figure 1. Flow diagram that shows the way in which the NS works. $P(t)$ is the population at generation t; $F1$ is the best solution; and $F2$ is the second best solutions, and so on (Coello et al., p. 94, 2007) [4].

Step 3: Evaluate all CPs;

Step 4: Determine the non-dominated solutions for the initial CPs.

Level 2: Search

Step 1: Determine the probability of moving and calculate the attracting force vector for each CP;

Step 2: Select the leader;

Step 3: Move each CP to the new position and find their velocities;

Step 4: Mutate some CPs;

Step 5: Rank CPs according to the NS approach.

Level 3: Terminating criterion controlling

Repeat search level steps until a terminating criterion is satisfied

The flowchart of the MOCSS algorithm is illustrated in Figure 2.

4. Constraint-handling method for the MOCSS

Consider the general form of a constrained multi-objective optimization problem [46,47] as follows:

Find $\mathbf{x}$ that minimizes

$$F(\mathbf{x}) = (f_1(\mathbf{x}), ..., f_k(\mathbf{x})), \tag{9}$$

subject to:

$$G(\mathbf{x}) = (g_1(\mathbf{x}), ..., g_h(\mathbf{x})) \leq 0, \tag{10}$$

where $\mathbf{x} = (x_1, ..., x_{nVar})$ is the vector of solution that minimizes objective function(s) $F(\mathbf{x})$ while satisfying constraint(s) $G(\mathbf{x}) \leq 0$. The number of design parameter(s), objective function(s), and constraint(s) are denoted by $nVar$, k, and h, respectively.

Multi-Objective Evolutionary Algorithms (MOEAs) are robust and efficient multi-objective optimization algorithms; however, EAs do not have any explicit mechanism to handle constraints, while most real-world design multi-objective optimization problems have multiple constraints [48]. The penalty function method is a traditional approach to handling the constraints of single-objective optimization problems. However, this method requires careful tuning of the penalty function coefficients to obtain a satisfactory design. Moreover, the application of this method to a multi-objective optimization problem raises another problem: How to combine multiple constraints with multiple objectives [11,48].

Many previous constraint-handling methods need to tune some parameters to make a balance between the objective(s) and constraint(s). This study uses a constraint-handling method proposed by Oyama (2007) [48], which does not need any parameters to be tuned for constraint handling and it can always be used

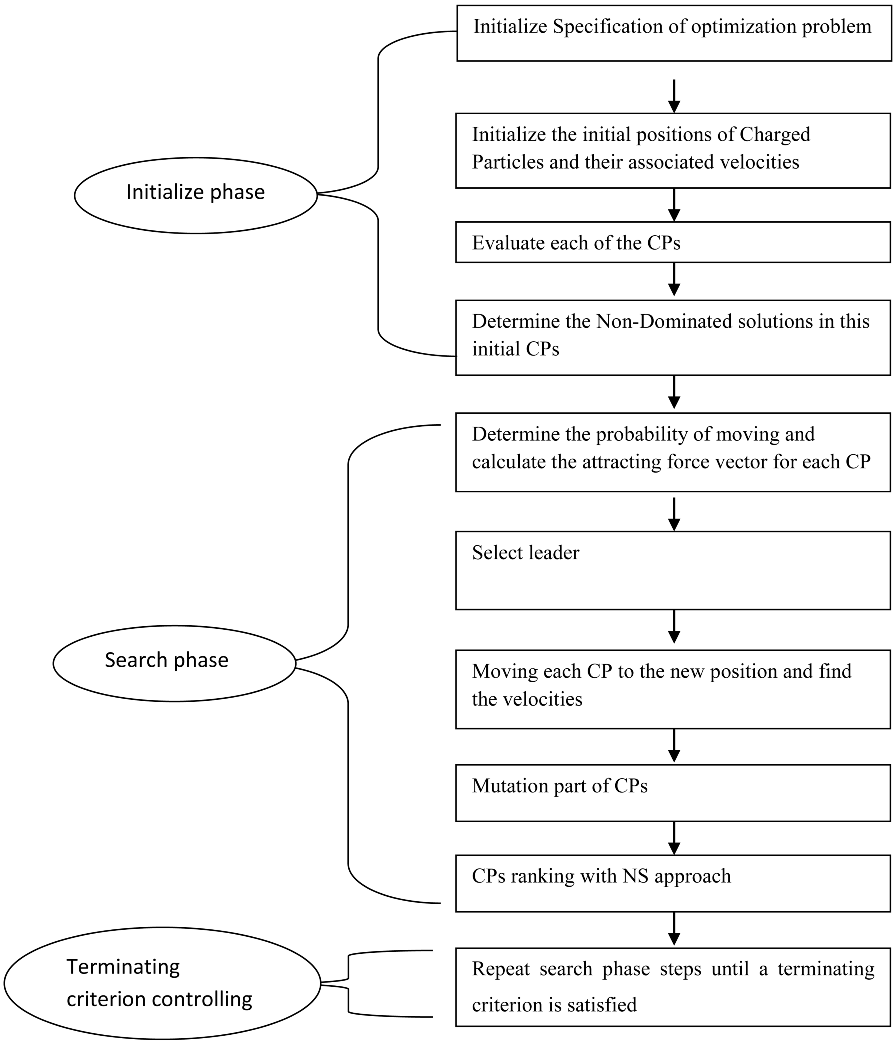

Initialize phase
Initialize Specification of optimization problem
Initialize the initial positions of Charged Particles and their associated velocities
Evaluate each of the CPs
Determine the Non-Dominated solutions in this initial CPs
Search phase
Determine the probability of moving and calculate the attracting force vector for each CP
Select leader
Moving each CP to the new position and find the velocities
Mutation part of CPs
CPs ranking with NS approach
Terminating criterion controlling
Repeat search phase steps until a terminating criterion is satisfied

Figure 2. The flow-chart of MOCSS algorithm.

even when all individuals in the initial population are infeasible, or the amount of violation of each constraint is significantly different. The method is described in the following.

Definition 1 (constrained Pareto dominance):

Solution i is said to constrained-dominate solution j if any of the following conditions are true:

1. Solutions i and j are both feasible, and solution i dominates solution j in the objective function space. It should be noted that solution $\mathbf{x}_i$ is said to dominate solution $\mathbf{x}_j$ if $f_k(\mathbf{x}_i)$ is no worse than $f_x(\mathbf{x}_j)$ for all objectives, and it is better for at least one of them [21,49]:

$$f_k(\mathbf{x}_i) \leq f_k(\mathbf{x}_j), \quad \forall i = 1, 2, ..., k$$

$$f_k(\mathbf{x}_i) < f_k(\mathbf{x}_j), \quad \exists i \in \{1, 2, ..., k\}. \tag{11}$$

Thus, a set of solutions is said to be a Pareto front or Pareto solution if no element of this set dominates any other solutions [50]. For more details on Pareto optimal solutions, one can be referred to [31,51].

2. Solution i is feasible and solution j is not.

3. Solutions i and j are both infeasible; yet, solution i dominates solution j in the constraint space.

Definition 2 (constraint space dominance):

Solution i is said to dominate solution j in the constrained space if both of the following conditions are true:

1. Solutions i is not worse than solution j in all constraints, i.e.:

$$\forall G_n(\mathbf{x}_i) \leq G_n(\mathbf{x}_j). \tag{12}$$

2. Solution i is strictly better than solution j for at least one constraint, i.e.:

$$\forall G_n(\mathbf{x}_i) < G_n(\mathbf{x}_j), \tag{13}$$

where:

$$G_n(\mathbf{x}) = \max(0, g_n(\mathbf{x})), \quad n = 1, 2, ..., k. \tag{14}$$

By means of Oyama's constraint-handling approach, niching based on the number of constraint violations was applied to infeasible solutions. Here, a standard fitness sharing [52] was applied to infeasible designs based on their constraint violations as follows:

$$rank'(\mathbf{x}_i) = rank(\mathbf{x}_i) \times Penalty(\mathbf{x}_i),$$

$$Penalty(\mathbf{x}_i) = 1 + \sum_{j=1, j\neq i}^{npop} sh_{ij},$$

$$sh_{ij} = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{share}}\right)^{\alpha} & \mathrm{d}_{ij} < \sigma_{share} \\ 0 & \mathrm{d}_{ij} \geq \sigma_{share} \end{cases}$$

$$\sigma_{share} = \sum_{n=1}^{h} (g\max_n - g\min_n)/npop,$$

$$d_{ij} = \sqrt{\sum_{n=1}^{h} (g_n(\mathbf{x}_i) - g_n(\mathbf{x}_j))^2},$$

$$g\max_n = \max(\{g_n(\mathbf{x}_1), ..., g_n(\mathbf{x}_{npop})\}),$$

$$g\min_n = \min(\{g_n(\mathbf{x}_1), ..., g_n(\mathbf{x}_{npop})\}), \tag{15}$$

where *npop* is population size and α is set to 0.4.

5. Numerical investigation

5.1. Benchmark problems

There are many different multi-objective benchmark functions for evaluating the performance of algorithms [22,37,53]. In this paper, for validating the MOCSS, ten of these functions have been selected containing convex (ZDT1 [8] and MOP1 [18]), non-convex (ZDT2 and MOP2 [8,13,54]), and discontinuous Pareto fronts with more complex Pareto set problems (ZDT3 [8], DTLZ1, DTLZ2, DTLZ3, DTLZ4, and DTLZ5 test functions [55]). Table 1 presents the details of these examples.

5.2. Multi-objective performance metrics

To evaluate the performance of multi-objective optimization algorithms, a general approach is utilized to compare quantitative results [13,56] or the amount of relative distribution on the Pareto front for test functions [37]. In order to determine a quantitative assessment of the performance of a multi-objective optimization algorithm, three issues are normally taken into consideration [57]:

I) The distance of the Pareto front produced by an algorithm with respect to the real Pareto front;

II) The spread of solutions found;

III) The number of elements of the Pareto optimal set found.

For the first two, small values are better than the larger one, while the number of elements of the Pareto optimal set should be maximized for the last one.

In order to compare the results of different MOP problems, different performance metrics are usually utilized in the literature [31]; the following subsections describe these metrics.

I. **Generational Distance (GD).** The concept of generational distance was introduced by Van Veldhuizen and Lamont [13,54,58] for estimating how far the elements are in the set of generated Pareto fronts so far from those in the Pareto front true, and it is defined as follows:

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n}, \tag{16}$$

where n is the number of so far solutions in PF_{known}, and d_i is the Euclidean distance of the objective space between each of solutions and the nearest member of the true Pareto front. It should be noted that a value of $GD = 0$ indicates that all the generated elements are in the true Pareto front, i.e., $PF_{true} = PF_{known}$. Therefore, any other value indicates how "far" we are from the global Pareto front of our problem. This metric needs to know PF_{true};

II. **Maximum Spread (MS).** The metric of Maximum Spread (MS) measures how "well" PF_{true} is covered by PF_{known} through hyper-boxes formed by the extreme function values observed in PF_{true} and PF_{known}. It is defined as shown in Box I: where m is the number of objectives, $f_i^{\max}$, $f_i^{\min}$, $F_i^{\max}$, and $F_i^{\min}$ are the maximum and minimum of the ith objective in PF_{known} and PF_{true}, respectively. A larger value of MS implies better spread of solutions. In this study, $F_i^{\max}$ and $F_i^{\min}$ are considered as the maximum and minimum of the ith objective in all the Pareto fronts obtained by various algorithms [49]. This metric needs to know PF_{true};

III. **Spacing (S).** The Spacing (S) metric numerically describes the spread of the vectors in

Table 1. Benchmark for multi-objective optimization.

Name of benchmark	Mathematical models	Description
MOP1 Schaffer's Min-Min (SCH)	$F = (f_1(x), f_2(x)),\ -10^3 \leq x_i \leq 10^3,$ $f_1(x) = x^2,$ $f_2(x) = (x-2)^2.$	P_{true} connected, PF_{true} convex benchmark function with convex Pareto front
MOP2	$F = (f_1(x), f_2(x)),\ i = 1, 2, 3,\ -4 \leq x_i \leq 4,$ $f_1(x) = 1 - \exp(-\sum_{i=1}^{n} (x_i - \frac{1}{\sqrt{n}})^2),$ $f_2(x) = 1 - \exp(-\sum_{i+1}^{n} (x_i + \frac{1}{\sqrt{n}})^2).$	P_{true} connected, PF_{true} concave number of decision variables scalable
ZDT1	$F = (f_1(x), f_2(x)),\ 0 \leq x_i \leq 1,\ n = 30,$ $f_1(x) = x_1,$ $f_2(x, g) = g(x) \times \left(1 - \sqrt{\frac{f_1}{g(x)}}\right),$ $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i.$	Has a convex Pareto-optimal front
ZDT2	$F = (f_1(x), f_2(x)),\ 0 \leq x_i \leq 1,\ n = 30,$ $f_1(x) = x_1,$ $f_2(x, g) = g(x) \times (1 - \frac{f_1}{g(x)})^2,$ $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i.$	Has a non-convex Pareto-optimal front
ZDT3	$F = (f_1(x), f_2(x)),\ 0 \leq x_i \leq 1,\ n = 30,$ $f_1(x) = x_1,$ $f_2(x, g) = g(x) \times \left(1 - \sqrt{\frac{f_1}{g(x)}} - \frac{f_1}{g(x)} \times \sin(10\pi f_1)\right),$ $g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i.$	Has a Pareto-optimal front disconnected, consisting of several noncontiguous convex parts
DTLZ1	$F = (f_1(x), f_2(x), f_3(x)),\ 0 \leq x_i \leq 1,\ n = 10,$ $f_1(x) = x_1 x_2 (1 + g(x)),$ $f_2(x) = x_1 (1 - x_2)(1 + g(x)),$ $f_3(x) = (1 - x_1)(1 + g(x)),$ where: $g(x) = 100(n-2) + 100 \sum_{i=3}^{n} \left\{(x_i - 0.5)^2 - \cos[20\pi(x_i - 0.5)]\right\}$ $\sum_{i=1}^{3} f_i = 1$ with $f_i > 0$	Its PF is linear Pareto-optimal front, separable, multimodal.
DTLZ2	$F = (f_1(x), f_2(x), f_3(x)),\ 0 \leq x_i \leq 1,$ $f_1(x) = \cos(\frac{x_1\pi}{2}) \cos(\frac{x_2\pi}{2})(1 + g(x)),$ $f_2(x) = \cos(\frac{x_1\pi}{2}) \sin(\frac{x_2\pi}{2})(1 + g(x)),$ $f_3(x) = \sin(\frac{x_1\pi}{2})(1 + g(x)),$ where: $g(x) = \sum_{i=3}^{n} (x_i - 0.5)^2,$ $\sum_{i=1}^{3} f_i^2 = 1$ with $f_i > 0$	Its PF is non-convex.
DTLZ3	$F = (f_1(x), f_2(x), f_3(x)),\ 0 \leq x_i \leq 1,$ $f_1(x) = \cos(\frac{x_1\pi}{2}) \cos(\frac{x_2\pi}{2})(1 + g(x)),$ $f_2(x) = \cos(\frac{x_1\pi}{2}) \sin(\frac{x_2\pi}{2})(1 + g(x)),$ $f_3(x) = \sin(\frac{x_1\pi}{2})(1 + g(x)),$ where: $g(x) = 100(n-2) + 100 \sum_{i=3}^{n} \left\{(x_i - 0.5)^2 - \cos\ [20\pi(x_i - 0.5]\right\}$ and $x = (x_1, x_2, ..., x_n)^T \in [0\ 1]^T$	PF_{true} concave, scalable, multimodal

Table 1. Benchmark for multi-objective optimization (continued).

Name of benchmark	Mathematical models	Description
DTLZ4	$F=(f_1(x),f_2(x),f_3(x)),\ 0\le x_i\le 1,$ $f_1(x)=\cos(\frac{{x_1}^{\alpha}\pi}{2})\cos(\frac{{x_2}^{\alpha}\pi}{2})(1+g(x)),$ $f_2(x)=\cos(\frac{{x_1}^{\alpha}\pi}{2})\sin(\frac{{x_2}^{\alpha}\pi}{2})(1+g(x)),$ $f_3(x)=\sin(\frac{{x_1}^{\alpha}\pi}{2})(1+g(x)),$ where: $g(x)=\sum_{i=3}^{n}(x_i-0.5)^2,\ \alpha= 100$ and $x=(x_1,x_2,...,x_n)^T\in[0\ 1]^T$	PF_{true} concave, separable, unimodal.
DTLZ5	$F=(f_1(x),f_2(x),f_3(x)),\ 0\le x_i\le 1,$ $f_1(x)=\cos(\frac{\theta_1\pi}{2})\cos(\frac{\theta_2\pi}{2})(1+g(x)),$ $f_2(x)=\cos(\frac{\theta_1\pi}{2})\sin(\frac{\theta_2\pi}{2})(1+g(x)),$ $f_3(x)=\sin(\frac{\theta_1\pi}{2})(1+g(x)),$ where: $g(x)=\sum_{i=3}^{n}(x_i-0.5)^2$ $\theta_1=x_1,\theta_2=\frac{(1+2x_2,g(x))}{2(1+g(x))}$ and $x=(x_1,x_2,...,x_n)^T\in[0\ 1]^T$	PF_{true} unimodal The function value of a Pareto optimal solution satisfies $\sum_{i=1}^{3}f_i^2=1$

$$MS=\left[\frac{1}{m}\sum_{i=1}^{m}\left[\frac{\min(f_i^{\max},F_i^{\max})-\max(f_i^{\min},F_i^{\min})}{F_i^{\max}-F_i^{\min}}\right]^2\right]^{\frac{1}{2}}. \tag{17}$$

Box I

PF_{known} [33,59]. This Pareto front metric measures the distance variance of neighboring vectors in PF_{known}. Eqs. (18) and (19) define this metric:

$$S=\left[\frac{1}{n-1}\sum_{i=1}^{n}(d_i-\overline{d})^2\right]^{\frac{1}{2}}\quad\text{where }\overline{d}=\frac{\sum_{i=1}^{n}d_i}{n}, \tag{18}$$

$$d_i=\min_j\left(\left|f_1^i(\overrightarrow{x})-f_1^j(\overrightarrow{x})\right|+\left|f_2^i(\overrightarrow{x})-f_2^j(\overrightarrow{x})\right|\right),$$

$$i,j=1,2,...,n,\quad i\neq j, \tag{19}$$

where n is the number of vectors in PF_{known}. When $S=0$, all members are spaced evenly apart. Note that this becomes important in the deception problems where all Pareto front vectors are equally spaced. This metric does not require the user to know PF_{true};

IV. **Coverage of two set (CS).** In order to compare the dominant relationship between two populations resulting from two different MOEAs, Zitzler et al. [2003] proposed the CS [56] that is measured to show how the final population of one algorithm dominates the final population of another algorithm. Eq. (20) defines this metric:

$$CS(X',X'')=\frac{\lvert\{a''\in X'';\exists a'\in X':a'\le a''\}\rvert}{\lvert X''\rvert}, \tag{20}$$

where X' and X'' are two sets of solutions resulting from different algorithms, where $a'\le a''$ means that a' dominates a'' if and only if $a'<a''$ or $a'=a''$. Function CS is defined as the mapping of the order pair (X',X'') to the interval [0,1]. In general, if all solutions in X' dominate all solutions in X'', then $CS(X',X'')=1$. In addition, $CS(X',X'')=0$ implies that none of the solutions in X'' is dominated. Note that both $CS(X',X'')$ and $CS(X'',X')$ need to be considered independently since they have distinct meanings and $CS(X',X'')$ is not necessarily equal to $1-CS(X'',X')$. The advantage of this Pareto compliant metric is that it is easy to calculate and provide a relative comparison based on dominant numbers between two MOEAs [57];

V. **Hypervolume indicator of set S.** Let the reference point be denoted by $Ref=(r_1,r_2,...,r_k)$. The hypervolume indicator of S (denoted as $Hv(S)$) is defined as the volume of the hypercube restricted by all points in S and Ref.

$$Hv(S) = Leb(\bigcup_{\vec{x}\in S} |f_1(\vec{x}_1), r_1| \times \Bigg| f_2(\vec{x}_2),$$

$$r_2 \Bigg| \times ... \times |f_k(\vec{x}_k), r_k|)), \qquad (21)$$

where k is the number of dimensions, $Leb(S)$ indicates the Lebesgue measure of S, and $|f_1(\vec{x}_1), r_1| \times |f_2(\vec{x}_2), r_2| \times ... \times |f_k(\vec{x}_k), r_k|$ represents the hypercube formed by points, dominated by $\vec{x}$ as Ref. [60].

5.3. Numerical results

For the MOP 1 example, Figure 3 shows the exponential-like decrease of GD as the iterations proceed. Clearly, it can be seen that the MOCSS algorithm indeed converges almost exponentially. The estimated Pareto fronts and true Pareto fronts of other functions are shown in Figure 4. In all of these figures, the horizontal axis is the first objective function, and the vertical axis is the second one. The figure shows that the MOCSS algorithm is able to find proper solutions for the benchmark examples. Further, the solutions are scattered among the Pareto fronts identically. Therefore, this algorithm is able to correctly obtain the Pareto front.

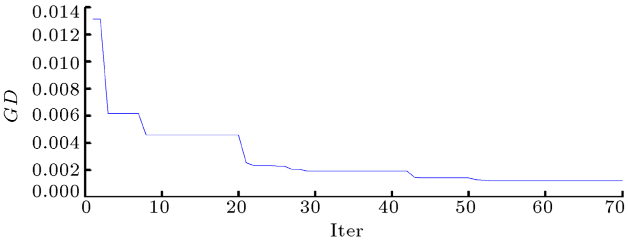


Figure 3. Convergence of the proposed MOCSS. The least-square distance (vertical axis) from the estimated front to the true front of MOP1 for the first 70 iterations (horizontal axis).

A careful scrutiny of Figure 4 indicates that the proposed MOCSS algorithm for finding solutions has outperformed all benchmarks. The answers that are close to the true Pareto front, uniformly dispersed on it, should be found. The performances of the proposed multi-objective approach are evaluated based on the multi-objective metrics in terms of GD, S, and MS. The results are summarized in Tables 2(a) to 2(c), (including Mean, standard deviation, Best, and Worst with 30 independent runs). In all benchmark functions, the values of GD and S are close to zero, and the value of MS is close to one. This means that the result of MS metric shows that the MOCSS approach has better covered PF_{known}, and the result of S metric shows that MOCSS approach has better spread of the answer.

In order to evaluate and compare the performances of the MOCSS with those of the other multi-objective optimization algorithms, the results of NSGA-II, VEGA, MODE, SPEA, Bees, DEMO, PDEA, MOEA/D. SPEA2, GDE3, NSGAII-DE, MODE-RMO, and MOFA are presented in Table 3. All results have been averaged over 30 independent runs.

Table 2(a). Results of the GD for the benchmarks (with 30 independent runs).

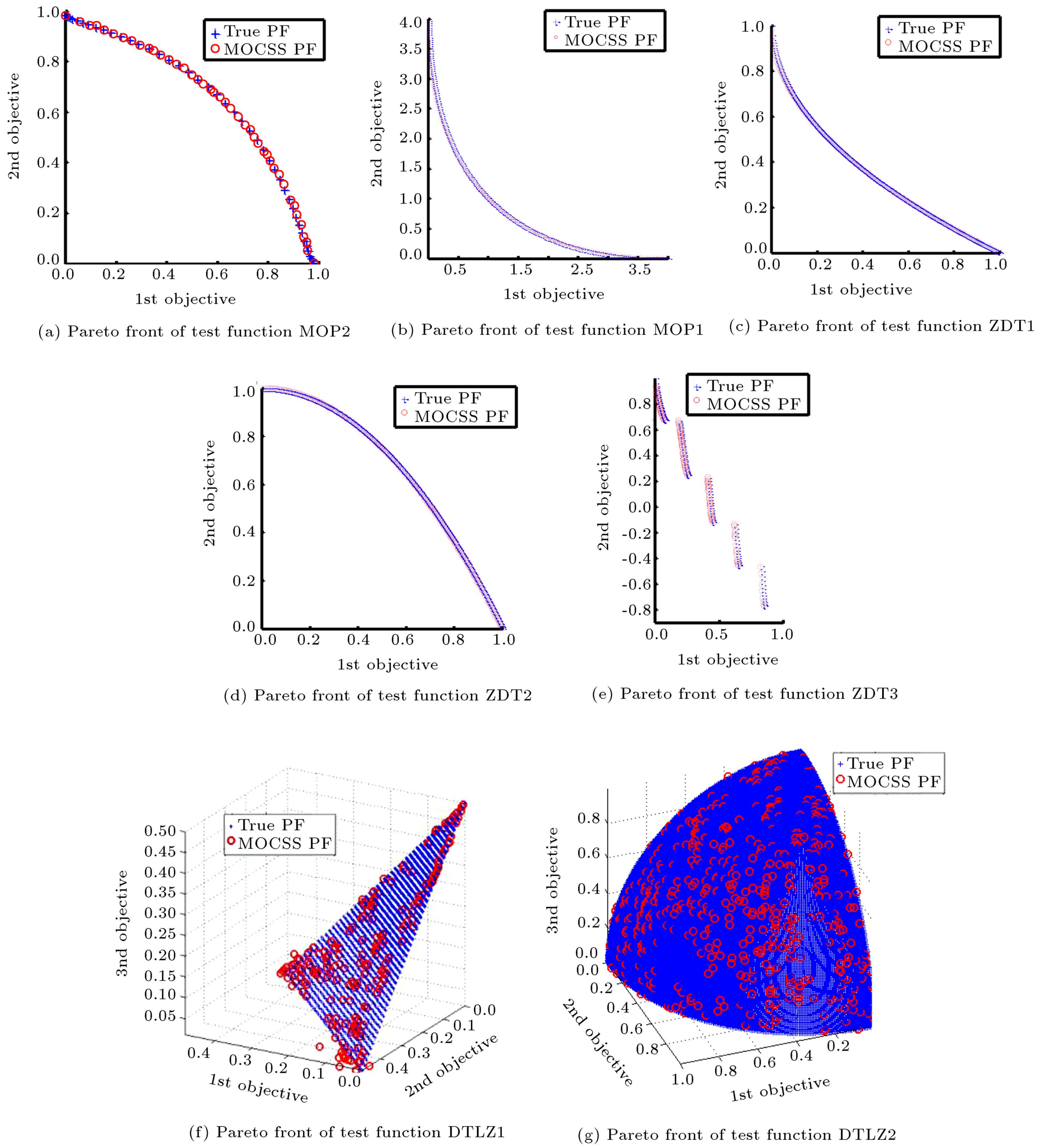
GD metric	MOP1	ZDT1	ZDT2	ZDT3
Mean	0.00136	0.000178	0.000148	0.0007938
Standard deviation	0.00013	0.000037	0.000012	0.0001043
Best	0.00110	0.000110	0.000123	0.000640
Worst	0.00170	0.000274	0.000164	0.000977

Table 2(b). Results of the MS for the benchmarks (with 30 independent runs).

MS metric	MOP1	ZDT1	ZDT2	ZDT3
Mean	0. 9934	1	0.9993	0.9591
Standard deviation	0.0109	1	0.0032	0.0783
Best	1	1	1	1
Worst	0.9596	1	0.9859	0.7924

Table 2(c). Results of the S for the benchmarks (with 30 independent runs).

S metric	MOP1	ZDT1	ZDT2	ZDT3
Mean	0.0463	0.0126	0.0138	0.0268
Standard deviation	0.0143	0.0026	0.0025	0.0102
Best	0.0272	0.0087	0.0108	0.0104
Worst	0.0969	0.229	0.0217	0.0533



(a) Pareto front of test function MOP2

(b) Pareto front of test function MOP1

(c) Pareto front of test function ZDT1

(d) Pareto front of test function ZDT2

(e) Pareto front of test function ZDT3

(f) Pareto front of test function DTLZ1

(g) Pareto front of test function DTLZ2

Figure 4. Pareto front of test functions: (a) MOP2, (b) MOP1, (c) ZDT1, (d) ZDT2, (e) ZDT3, (f) DTLZ1, and (g) DTLZ2.

In this table, results with Boldface indicate a better value. It can be seen that the MOCSS is one of the top best algorithms in finding optimum results.

5.4. Comparison study

In this section, the performance of the proposed MOCSS is compared with those of other established multi-objective algorithms, including Vector Evaluated Genetic Algorithm (VEGA), Non-dominated Sorting Genetic Algorithm II (NSGA-II), Multi-Objective Differential Evolution (MODE), Differential Evolution for Multi-Objective Optimization (DEMO), multi-objective Bees algorithms (Bees), Strength Pareto Evolutionary Algorithm (SPEA) and Multi-objective Firefly Algorithm (MOFA), Strength Pareto Evolutionary Algorithm (SPEA2), Multi-Objective Evolution Algorithms with the Tchebycheff approach (MOEA/D), Pareto Differential Evolution Approach (PDEA), NSGAII based on Differential Evolution (NSGAII-DE), the third version of Generalized Differential Evolution (GDE3), and Multi-Objective Differential Evolution-the Ranking-based Mutation Operator (MODE-RMO).

Table 3. Comparison of GD for $n = 50$ (charges particle) and iteration = 500, as presented in Refs. [30,37,25]. All results have been averaged over 30 independent runs. A result with Boldface indicates better value obtained.

Methods	Test function		
	ZDT1	**ZDT2**	**ZDT3**
Bees	2.40E-02	1.69E-02	1.91E-01
VEGA	3.79E-02	2.37E-03	3.29E-01
SPEA	1.78E-03	1.34E-03	4.75E-02
NSGA-II	3.33E-02	7.24E-02	1.14E-01
MODE	5.80E-03	5.50E-03	2.15E-02
DEMO	1.08E-03	7.55E-04	1.18E-03
MOFA	1.90E-04	1.52E-04	**1.97E-04**
PDEA	6.15E-04	6.52E-04	5.63E-04
SPEA2	3.40E-03	9.10E-03	1.80E-03
MOEA/D	9.59E-04	5.81E-04	1.73E-03
MODE-RMO	3.85E-03	6.97E-03	4.76E-03
GDE3	2.40E-03	8.20E-03	2.76E-03
NSGAII-DE	5.83E-03	7.75E-03	5.31E-03
MOCSS	**1.78 E-04**	**1.48 E-04**	7.94E-04

Table 4. Comparison between MOCSS and other algorithms on the basis of Wilcoxon's Ranksum test.

Methods	Test function		
	ZDT1	**ZDT2**	**ZDT3**
Bees	-1^*	-1	-1
VEGA	-1	-1	-1
SPEA	-1	-1	-1
NSGA-II	-1	-1	-1
MODE	-1	-1	-1
DEMO	-1	-1	-1
MOFA	-1	-1	$+1^{**}$
PDEA	-1	-1	0^{***}
SPEA2	-1	-1	-1
MOEA/D	-1	-1	-1
MODE-RMO	-1	-1	-1
GDE3	-1	-1	-1
NSGAII-DE	-1	-1	-1

$^*-1$: worse; $^{**}+1$: better; and $^{***}0$: equal.

The performances of the proposed multi-objective approach are evaluated and compared using the multi-objective metric in terms of GD (how far the known Pareto front is from the true Pareto front) with the above-mentioned MO approach given in Table 3. Overall, the MOCSS has better performance than the other thirteen cases, and it has the best convergence in ZDT1, ZDT2, and ZDT3 (except MOFA and PDEA) benchmark functions.

To check if the final results obtained with the best performing algorithm differ from the final results of the rest of the competing algorithms in a statistically significant manner, the Wilcoxon's Ranksum test for independent samples [61] is used at a significance level of 5%, as presented in Table 4. The numerical values of -1, 0, 1 correspond to whether the other methods are inferior to, equal to, and superior to our proposed algorithm, as indicated in Table 4.

The MOCSS is implemented and compared with NSGA-II and MOEA/D using DTLZ1 and DTLZ2 test functions in the coverage of two set metrics. Table 5 shows the coverage of two set metric values of the three approaches, averaged on 30 independent runs. A careful inspection of Tables 5 reveals that, in terms of coverage of two set metrics, the final solutions obtained by the MOCSS are better than those obtained by NSGA-II and MOEA/D for DTLZ1 and DTLZ2 test instances.

The Hypervolume indicator is employed to guide the diversity preservation in our approach. The reference points used for assessments are $r = 1.1^d$ and $r = 1.1^d$ for DTLZ1 and DTLZ2, respectively. Table 6 presents average and standard deviation relative hypervolume for MOCSS, NSGA-II, and SPEA2 approaches. A larger hypervolume value is preferable when comparing the performances of different solution sets. Therefore, the MOCSS approach performs significantly better than the other two approaches. The results of GD obtained by GDE3, MODE-RMO, NSGAII-DE, and MOCSS, besides Wilcoxon's Ranksum test, are presented in Table 7. According to Table 7, the MOCSS outperforms NSGAII-DE in 4 problems and performs evenly in 1 problem. In addition, it outperforms GDE3 in 3 problems and loses in 2 problems. It also outperforms MODE-RMO in 2 problems, loses in 2 problems, and performs evenly in 1 problems. Briefly, for the tri-objective test functions, the MOCSS has better GD value in DTLZ1, DTLZ2, and DTLZ3 than another approach.

Finally, times were also evaluated (using the same hardware platform and the exact same environment for each of the two algorithms) in order to establish if our MOCSS algorithm was really faster than the NSGA-II or not. Table 8 shows that NSGA-II covers the entire Pareto front and is faster in computational time by -5 to 13 percent (on average 4 percent) than the MOCSS

Table 5. Average coverage of two set metrics between MOCSS, NSGA-II, and MOPSO (pop. = 50 and independent runs = 30).

Approach	MOCSS		MOEA/D		NSGA-II	
	NSGA-II	MOEA/D	MOCSS	NSGA-II	MOCSS	MOEA/D
DTLZ1	0.091	0.051	0.018	0.078	0.011	0.005
DTLZ2	0.072	0.048	0.028	0.099	0.016	0.001

in this test function. However, the implementation of the MOCSS method produces excellent results, as shown in Table 3.

6. Engineering design problem

6.1. Welded beam design

The design of a welded beam is a classical benchmark that has been solved by many researchers. The welded beam design is a real-life application problem [11,62], whose aim is to minimize the cost and the endpoint's deflection subject to constraints on shear stress, bending stress, and buckling load (Figure 5). The detailed formulation can be found in [11,45,62,63].

Table 6. Average and standard deviation relative hypervolume among MOCSS, NSGA-II, and SPEA2 (independent runs = 30) [46].

Method	DTLZ1, $r = 0.7^d$	DTLZ2, $r = 1.1^d$
NSGA-II	0.94333 (0.11423)	0.86913(0.00803)
SPEA2	0.98010(0.00152)	0.90760(0.00350)
MOCSS	0.96201(0.01067)	0.92056(0.01538)

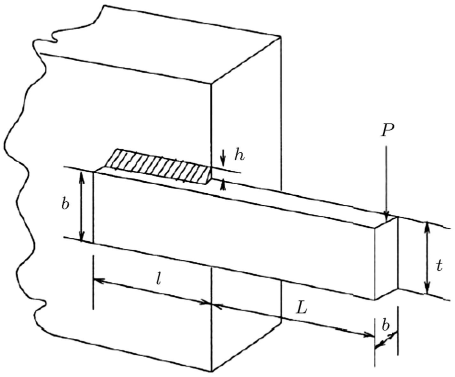


Figure 5. The welded beam design problem.

It is desired to find four design parameters (thickness b, width t, length of weld L, and weld thickness h) for which the cost function of the beam and the deflection function at the open end are objective functions [45]:

$$\min f_1(x) = 1.1047h^2L + 0.0481tb(14 + t),$$

$$\min f_2(x) = \delta(x) = \frac{2.1952}{t^3b}, \tag{22}$$

subject to:

Table 7. Comparison between MOCSS and other algorithms on average *GD* metrics and the basis of Wilcoxon's Ranksum test [30] (independent runs = 20 and pop. = 100).

Example	MOCSS	GDE3		MODE-RMO		NSGAII-DE	
DTLZ1	2.72E-04	7.08E-02	-1^*	2.58E-04	0^{***}	4.80E-03	-1
DTLZ2	4.59E-04	7.25E-04	-1	7.28E-04	-1	1.96E-03	-1
DTLZ3	6.59E-04	3.27E+00	-1	7.88E-03	-1	1.82E-01	-1
DTLZ4	5.24E-04	7.12E-04	$+1^{**}$	7.26E-04	$+1$	1.46E-03	-1
DTLZ5	1.93E-04	1.09E-05	$+1$	9.12E-06	$+1$	1.87E-04	0

$^*-1$: worse; $^{**}+1$: better; and $^{***}0$: equal.

Table 8. Average and Standard deviation computational time between MOCSS and NSGA-II (independent runs = 30).

Test function	MOCSS		NSGA-II	
	Pop.= 50 and Iter.=50	Pop.=100 and Iter.=100	Pop.=50 and Iter.=50	Pop.=100 and Iter.=100
MOP1	31.05 ± 0.31	218.75 ± 1.53	30.11 ± 0.56	217.19 ± 1.31
ZDT1	28.46 ± 0.28	192.46 ± 2.34	24.85 ± 1.13	187.67 ± 2.03
ZDT2	28.72 ± 0.27	192.23 ± 2.34	25.28 ± 0.99	186.16 ± 3.36
ZDT3	28.36 ± 0.34	191.35 ± 0.96	27.36 ± 1.01	202.53 ± 1.21

$$g_1(x) = \tau(x) - 13600 \leq 0,$$

$$g_2(x) = \sigma(x) - 30000 \leq 0,$$

$$g_3(x) = h - b \leq 0,$$

$$g_4(x) = 6000 - P(x) \leq 0,$$

where:

$$\sigma(x) = \frac{504000}{t^2 b},$$

$$P(x) = 64764.022(1 - 0.0282346t)tb^3,$$

$$D = \sqrt{0.25(L^2 + (h+t)^2)},$$

$$Q = 6000(14 + 0.5L),$$

$$\tau'(x) = \frac{6000}{\sqrt{2}hL},$$

$$\tau''(x) = \frac{QD}{2[0.707hL(\frac{L^2}{12} + D^2)]},$$

$$\tau(x) = \sqrt{(\tau'(x))^2 + (\tau''(x))^2 + \frac{L\tau'(x)\tau''(x)}{D}},$$

where the simple limits for variables are $0.1 \leq L, t \leq 10$ and $0.125 \leq h, b \leq 5$.

In the welded beam design problem, the nonlinear constraints can cause difficulties in finding the Pareto front. This design problem has been solved by the MOCSS. The Pareto front of 50 solution points after 1000 iterations is obtained by the MOCSS, as shown in Figure 6. The obtained results include distribution, spread, and smoothening, which are the same with or better than the results obtained in other researches [11,45,62].

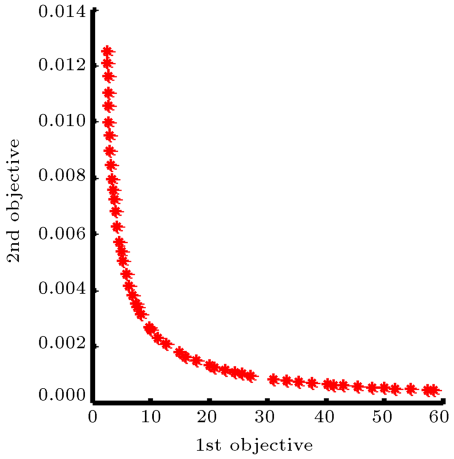


Figure 6. Pareto front for the bi-objective beam design where the horizontal axis corresponds to cost and the vertical axis corresponds to deflection.

6.2. *Design of a disc brake*

The multi-disc brake design problem is another benchmark for constrained, mixed, and multi-objective optimizations, studied by Osyczka and Kundu (1995) [64], Ray and Liew (2002) [62], and Gong et al. (2009) [11]. The objectives of the design include minimizing the overall mass of the brake and the braking time. The design variables include the inner radius of the discs, outer radius of the discs, the engaging force, and the number of friction surfaces, which are represented by r, R, F, and s, respectively. The constraints for the design include the minimum distance between the radii, maximum length of the brake, pressure, temperature, and torque limitations [62]:

$$\min f_1(x) = 4.9 \times 10^{-5}(R^2 - r^2)(s-1),$$

$$\min f_2(x) = \frac{9.82 \times 10^6(R^2 - r^2)}{Fs(R^3 - r^3)}, \tag{23}$$

subject to:

$$g_1(x) = 20 - (R - r) \leq 0,$$

$$g_2(x) = 2.5(s+1) - 30 \leq 0,$$

$$g_3(x) = \frac{F}{3.14(R^2 - r^2)} - 0.4 \leq 0,$$

$$g_4(x) = \frac{0.00222F(R^3 - r^3)}{(R^2 - r^2)^2} - 1 \leq 0,$$

$$g_5(x) = 900 - \frac{0.0266Fs(R^3 - r^3)}{(R^2 - r^2)} \leq 0,$$

where the simple limits for variables are $55 \leq r \leq 80$, $75 \leq R \leq 110$, $1000 \leq F \leq 3000$, and $2 \leq s \leq 20$.

In the disc break design problem, the non-linear constraints can cause difficulties in finding the Pareto front. This design problem has been solved using the MOCSS. The Pareto front of 50 solution points after 1000 iterations obtained by the MOCSS is shown in Figure 7.

7. Conclusions

In this paper, a new algorithm was formulated successfully for multi-objective optimization, namely multi-objective charged system search, based on the recently developed single-objective charged system search optimization algorithm. To obtain a good convergence to the Pareto front for an algorithm, a Non-dominated Sorting (NS) mechanism was used; to prevent early convergence, a mutation function was utilized, too. The proposed MOCSS was tested against a set of well-chosen test functions. The comparison of the

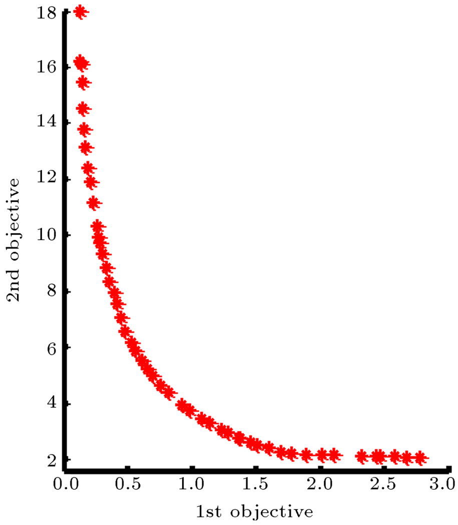


Figure 7. Final optimum result of the disc brake designing example.

GD metric results can be used as a yardstick to conclude that the MOCSS has better performance than others and has the best convergence in ZDT1, ZDT2, and ZDT3 (expect MOFA and PDEA) benchmark functions. The simulations for the benchmark and test functions suggest that the MOCCS is a very efficient algorithm for multi-objective optimization. To check if the final results obtained by the best-performing algorithm differ from the final results of the rest of the competing algorithms in a statistically significant manner, the Wilcoxon's Ranksum test for independent samples was used at a significance level of 5%. It outperformed all the contestant algorithms in a statistically significant manner.

In the disc break design problem and the welded beam design problem, both the non-linear constraints can cause difficulties in finding the Pareto front. These design problems were solved by the MOCSS. The obtained results included distribution, spread, and smoothening, which are the same with or better than the results obtained in other researches. The MOCSS can deal with highly nonlinear problems with complex constraints and diverse Pareto optimal sets.

As for future works, the formulation of a discrete MOCSS will be an important topic. In addition, hybridization with other algorithms may also be fruitful. Further, the possibility of extending this algorithm for dynamic functions may be considered.

References

1. Coello, C.C., Lamont, G.B., and Van Veldhuizen, D.A., *Evolutionary Algorithms for Solving Multi-Objective Problems*, Second Edition ed., Springer Science & Business Media, New York (2007).

2. Xiang, Y. and Zhou, Y. "A dynamic multi-colony artificial bee colony algorithm for multi-objective optimization", *Applied Soft Computing*, **35**, pp. 766-785 (2015).

3. Yang, G.-Q., Liu, Y.-K., and Yang, K. "Multi-objective biogeography-based optimization for supply chain network design under uncertainty", *Computers & Industrial Engineering*, **85**, pp. 145-156 (2015).

4. Lark, R.M. "Multi-objective optimization of spatial sampling", *Spatial Statistics*, **18**, pp. 412-430 (2016).

5. Akay, B. and Karaboga, D. "A survey on the applications of artificial bee colony in signal, image, and video processing", *Signal, Image and Video Processing*, **9**(4), pp. 967-990 (2015).

6. Nseef, S.K., Abdullah, S., Turky, A., and Kendall, G. "An adaptive multi-population artificial bee colony algorithm for dynamic optimisation problems", *Knowledge-Based Systems*, **104**, pp. 14-23 (2016).

7. Kaveh, A., Laknejadi, K., and Alinejad, B. "Performance-based multi-objective optimization of large steel structures", *Acta Mechanica*, **223**(2), pp. 355-369 (2012).

8. Osyczka, A. "Multicriteria optimization for engineering design", *Design Optimization*, **1**, pp. 193-227 (1985).

9. Kaveh, A. and Laknejadi, K. "A hybrid evolutionary graph based multi-objective algorithm for layout optimization of truss structures", *Acta Mechanica*, **224**(2), pp. 343-364 (2013).

10. Coello, C.A.C. "An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends", in: *Proceedings of the Congress on Evolutionary Computation*, pp. 3-13 (1999).

11. Gong, W., Cai, Z., and Zhu, L. "An effective multi-objective differential evolution algorithm for engineering design", *Structural and Multidisciplinary Optimization*, **38**(2), pp. 137-157 (2009).

12. Pareto, V., *Cours d'économie Politique*, Librairie Droz (1964).

13. Kishor, A., Singh, P.K., and Prakash, J. "NSABC: Non-dominated sorting based multi-objective artificial bee colony algorithm and its application in data clustering", *Neurocomputing*, **216**, pp. 514-533 (2016).

14. Li, T., Sun, X., Lu, Z., and Wu, Y. "A novel multiobjective optimization method based on sensitivity analysis", *Mathematical Problems in Engineering*, **2016** (2016).

15. Chiong, R., *Nature-Inspired Algorithms for Optimisation*, Springer-Verlag, Berlin Heidelberg (2009).

16. Miettinen, K., *Nonlinear Multiobjective Optimization*, MA: Kluwer Academic Publishers, Boston (1999).

17. Yang, X.-S., Karamanoglu, M., and He, X. "Multi-objective flower algorithm for optimization", *Procedia Computer Science*, **18**, pp. 861-868 (2013).

18. Schaffer, J.D. "Multiple objective optimization with vector evaluated genetic algorithms", in: *Proceedings*

of the 1st international Conference on Genetic Algorithms and their Applications, L. Erlbaum Associates Inc., USA, pp. 93-100 (1985).

19. Coello, C.A.C.C. and Pulido, G.T. "A micro-genetic algorithm for multiobjective optimization", in: E. Zitzler, K. Deb, L. Thiele, C.A.C. Coello, and D. Corne, Eds., *First International Conference on Evolutionary Multi-Criterion Optimization*, Springer-Verlag, pp. 126-140 (2001).

20. Srinvas, N. and Deb, K. "Multi-objective function optimization using non-dominated sorting genetic algorithms", *Evolutionary Computation*, **2**(3), pp. 221-248 (1994).

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

22. Zitzler, E. and Thiele, L. "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach", *IEEE Transactions on Evolutionary Computation*, **3**(4), pp. 257-271 (1999).

23. Zitzler, E., Laumanns, M., and Thiele, L., *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*, in Swiss Federal Institute Technology, Zurich, Switzerland, pp. 95-100 (2001).

24. Knowles, J.D. and Corne, D.W. "Approximating the nondominated front using the Pareto archived evolution strategy", *Evolutionary Computation*, **8**(2), pp. 149-172 (2000).

25. Madavan, N.K. "Multiobjective optimization using a Pareto differential evolution approach", in: *Congress on Evolutionary Computation (CEC'2002)*, New Jersey, pp. 1145-1150 (2002).

26. Zhang, Q. and Li, H. "MOEA/D: A multiobjective evolutionary algorithm based on decomposition", *IEEE Transactions on Evolutionary Computation*, **11**(6), pp. 712-731 (2007).

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

28. Kaveh, A. and Laknejadi, K. "A hybrid multi-objective particle swarm optimization and decision making procedure for optimal design of truss structures", *Iranian Journal of Science and Technology*, **35**(C2), pp. 137-154 (2011).

29. Kukkonen, S. and Lampinen, J. "GDE3: The third evolution step of generalized differential evolution", in: *2005 IEEE Congress on Evolutionary Computation, IEEE*, pp. 443-450 (2005).

30. Chen, X., Du, W., and Qian, F. "Multi-objective differential evolution with ranking-based mutation operator and its application in chemical process optimization", *Chemometrics and Intelligent Laboratory Systems*, **136**, pp. 85-96 (2014).

31. Coello, C.A.C. and Lechuga, M.S. "A proposal for multiple objective particle swarm optimization", in: *Proceedings of the Congress on Evolutionary Computation (CEC'2002)*, pp. 1051-1056 (2002).

32. Robič, T. and Filipič, B. "DEMO: Differential evolution for multiobjective optimization", in: *International Conference on Evolutionary Multi-Criterion Optimization*, Springer, pp. 520-533 (2005).

33. Kaveh, A. and Laknejadi, K. "A novel hybrid charge system search and particle swarm optimization method for multi-objective optimization", *Expert Systems with Applications*, **38**(12), pp. 15475-15488 (2011).

34. Babu, B. and Gujarathil, A.M. "Multi-objective differential evolution (MODE) for optimization of supply chain planning and management", in: *2007 IEEE Congress on Evolutionary Computation, IEEE*, pp. 2732-2739 (2007).

35. Pham, D. and Ghanbarzadeh, A. "Multi-objective optimisation using the bees algorithm", in: *3rd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS 2007),* Whittles, Dunbeath, Scotland, pp. 111-116 (2007).

36. Jadaan, O.A., Rajamani, L., and Rao, C. "Non-dominated ranked genetic algorithm for solving constrained multi-objective optimization problems", *Journal of Theoretical & Applied Information Technology*, **5**(5), pp. 640-651 (2009).

37. Yang, X.-S. and Deb, S. "Multiobjective cuckoo search for design optimization", *Computers & Operations Research*, **40**(6), pp. 1616-1624 (2013).

38. Yang, X.-S. "Multiobjective firefly algorithm for continuous optimization", *Engineering with Computers*, **29**(2), pp. 175-184 (2013).

39. Kaveh, A. and Laknejadi, K. "A new multi-swarm multi-objective optimization method for structural design", *Advances in Engineering Software*, **58**, pp. 54-69 (2013).

40. Kaveh, A. and Laknejadi, K. "A swarm based memetic evolutionary algorithm for multi-objective optimization of large structures", *Asian Journal of Civil Engineering*, **16**(5), pp. 621-649 (2015).

41. Erfani, T. and Sergei, V.U. "Directed search domain: a method for even generation of the Pareto frontier in multiobjective optimization", *Engineering Optimization*, **43**(5), pp. 467-484 (2011).

42. Kaveh, A. and Massoudi, M.S. "Multi objective Optimization of structures using charged system search", *Scientia Iranica*, **21**(6), pp. 1845-1860 (2014).

43. Kaveh, A. and Talatahari, S. "A novel heuristic optimization method: charged system search", *Acta Mechanica*, **213**(3-4), pp. 267-289 (2010).

44. Kaveh, A. and Talatahari, S. "Charged system search for optimal design of frame structures", *Applied Soft Computing*, **12**(1), pp. 382-393 (2012).

45. El-Sawy, A.A., Hussein, M.A., Zaki, E.-S.M., and Mousa, A.A.A. "Local search-inspired rough sets for improving multiobjective evolutionary algorithm", *Applied Mathematics*, **5**(13), pp. 1993-2007 (2014).

46. Wagner, T., Beume, N., and Naujoks, B. "Pareto-, aggregation-, and indicator-based methods in many-objective optimization", in: *4th International Conference on Evolutionary Multi-Criterion Optimization*, Springer, Japan, pp. 742-756 (2007).

47. Luo, J., Liu, Q., Yang, Y., Li, X., Chen, M.-R., and Cao, W. "An artificial bee colony algorithm for multi-objective optimisation", *Applied Soft Computing*, **50**, pp. 235-251 (2017).

48. Oyama, A., Shimoyama, K., and Fujii, K. "New constraint-handling method for multi-objective and multi-constraint evolutionary optimization", *Transactions of the Japan Society for Aeronautical and Space Sciences*, **50**(167), pp. 56-62 (2007).

49. Van Veldhuizen, D.A., *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*, in: Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, DTIC Document, Ohio (1999).

50. Neema, M.N. and Ohgai, A. "Multi-objective location modeling of urban parks and open spaces: Continuous optimization", *Computers, Environment and Urban Systems*, **34**(5), pp. 359-376 (2010).

51. Deb, K., *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, New York (2001).

52. Goldberg, D.E. and Richardson, J. "Genetic algorithms with sharing for multimodal function optimization", in: *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, Hillsdale, NJ: Lawrence Erlbaum, Mahwah, pp. 41-49 (1987).

53. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., and Tiwari, S. "Multiobjective optimization test instances for the CEC 2009 special session and competition", in: University of Essex, Colchester, UK and Nanyang Technological University, Singapore, Special Session on Performance Assessment of Multi-Objective Optimization Algorithms, Technical Report (2008).

54. Huo, J. and Liu, L. "An improved multi-objective artificial bee colony optimization algorithm with regulation operators", *Information*, **8**(1), p. 18 (2017).

55. Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. "Scalable test problems for evolutionary multiobjective optimization", in: A. Ajith and G. Robert, Eds., *Evolutionary Multiobjective Optimization, Theoretical Advances and Applications*, Springer, USA, pp. 105-145 (2005).

56. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., and Da Fonseca, V.G. "Performance assessment of multiobjective optimizers: an analysis and review", *IEEE Transactions on Evolutionary Computation*, **7**(2), pp. 117-132 (2003).

57. Zitzler, E., Deb, K., and Thiele, L. "Comparison of multiobjective evolutionary algorithms: Empirical results", *Evolutionary Computation*, **8**(2), pp. 173-195 (2000).

58. Van Veldhuizen, D.A. and Lamont, G.B., *Multiobjective Evolutionary Algorithm Research: A History and Analysis*, in: Citeseer, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology (1998).

59. Schott, J.R., *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization*, in: Department of Aeronautics and Astronautics, DTIC Document, Cambridge (1995).

60. Li, K., Kwong, S., Cao, J., Li, M., Zheng, J., and Shen, R. "Achieving balance between proximity and diversity in multi-objective evolutionary algorithm", *Information Sciences*, **182**(1), pp. 220-242 (2012).

61. Wilcoxon, F. "Individual comparisons by ranking methods", *Biometrics Bulletin*, **1**(6), pp. 80-83 (1945).

62. Ray, T. and Liew, K.M. "A swarm metaphor for multiobjective design optimization", *Engineering Optimization*, **34**(2), pp. 141-153 (2002).

63. Deb, K. "Evolutionary Multi-Criterion Optimization", in: K. Miettinen, P. Neittaanmäki, M.M. Mäkelä and J. Périaux, Eds., *Evolutionary Algorithms in Engineering and Computer Science*, pp. 135-161 (2004).

64. Osyczka, A. and Kundu, S. "A genetic algorithm-based multicriteria optimization method", in: *Proceedings 1st World Congress Structural Multidisciplinary Optimization*, pp. 909-914 (1995).

Biographies

Abolfazl Ranjbar is a lecturer of surveying in University of Tabriz. He is a Ph.D. Candidate in Department of GIS Engineering at Tehran University. Abolfazl Ranjbar research interests and published papers are in the domains of Location-Allocation, optimization, spatial databases and spatial optimization.

Siamak Talatahari is an Associate Professor of Structural Engineering in University of Tabriz. Dr. Talatahari is recognized as "Elite" by Iranian Elites Organization and as the "Distinguished Researcher" in 2010. He is the author of more than 80papers published in international journals and more than 20 other papers presented at international conferences. He is also the editor of two international books which will be published by "Elsevier" in the end of 2012.

F. Hakimpour received his PhD in Geographic Information Science from University of Zurich in 2003, his MSc from ITC, the Netherlands, and his BSc in Computer Science from University of Tehran. He is an

Assistant Professor and a member of GIS Department at the School of Surveying and Geospatial Engineering at University of Tehran. He is also vice principal for research and graduate studies at the same school. Dr. Hakimpour is the author of more than 10 papers published in international journals and several papers presented at international conferences. His research interests and published papers are in the domains of spatial Web, semantic Web, spatial databases, modelling movement and spatial optimization.