Application of Multi-objective Charged System Search Algorithm for Optimization Problems

S. Talatahari[*,1,2], F. Hakimpour[3], A. Ranjbar[3]

[1] *Department of Civil Engineering, University of Tabriz, Tabriz, Iran*
[2] *Engineering Faculty, Near East University, 99138 Nicosia, North Cyprus, Mersin 10, Turkey*
[3] *School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran, Iran* fhakimpour@ut.ac.ir abranjbar57@ut.ac.ir

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 are used to evaluate the proposed algorithm and the results have been compared with other new approaches. The results of implementing the new algorithm on some test problems show that the proposed algorithm outperforms the other algorithms in terms of Generational Distance, Maximum Spread, Spacing, Coverage of two Set and Hypervolume Indicator. Results of well-known mathematical examples indicate that the new approach is highly competitive and can be considered as a viable alternative to solve 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.

Keywords: Charged System Search, Meta-heuristics, Multi-objective Optimization, Pareto optimal, and Multi-Objective Charged System Search.

[*] Corresponding author: S. Talatahari, Department of Civil Engineering, University of Tabriz, Tabriz, Iran and Engineering Faculty, Near East University, 99138 Nicosia, North Cyprus, Mersin 10, Turkey.
E-mail address: siamak.talat@gmail.com, talatahari@tabrizu.ac.ir (S. Talatahari)
Tel: +98 914 4041564

The Application of Multi-objective Charged System Search Algorithm for Optimization Problems

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 are used to evaluate the proposed algorithm and the results have been compared with other new approaches. The results of implementing an algorithm on some test problems show that the proposed algorithm outperforms the other algorithms in terms of Generational Distance, Maximum Spread, Spacing, Coverage of two Set 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 solve 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.

Keywords: Charged System Search, Meta-heuristics, Multi-objective Optimization, Pareto optimal, and Multi-Objective Charged System Search.

1. Introduction

Many realistic problems contain simultaneous optimization of several objectives in which these objectives may conflict with each other and with other nonlinear constraints, if exist [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 a 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 are:

1) In single-objective optimization, the fitness of solutions is reachable easily due to existing just one objective function while for multi-objective optimization, no single unique solution can be determined as the best; instead, a set of non-dominated solutions should be found in order to get a good approximation of the true Pareto fronts [3, 12-14], which leads to trade-off among the objectives [15, 16] .

2) Algorithms which work well for single objective problems usually cannot directly be used for multi-objective ones, and it is necessary to consider some special conditions. As a simple way, one can combine multi-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 uniformly 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 Multiobjective 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 charge 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]. Recent years, some other types of algorithms were also 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].

Without considering the number of these methods and their differences, they share some defections as follows:

I. The final distribution of Pareto points often is not well-spread; therefore maximum information on the Pareto cannot often obtain [13, 41].
II. Finding results often need a heavy computation and are time–consuming.

So developing a new multi-objective optimization method to resolve some of these drawbacks seems necessary.

A hybrid charged system search and particle swarm multi-objective optimization is due to Kaveh and Laknejadi (2011) [33], where the answers space was divided based to some spaces in order to find a uniform Pareto points. A multi-objective charged system search has been developed by Kaveh and Massoudi[42], These algorithm have extended of the single objective charged system search (CSS), 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 non-dominated method has been 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 details. The fundamental concept of utilized constrains handling method for MOCSS in detail described in section 4. The benchmarks 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 on other electrically charged objects. The Coulomb law determines electric field surrounding a point charge. Its value is proportion with the product of two charges particles and inversely square of the separation distance between the particles directed along the line. By Gauss law, the magnitude of the electric field at a point inside a charged sphere can be determined (proportion 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) which act as a charged sphere and can apply electrical forces on the other CPs. The resultant force acts on each CP creating acceleration according to Newton's second law. Finally, 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 to calculating the electrical force between two CPs is as follows:

$$\mathbf{F}_j = q_j \sum_{i,i \neq j} \left(\frac{q_i}{a^3} r_{ij} . i_1 + \frac{q_i}{{r_{ij}}^2} . i_2 \right) p_{ij} (\mathbf{X}_i - \mathbf{X}_j), \qquad \begin{cases} j = 1, 2, ..., n \\ i_1 = 1, i_2 = 0 \Leftrightarrow r_{ij} < a \\ i_1 = 0, i_2 = 1 \Leftrightarrow r_{ij} \geq a \end{cases} \tag{1}$$

where

a , The radius of the charged sphere

n is the total number of CPs

Where $\mathbf{F}_j$ is the resultant force acting on the j th 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 $\mathbf{X}_j$ are the positions of the ith and j th CPs

The initial position of CPs is obtained by equation (2) in the search space:

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

where $x_{i,j}$ determines the initial value of the ith variable for the j th 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 in 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 so far best and the worst fitness of all particles; $fit(i)$ represents the objective function value or the fitness of the agent i; and n is the total number of CPs. In equation (4), the kind of force is attractive in which all good CPs can attract bad CPs and only some of bad agents attract good agents, due to provide appropriate exploitation and exploration abilities, as:

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

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

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

where $\mathbf{X}_i$ and $\mathbf{X}_j$ are the positions of the ith and j th 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:

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

Also 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:

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

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

Where *iter* is the actual iteration number 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 and Non-dominate Sorting (NS) for good convergence and high diversity of Pareto front [45], respectively. NS sorts the solutions on the base of Non-dominate and then forms the levels of Pareto fronts. For selecting the numbers of the best solution, the solutions with highest Pareto front rank are chosen and if needed the other solutions are selected for the next Pareto front. This process is repeated until the Crowding Distance (CD) condition is satisfied (according to Fig. 1). In this article utilized the mutation function from GA in order to prevent early convergence. The following pseudo-code summarized the MOCSS algorithm:

Level 1: Initialization

Step 1: Initialize Specification of optimization problem, and algorithm parameters

Step 2: Initialize the initial positions of Charged Particles and their associated velocities

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 Fig. 2.

4. Constrain Handling method for the MOCSS

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

Find $\mathbf{x}$ that minimize

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

Subject to

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

Where $\mathbf{x} = (x_1, ..., x_{nVar})$ is the vector of solution that minimizes objective function(s) $F(\mathbf{x})$ while satisfying the constraint(s) $G(\mathbf{x}) \leq 0$. The numbers of design parameter(s), objective function(s) and constraint(s) are $nVar$, 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 for 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, 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 balance between the objective(s) and constraint(s). In this research, was used 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 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 as follows.

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 the solution $\mathbf{x}_i$ is said to dominate solution $\mathbf{x}_j$ if $f_k(\mathbf{x}_i)$ is no worse than $f_k(\mathbf{x}_j)$ for the all objectives and it is better for at least one of them, [21, 49] :

$$f_k(\mathbf{x}_i) \leq f_k(\mathbf{x}_j),\ \forall i = 1, 2, ..., k \text{ and } f_k(\mathbf{x}_i) < f_k(\mathbf{x}_j),\ \exists i \in \{1, 2, ..., k\} \quad (11)$$

So, 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 acquisition more detail about 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, but solution *i* dominates solution *j* in the constraint space.

Definition 2 (Constraint space dominance): Solution *i* is said to dominate solution *j* in the constraint space if both of the following conditions are true,

1. Solutions *i* is no worse than solution *j* in all constraints, i.e.,

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

and

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

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

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

$$\begin{aligned}
&rank'(\mathbf{x}_i) = rank(\mathbf{x}_i) \times \mathrm{Penalty}(\mathbf{x}_i) \\
&\mathrm{Penalty}(\mathbf{x}_i) = 1 + \sum_{j=1, j \ne i}^{npop} sh_{ij} \\
&sh_{ij} = \begin{cases} 1 - \left(\dfrac{d_{ij}}{\sigma_{share}} \right)^{\alpha} & \mathrm{d}_{ij} < \sigma_{share} \\ 0 & \mathrm{d}_{ij} \ge \sigma_{share} \end{cases} \\
&\sigma_{share} = \sum_{n=1}^{h} (g\max_n - g\min_n) / npop \\
&d_{ij} = \sqrt{\sum_{n=1}^{h} (\mathrm{g}_n(\mathbf{x}_i) - \mathrm{g}_n(\mathbf{x}_j))^2} \\
&g\max_n = \max(\mathrm{g}_n(\mathbf{x}_1), .., \mathrm{g}_n(\mathbf{x}_{npop})) \\
&g\min_n = \min(\mathrm{g}_n(\mathbf{x}_1), .., \mathrm{g}_n(\mathbf{x}_{npop}))
\end{aligned} \tag{15}$$

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]),discontinuous Pareto fronts with more complex Pareto sets 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 comparison the quantitative of 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; and

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

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

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

I. Generational Distance (GD)

The concept of generational distance was introduced by Van Veldhuizen and Lamont [13, 54, 58] as a way of estimating how far the elements are in the set of generated Pareto front so far from those in the Pareto front true and is defined as:

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \qquad (16)$$

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 elements generated are in the true Pareto front i.e. $PF_{true} = PF_{known}$. Therefore, any other value will indicate how "far" we are from the global Pareto front of our problem. This metric need to know PF_{true}.

II. Maximum Spread (MS)

The metric of maximum spread (MS) measures how "well" the PF_{true} is covered by the PF_{known} through hyper-boxes formed by the extreme function values observed in the PF_{true} and PF_{known}. It is defined as:

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

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 *i*th objective in PF_{known} and PF_{true}, respectively. A larger value of MS implies a better spread of solutions. In this study $F_i^{\max}$ and $F_i^{\min}$ are considered as the maximum and minimum of the *i*th objective in all the Pareto fronts obtained by various algorithms [49]. This metric need to know PF_{true}.

III. Spacing (S)

The spacing (S) metric numerically describes the spread of the vectors in PF_{known} [33, 59]. This Pareto front metric measures the distance variance of neighboring vectors in PF_{known}. Equations (18) and (19) define this metric:

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

$$d_i = \min_j \left(\left| f_1^i(\vec{x}) - f_1^j(\vec{x}) \right| + \left| f_2^i(\vec{x}) - f_2^j(\vec{x}) \right| \right) \quad ,i, j = 1,2,...,n,\ i \neq j \quad (19)$$

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 dominance relationship between two populations resulting from two different MOEAs, Zitzler et al. [2003] propose the CS[56], that is measured to show how the final population of one algorithm dominates the final population of another algorithm. Equations (20) define this metric:

$$CS(X', X'') = \frac{\left|\{a'' \in X''; \exists a' \in X' : a' \leq a''\}\right|}{\left|X''\right|} \tag{20}$$

Where X' and X'' are two sets of solutions resulting from different algorithms. Where $a' \leq a''$ means that a' dominate 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 $CS(X', X'') = 1$. Also $CS(X', X'') = 0$ implies that none of the solutions in X'' are dominate by. Note that both $CS(X', X'')$ and $CS(X'', X')$ need to be considered independently since they have the distinct meanings, since $CS(X', X'')$ is not necessarily equal to $1 - CS(X'', X')$. The advantage of this Pareto compliant metric is that it is easy to calculate and provides a relative comparison based upon dominance numbers between two MOEAs [57].

V. Hypervolume Indicator of set S

Let the reference point is denoted as $\mathrm{Re}f = (r_1, r_2, ..., r_k)$, 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.

$$Hv(S) = Leb(\bigcup_{\vec{x} \in S} \left|f_1(\vec{x}_1), \mathrm{r}_1\right| \times \left|f_2(\vec{x}_2), \mathrm{r}_2\right| \times ... \times \left|f_k(\vec{x}_k), \mathrm{r}_k\right|)) \tag{21}$$

Where k is the number of dimension, $Leb(S)$ indicates the Lebesgue measure of S, and $\left|f_1(\vec{x}_1), \mathrm{r}_1\right| \times \left|f_2(\vec{x}_2), \mathrm{r}_2\right| \times ... \times \left|f_k(\vec{x}_k), \mathrm{r}_k\right|$ represents the hypercube formed by points which are dominated by $\vec{x}$ but Ref [60].

5.3. Numerical Results

For the MOP 1 example, Fig. 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 Fig. 4. In all 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 solutions properly for the benchmark examples. Also, the solutions are scattered among the Pareto Front identically. Therefore, this algorithm is able to correctly obtain the Pareto Front.

A careful scrutiny of Fig. 4 indicates that, the proposed MOCSS algorithm finding solutions has outperformed all benchmarks. Find the answers in addition to being close to the true Pareto Front are a uniform dispersion on it. The performances of the proposed multi-objective approach are evaluated using the multi-objective metrics in term of GD, S and MS and the results are summarized in Table 2, (including Mean, Std. Dev., Best and Worst with 30 independent runs). In all benchmark functions, the values of GD and S are close to zero and the value MS is close to one. This means that the result of MS metric shows that the MOCSS approach has the better covered the PF_{known} and also the result of S metric shows that MOCSS approach has the better spread of answer.

In order to evaluate the performances of the MOCSS with 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 also presented in Table 3. All results have been averaged over 30 independent runs. In this table, results with Boldface indicate better value. It can be seen that an algorithm is one of top best algorithms in finding optimum results.

5.4. Comparison Study

In this section, the performance of the proposed MOCSS is compared with 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), Multiobjective 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).

The performances of the proposed multi-objective approach are evaluated and compared using the multi-objective metric in term of GD (how far the known Pareto front is from the true Pareto front) with above mentioned MO approach given in Table 3. In a total view, MOCSS has better performances than other thirteen cases, and it has the best convergence in the 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 rest of the competing algorithms in a statistically significant manner, the Wilcoxon's Ranksum test for independent samples [61] is used at 5% significance level, as presented in Table 4. The numerical values -1, 0, 1 correspond to whether the other methods are inferior to, equal to, and superior to our proposed algorithm, as indicated in Tables 4.

Implement the MOCSS and compare it with respect to the NSGA-II and MOEA/D using the DTLZ1 and DTLZ2 test functions in Coverage of two Set metrics. Table 5 shows the Coverage of two Set e 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 MOCSS is 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. In the Table of 6 are presented Average and Std. Dev. relative Hypervolume for MOCSS, NSGA-II and SPEA2 Approach. A larger Hypervolume value is preferable when comparing the performances of different solutions set. So, MOCSS approach performing significantly better on two approaches. The results of GD obtained by GDE3, MODE-RMO, NSGAII-DE and MOCSS and besides Wilcoxon's Ranksum test are presented in Table 7. From Table 7, MOCSS outperforms NSGAII-DE in 4 problems and ties in 1 problem. Also, MOCSS outperforms GDE3 in 3 problems and loses in 2 problems and has outperformed MODE-RMO in 2 problems, loses in 2 problems and ties in 1 problems. Briefly, for the tri-objective test functions, 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. Table 8 shows NSGA-II covers the entire Pareto front and is -5 to 13 percent (on average 4 percent) computational time faster than the MOCSS in this test functions. However, the implementation of the MOCSS method gives results far best showing (see Table 3).

6. Engineering Design Problem

6.1. Welded Beam Design

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

$$\begin{aligned} &\min\ f_1(x) = 1.1047h^2L + 0.0481tb(14+t) \\ &\min\ f_2(x) = \delta(x) = \frac{2.1952}{t^3 b} \end{aligned} \qquad (22)$$

subject to

$$\begin{aligned} &g_1(x) = \tau(x) - 13600 \le 0 \\ &g_2(x) = \sigma(x) - 30000 \le 0 \\ &g_3(x) = h - b \le 0 \\ &g_4(x) = 6000 - P(x) \le 0 \end{aligned}$$

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 $0.1 \le L,t \le 10$ and $0.125 \le h,b \le 5$.

In the welded beam 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 is obtained by MOCSS as shown in Fig. 6. The obtained results are distribution, spread and smooth which are the same or better than the results obtained in other researches [11, 45, 62].

6.2. Design of a disc brake

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

$$\min\ f_1(x) = 4.9\times10^{-5}(R^2 - r^2)(s-1)$$
$$\min\ f_2(x) = \frac{9.82\times10^{6}(R^2 - r^2)}{Fs(R^3 - r^3)} \qquad (23)$$

subject to

$$g_1(x) = 20 - (R - r) \le 0$$
$$g_2(x) = 2.5(s+1) - 30 \le 0$$
$$g_3(x) = \frac{F}{3.14(R^2 - r^2)} - 0.4 \le 0$$
$$g_4(x) = \frac{0.00222F(R^3 - r^3)}{(R^2 - r^2)^2} - 1 \le 0$$
$$g_5(x) = 900 - \frac{0.0266Fs(R^3 - r^3)}{(R^2 - r^2)} \le 0$$

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

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 MOCSS is shown in Fig.7.

7. Conclusions

In this paper, a new algorithm successfully is formulated 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 the Pareto front for an algorithm, is used a Non-dominate Sorting (NS) mechanism and for prevent of early convergence, a mutation function is utilized, as well. The proposed MOCSS has been tested against a set of well-chosen test functions. Comparing the GD metric results can be concluded that, MOCSS has better performances than other cases, and it has the best convergence in the ZDT1, ZDT2 and ZDT3 (expect MOFA and PDEA) benchmark functions. The simulations for the benchmark and test functions suggest that MOCCS is a very efficient algorithm for multi-objective optimization. To check if the final results obtained with the best performing algorithm differ from the final results of rest of the competing algorithms in a statistically significant manner, the Wilcoxon's Ranksum test for independent samples is used at 5% significance level. 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 have been solved using the MOCSS. The obtained results are distribution, spread and smooth which are the same or better than the results obtained in other researches. It can deal with highly nonlinear problems with complex constraints and diverse Pareto optimal sets.

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

Reference

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 %@ 1568-4946 (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 %@ 0360-8352 (2015).
4. Lark, R.M. "Multi-objective optimization of spatial sampling", Spatial Statistics, 18, pp. 412-430 %@ 2211-6753 (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, pp. 967-990 %@ 1863-1703 (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 %@ 0950-7051 (2016).
7. Kaveh, A., Laknejadi, K. and Alinejad, B. "Performance-based multi-objective optimization of large steel structures", Acta Mechanica, 223, 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, 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, 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 %@ 0925-2312 (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 %@ 1024-123X, pp. (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, 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, 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, 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, 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, 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, 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, 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, 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, pp. 640-651 (2009).
37. Yang, X.-S. and Deb, S. "Multiobjective cuckoo search for design optimization", Computers & Operations Research, 40, pp. 1616-1624 (2013).

38. Yang, X.-S. "Multiobjective firefly algorithm for continuous optimization", Engineering with Computers, 29, 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, 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, pp. 467-484 (2011).
42. Kaveh, A. and Massoudi, M.S. "Multi objective Optimization of structures using charged system search", Scientia Iranica, 21, pp. 1845-1860 (2014).
43. Kaveh, A. and Talatahari, S. "A novel heuristic optimization method: charged system search", Acta Mechanica, 213, pp. 267-289 (2010).
44. Kaveh, A. and Talatahari, S. "Charged system search for optimal design of frame structures", Applied Soft Computing, 12, 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, 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 %@ 1568-4946 (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, pp. 56-62 (2007).
49. Van Veldhuizen, D.A. "Multiobjective evolutionary algorithms: classifications, analyses, and new innovations", in: Department of Electrical and Cumputer Engineering, Graduate Shcool 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, 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, pp. 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, pp. 117-132 (2003).
57. Zitzler, E., Deb, K. and Thiele, L. "Comparison of multiobjective evolutionary algorithms: Empirical results", Evolutionary computation, 8, 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, pp. 220-242 (2012).
61. Wilcoxon, F. "Individual comparisons by ranking methods", Biometrics bulletin, 1, pp. 80-83 (1945).
62. Ray, T. and Liew, K.M. "A swarm metaphor for multiobjective design optimization", Engineering optimization, 34, 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).

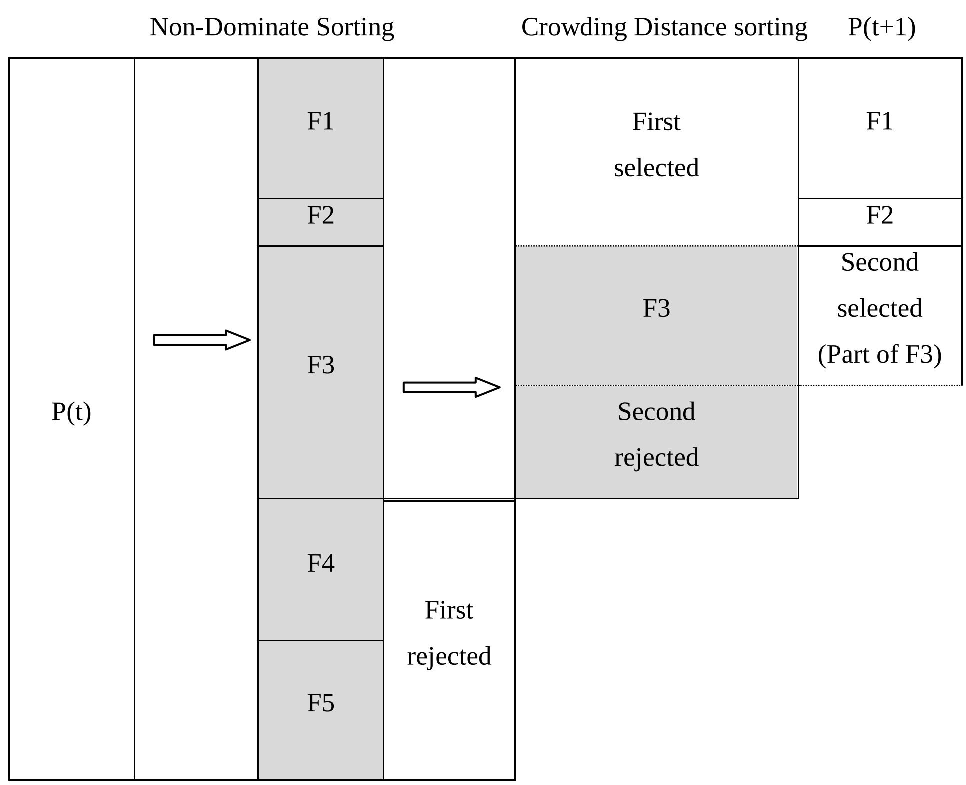

Fig. 1. Flow diagram that shows the way in which the NS works. P(t) is the population at generation t. F1 are the best solution. F2 are the second best solutions and so on. (Coello et al., p94, 2007)

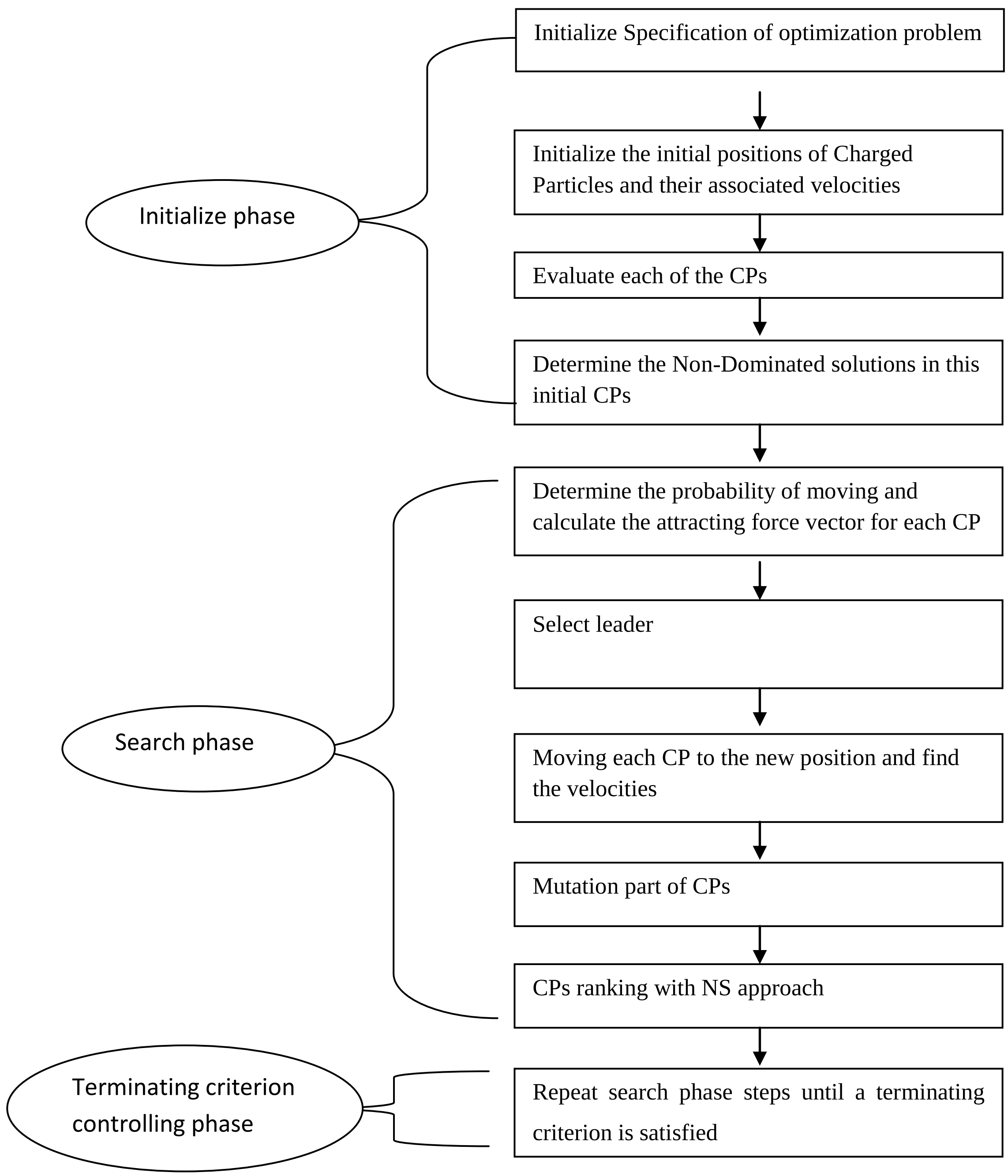

Fig. 2. MOCSS algorithm

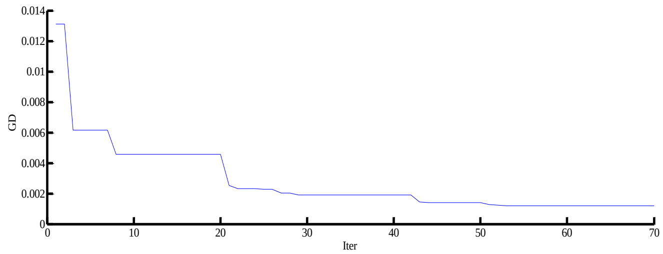

Fig. 3. Convergence of the proposed MOCSS. The least-square distance (vertical axis) from the estimated front to the true front of MOP1 for the first 70 iterations (horizontal axis)

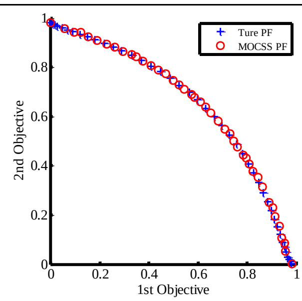

Fig. 4. Pareto front of test function MOP2

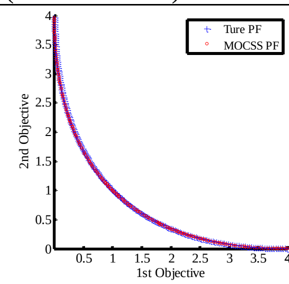

Fig. 4. Pareto front of test function MOP1

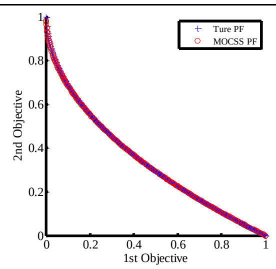

Fig. 4. Pareto front of test function ZDT1

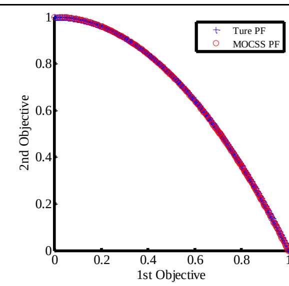

Fig. 4. Pareto front of test function ZDT2

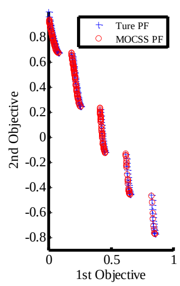

Fig. 4. Pareto front of test function ZDT3

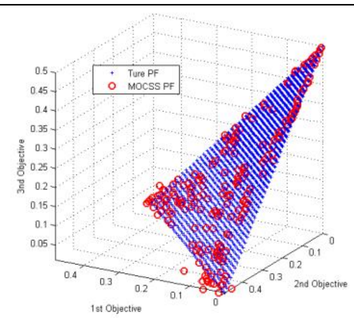

Fig. 4. Pareto front of test function DTLZ1

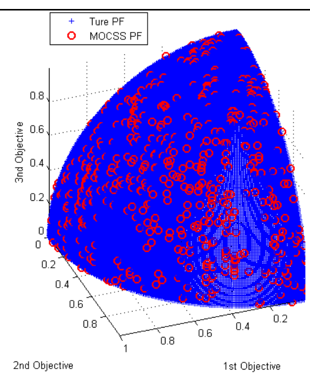

Fig. 4. Pareto front of test function DTLZ2

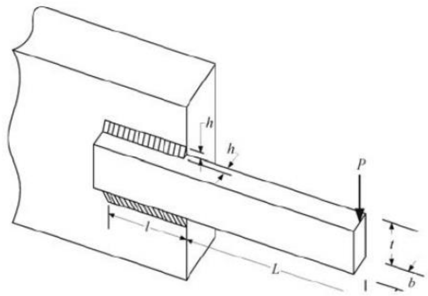

Fig. 5. The welded beam design problem

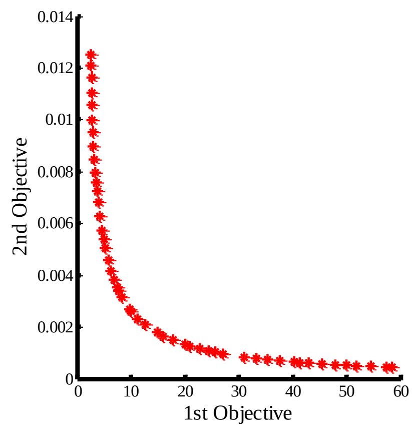

Fig. 6. Pareto front for the bi-objective beam design

where the horizontal axis corresponds to cost and the vertical axis corresponds to deflection

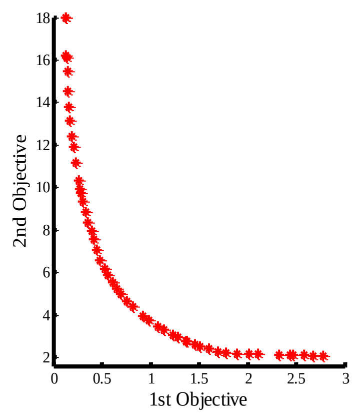

Fig. 7. Pareto front for the bi-objective disc brake design

where the horizontal axis corresponds to minimize the overall mass and the vertical axis corresponds to braking time

Table 1: benchmark for multi-objective optimization

Name of benchmark	Mathematical models	Description
MOP1 Schaffer's Min–Min (SCH)	$F=(f_1(x), f_2(x))$, $-10^3 \le x_i \le 10^3$ $f_1(x)=x^2$ $f_2(x)=(x-2)^2$	*Ptrue* connected, *PFtrue* convex benchmark function with convex Pareto front
MOP2	$F=(f_1(x), f_2(x))$,$i=1,2,3$, $-4 \le x_i \le 4$ $f_1(x)=1-\exp(-\sum_{i=1}^{n}(x_i-\frac{1}{\sqrt{n}})^2)$ $f_2(x)=1-\exp(-\sum_{i+1}^{n}(x_i+\frac{1}{\sqrt{n}})^2)$	*Ptrue* connected, *PFtrue* concave number of decision variables scalable
ZDT1	$F=(f_1(x), f_2(x))$, $0 \le x_i \le 1$, $n=30$ $f_1(x)=x_1$ $f_2(x,g)=g(x)\times(1-\sqrt{\frac{f_1}{g(x)}})$ $g(x)=1+\frac{9}{n-1}\sum_{i=2}^{n}x_i$	Has a convex Pareto-optimal front
ZDT2	$F=(f_1(x),f_2(x))$, $0 \le x_i \le 1$, $n=30$ $f_1(x)=x_1$ $f_2(x,g)=g(x)\times(1-\frac{f_1}{g(x)})^2$ $g(x)=1+\frac{9}{n-1}\sum_{i=2}^{n}x_i$	Has a non-convex Pareto-optimal front
ZDT3	$F=(f_1(x), f_2(x))$, $0 \le x_i \le 1$, $n=30$ $f_1(x)=x_1$ $f_2(x,g)=g(x)\times(1-\sqrt{\frac{f_1}{g(x)}}-\frac{f_1}{g(x)}\times\sin(10\pi f_1))$ $g(x)=1+\frac{9}{n-1}\sum_{i=2}^{n}x_i$	Has a Pareto-optimal front disconnected, consisting of several noncontiguous convex parts
DTLZ1	$F=(f_1(x),f_2(x),f_3(x))$, $0 \le x_i \le 1$, $n=10$ $f_1(x)=x_1x_2(1+g(x))$ $f_2(x)=x_1(1-x_2)(1+g(x))$ $f_3(x)=(1-x_1)(1+g(x))$ where $\mathrm{g(x)}=100(\mathrm{n}-2)+100\sum_{i=3}^{n}\{(x_i-0.5)^2-\cos[20\pi(x_i-0.5]\}$ $\sum_{i=1}^{3}f_i=1\ with\ f_i>0$	Its PF is linear Pareto-optimal front, separable, multimodal.

DTLZ2	$F=(f_1(x),f_2(x),f_3(x)),\ 0\le x_i\le 1$ $f_1(x)=\cos(\frac{x_1\pi}{2})\cos(\frac{x_2\pi}{2})(1+g(x))$ $f_2(x)=\cos(\frac{x_1\pi}{2})\sin(\frac{x_2\pi}{2})(1+g(x))$ $f_3(x)=\sin(\frac{x_1\pi}{2})(1+g(x))$ where $\mathrm{g(x)}=\sum_{i=3}^{n}(x_i-0.5)^2$ $\sum_{i=1}^{3}f_i^2=1\ with\ f_i>0$	Its PF is non-convex.
DTLZ3	$F=(f_1(x),f_2(x),f_3(x)),\ 0\le x_i\le 1$ $f_1(x)=\cos(\frac{x_1\pi}{2})\cos(\frac{x_2\pi}{2})(1+g(x))$ $f_2(x)=\cos(\frac{x_1\pi}{2})\sin(\frac{x_2\pi}{2})(1+g(x))$ $f_3(x)=\sin(\frac{x_1\pi}{2})(1+g(x))$ where $\mathrm{g(x)}=100(\mathrm{n}-2)+100\sum_{i=3}^{n}\{(x_i-0.5)^2-\cos[20\pi(x_i-0.5]\}$ $and\ \ x=(x_1,x_2,...,x_n)^T\in[0\ 1]^T$	*PFtrue* concave, scalable, multimodal
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 $\mathrm{g(x)}=\sum_{i=3}^{n}(x_i-0.5)^2,\alpha=100$ $and\ \ x=(x_1,x_2,...,x_n)^T\in[0\ 1]^T$	*PFtrue* 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 $\mathrm{g(x)}=\sum_{i=3}^{n}(x_i-0.5)^2$ $\theta_1=x_1,\theta_2=\frac{(1+2x_2g(x))}{2(1+g(x))}\ \ and\ \ x=(x_1,x_2,...,x_n)^T\in[0\ 1]^T$	*PFtrue* unimodal The function value of a Pareto optimal solution satisfies $\sum_{i=1}^{3}f_i^2=1$

Table 2: 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
Std. Dev.	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: 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
Std. Dev.	0.0109	1	0.0032	0.0783
Best	1	1	1	1
Worst	0.9596	1	0.9859	0.7924

Table 2: 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
Std. Dev.	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

Table 3: Comparison of GD for n=50 (charges particle) and iteration=500.

(Yang, 2013 and Madavan, 2002 and Chen et al. 2014))

(All results have been averaged over 30 independent runs. A result with Boldface indicates better value obtained)

Test function / Methods	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. (-1→worse, 0→equal and +1→better)

Test function / Methods	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

Table 5: Average Coverage of two Set 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

Table 6: Average and Std. Dev. relative Hypervolume between MOCSS, NSGA-II and SPEA2 (independent runs=30) (Wagner et al., 2007)

	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)

Table 7: Comparison between MOCSS and other algorithms on Average GD metrics and the basis of Wilcoxon's Ranksum test. (-1→worse, 0→equal and +1→better) (Chen et al. 2014) (independent runs=20 and pop.=100)

	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

Table 8: Average and Std. Dev. 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

	Siamak Talatahari is an assistant 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.
	Farshad Hakimpour is an assistant professor of Geomatics Dept in Tehran University. Dr. Hakimpour is the author of more than 10 papers published in international journals and other papers presented at international conferences. Dr 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 School of Surveying and Geospatial Engineering. Dr. Hakimpour’s research interests and published papers are in the domains of spatial Web, semantic Web, spatial databases, modelling movement and spatial optimization.
	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.