

## A Hybrid Simplex Non-dominated Sorting Genetic Algorithm for Multi-Objective Optimization

Seid H Pourtakdoust\* and Seid M Zandavi

Department of Aerospace Engineering, Sharif University of Technology, Tehran, Iran

### Abstract

This paper introduces a hybrid scheme for multi-objective optimization problems via utility of two established heuristic algorithms. The proposed hybrid scheme consists of two parts that include the Nelder-Mead simplex algorithm (SA) as well as the Non-dominated Sorting Genetic Algorithm II (NSGA II). In this respect, subsequent to NSGA II sorting for the optimum points, the SA searches the optimum set to find the local optimal points and thus localize a promising area that is likely to contain the global minimum. This is especially helpful since SA is an efficient algorithm that can accurately and quickly exploit the promising area for the optimum point. The proposed hybrid scheme is applied to multi-objective optimization of some benchmark functions and its performance is compared against those of the classical NSGA II as well as the Multi-Objective Particle Swarm Optimization (MOPSO). The numerical results show that the proposed hybrid scheme provides competitive results that outperform those of the existing algorithms.

**Keywords:** Heuristic hybrid algorithm; Multi-objective optimization problem; Simplex algorithm; Genetic algorithm

### Introduction

Often, in many engineering applications it is required to find the best approximate solution of multi-objective optimization problems quick and with good accuracy. Multi-objective optimization problems are very common and important in real world applications, and as such many researchers are still working to devise various novel heuristic and mathematical approaches for their solution. Mathematical techniques were more common in the earlier times for optimal solution, but as many of the conventional and especially deterministic pure mathematical optimization schemes suffer from stagnation in local optima, there is a demand for new innovations for heuristic methods [1]. The history of heuristic algorithms as a revolutionary idea for stochastic optimization techniques goes back to 1984 [1,2]. Further, it is notable that stochastic optimization techniques are readily applicable to the real problem because of gradient free mechanism and local optima trap avoidance. Stochastic and multi-objective optimization is in use in different fields of study such as mechanical engineering [3], civil engineering [4], chemistry [5,6] and other [7]. There are also some heuristic algorithms suggested to solve challenging problems in power systems such as cuckoo search algorithm [8], Pseudo-Gradient based Particle Swarm Optimization with Constriction Factor (PG-PSOCF) [9] and Hybrid Particle Swarm Optimization (HPSO) [10].

Within the past decade, a number of multi-objective heuristic algorithms has been suggested [11–14] that represent various trade-offs between the objectives including optimal solutions of a multi-objective problem [15]. In contrary to classical optimization methods converting the multi-objective optimization problem to a single-objective optimization problem by emphasizing one particular Pareto optimal solutions at a time, multi-objective heuristic algorithms give a set of optimal solutions called Pareto optimal solutions (PAS). A key point in this regard is to find accurate approximation of true Pareto optimal solutions with the highest diversity for multi-objective optimization algorithms [16]. PAS help decision makers to select a certain optimal points in a diverse range of design options based on their experience and design requirements. Among the most well-known stochastic

optimization techniques are Non-dominated Sorting Genetic Algorithm (NSGA) [17], Strength-Pareto Evolutionary Algorithm (SPEA) [18,19], Pareto Archived Evolution Strategy (PAES) [20], Pareto frontier Differential Evolution (PDE) [21], Efficient Ant Colony Optimization [22], Hybridization of Fruit Fly Optimization Algorithm and Firefly Algorithm (FOA-FA) [23], Non-dominated Sorting Genetic Algorithm II (NSGA II) [24] and Multi-Objective Particle Swarm Optimization (MOPSO) [25]. MOPSO is one of the most popular multi-objective meta-heuristic algorithms that are conceptually similar to PSO. In more recent MOPSO studies [26–28] extra conditions such as diversity estimation are utilized to reach better exploration characteristics. In this sense, the concept of Pareto dominance for selecting leaders from a non-dominated external archive has been utilized by MOPSO where the leaders of swarms that guide the particles to the Pareto Frontier are selected from the top portion of the archive at each iteration. A dynamic fitness inheritance technique [29] is also applied to reduce the computation cost of the optimization process. Another well-known multi-objective meta-heuristics is NSGA II [24] that is a multi-objective version of the well-regarded Genetic Algorithm (GA) [30,31]. NSGA II was proposed to remove several deficiencies of its first version [17] that included high computational cost of non-dominated sorting, lack of elitism and lack of sharing parameters. In the second version, the first population is generated randomly and each individual is grouped based on the non-dominated sorting method and new population is generated by genetic operators such as selection, crossover and mutation operators. The new population

\*Corresponding author: Seid H Pourtakdoust, Department of Aerospace Engineering, Sharif University of Technology, Tehran, Iran, Tel: 982166164610; E-mail: [pourtak@sharif.edu](mailto:pourtak@sharif.edu)

Received August 26, 2016; Accepted August 29, 2016; Published August 31, 2016

Citation: Pourtakdoust SH, Zandavi SM (2016) A Hybrid Simplex Non-dominated Sorting Genetic Algorithm for Multi-Objective Optimization. Int J Swarm Intel Evol Comput 5: 138. doi: [10.4172/2090-4908.1000138](https://doi.org/10.4172/2090-4908.1000138)

Copyright: © 2016 Pourtakdoust SH, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

is next sorted again and the new elements of the final population are selected via higher non-dominated level. The process of selecting the non-dominated individuals is repeated till reaching a population of the same size as the initial population.

More recently over the past three years a number of multi-objective optimization algorithms have also been proposed. These new studied include Multi-objective Grey Wolf Optimizer [1], Multi-objective Cat Swarm Optimization [32], Multi-objective Differential Evolution [33], Multi-objective Gravitational Search Algorithm [34], Multi-objective Ant Colony Optimization [35], Multi-objective Artificial Bee Colony algorithm [36] and the Multi-objective Teaching-Learning-Based Optimization algorithm [37]. These studies show the ability of meta-heuristics in handling multiple objectives. Although all algorithms are able to approximate the true Pareto optimal front of a given problem, they are not able to solve all optimization problems accurately. Therefore, it is still very likely that new hybrid algorithms to be proposed that can solve new problems that not solved before and/or with a better accuracy as compared with the existing techniques.

In this respect a new hybrid heuristic optimization algorithm is proposed in the current work for multi-objective problems. The proposed hybrid algorithm hybridizes the Nelder-Mead simplex algorithm with the Non-dominated Sorting Genetic Algorithm II (NSGA II) to find the best global point. The performance of the new proposed algorithm is demonstrated via some complex benchmark functions.

The organization of this paper is as follows: section 2 deals with the mathematical representation of multi-objective optimization. Section 3 describes the simplex method. Section 4 is devoted to non-dominated sorting genetic algorithm II. The proposed hybrid algorithm is introduced in section 5. Section 6 provides the numerical simulation results. Finally, a conclusion is drawn in section 7.

## Multi-Objective Optimization

The problem is to minimize the fitness function of a multi-objective problem. The model is described as follows:

$$\begin{aligned}
 F(x) &= \{ f_m(x) \in \mathbb{R}^M \quad m=1, 2, \dots, M \} \\
 \text{subject to} \quad & g_j(x) \leq 0 \quad j=1, 2, \dots, J \\
 & h_k(x) = 0 \quad k=1, 2, \dots, K
 \end{aligned} \tag{1}$$

where  $x = \{x_n \in \mathbb{R}^N \quad n=1, 2, \dots, N\}$  is the N-dimensional vector of decision variables that represents the possible solutions, M is the number of objectives,  $f_m$  is the objective function, and  $(g_j, h_k)$  are the inequality and equality constraint functions, respectively.

To clarify the concept of Pareto optimality, consider two vectors  $\{F^a, F^b\}^M$ .  $F^a$  dominates  $F^b$  (denoted by  $F^a \prec F^b$ ) if and only if  $(f_m^a \leq f_m^b)$ . The dominance relations  $F^a \prec F^c$  and  $F^a \prec F^c$  for a two-objective problem are indicated by labelled circles in Figure 1. Hence, a vector of decision variables  $X^a$  is a non-dominated solution if and only if there is no other solution  $X^b$  such that  $F(x^a) \prec F(x^b)$ . The non-dominated solution is also known as the Pareto optimal solution. The set of non-dominated solution of a multi-objective optimization problem is known as the Pareto Optimal set (P). The set of objective vectors with respect to (P) is known as the Pareto Front ( $PF = \{F(x) \in \mathbb{R}^M \mid x \in P\}$ ). The

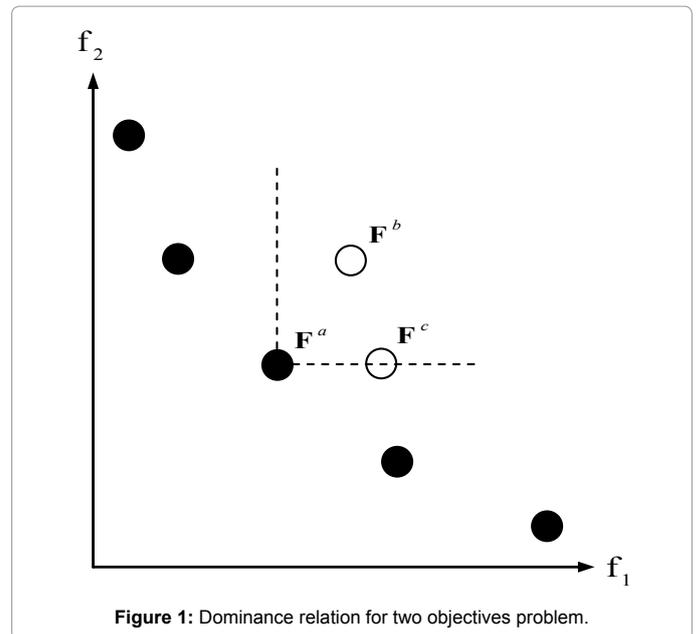


Figure 1: Dominance relation for two objectives problem. PF for a two-objective problem is illustrated by the black circles in Figure 1.

It is desired to find as many non-dominated solutions as possible according to the conflicting and different objective functions and constraints in the multi-objective optimization algorithm. The Pareto front corresponding to the non-dominated set should be as close to and well distributed over the true Pareto front as possible. However, it is possible to have different solutions that map to the same fitness value in objective space.

## Simplex Method

Simplex is a heuristic optimization algorithm. It is a direct search method that does not use the derivatives of the objective function. Simplex is a geometrical object produced by n+1 points ( $X_0, \dots, X_n$ ) in an n-dimensional space [38,39]. Thus for example, the simplex in two-dimension space is a triangle. The basic idea of the simplex algorithm is to compare the value of the objective function at n+1 vertices of a simplex object and move it gradually toward the optimum point via an iterative process. The following equations can be used to generate the vertices of a regular simplex of size  $a$  (equilateral triangle in two-dimensional space), within the n-dimensional space [39]:

$$x_j = x_0 + p x_j + \sum_{\substack{s=1 \\ s \neq j}}^n q u_s \tag{2}$$

where  $X_0$  is the initial base point,  $u_s$  is the unit vector along the coordinate axis  $s$  and  $p, q$  are defined as follows:

$$p = \frac{a}{n\sqrt{2}} (\sqrt{n+1} + n - 1) \tag{3}$$

$$q = \frac{a}{n\sqrt{2}} (\sqrt{n+1} - 1) \tag{4}$$

Through a sequence of elementary geometric transformations (reflection, contraction and expansion), the simplex, shown in Figure 2, moves toward the optimum point where after each transformation, the current worst vertex is replaced by a better one.

The reflected, expanded and contracted points are given by:  $X_r, X_e$  and  $X_c$ , respectively.

$$x_r = (1 + \alpha) \bar{x}_0 - \alpha x_h \quad \alpha > 0 \quad (5)$$

$$x_e = \gamma x_r + (1 - \gamma) \bar{x}_0 \quad \gamma > 1 \quad (6)$$

$$x_c = \beta x_h + (1 - \beta) \bar{x}_0 \quad 0 \leq \beta \leq 1 \quad (7)$$

Here,  $\bar{x}_0$  is the centroid of all vertices  $x_j$  except  $j = h$ ; where  $h$  is the index of the worst point. The parameters  $\alpha$ ,  $\beta$  and  $\gamma$  are called reflection, expansion and contraction coefficients, respectively.

Reflection is the operation by which the worst vertex, called high, is reflected with respect to the centroid  $\bar{x}_0$ . If the reflected point is better than all other points, the method expands the simplex in the reflection direction; otherwise, if it is at least better than the worst, the algorithm performs again the reflection with the new worst point [39]. The contraction is performed when the worst point is at least as good as the reflected point.

### Non-dominated Sorting Genetic Algorithm II (NSGA II)

The NSGA II [24] algorithm is one of the most efficient and famous algorithms for multi-objective optimization. It uses fast non-dominated sorting to rank the population fronts and a parameter called crowding distance is calculated in the same front. Subsequently, the tournament selection is made between two individuals that are randomly selected from the parent population. The individual with lower front number is selected if the two individuals come from different fronts. The individual with higher crowding distance is selected if the two individuals are from the same front. Next the crossover and mutation operators are used to generate a new offspring population. And finally, the parent and offspring population are combined together where a fast non-dominated sorting and crowding distance assignment procedure is used to rank the combined population and only the best  $N$  individuals are selected as the new parent population [40].

NSGA II is proposed on basis of non-dominated sorting genetic algorithm (NSGA) [17] and the main advantages of NSGA II compared with NSGA are as follows: (1) it uses a fast non-dominated sorting approach, (2) it has no sharing parameter and (3) it uses an elitist strategy [40]. NSGA II has good global search ability and well distributed non-dominated solution in the Pareto front. Nevertheless, the convergence capability of NSGA II is limited and its elite strategy may result in the loss of the non-dominated solutions in the Pareto front [40].

### Simplex Non-Dominated Sorting Genetic Algorithm II

In This section, the proposed hybrid algorithm is introduced as a tool for multi-objective optimization problems. First the basic structure of the proposed algorithm is presented followed by detail description of the constructive modules.

#### General setting out of the algorithm

The block diagram of Simplex-NSGA II is shown in Figure 3. The parameters of Simplex-NSGA II are set during the initialization. Simplex-NSGA II utilizes two heuristic algorithms, simplex optimization method and NSGA II to find the optimal solution. In this approach, simplex optimization is utilized as exploration to make better diversity within the search space. In addition NSGA II is used as

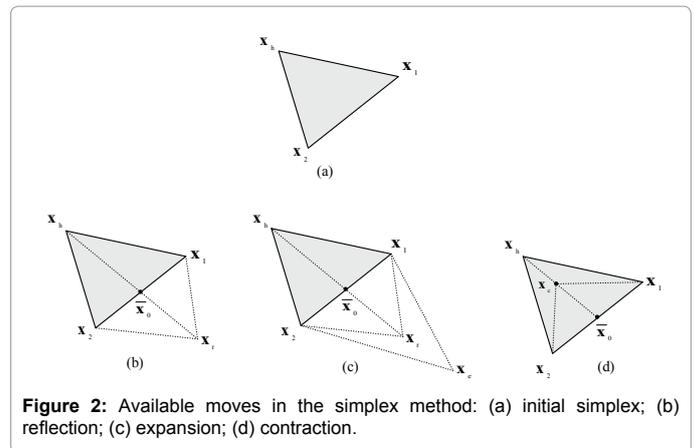


Figure 2: Available moves in the simplex method: (a) initial simplex; (b) reflection; (c) expansion; (d) contraction.

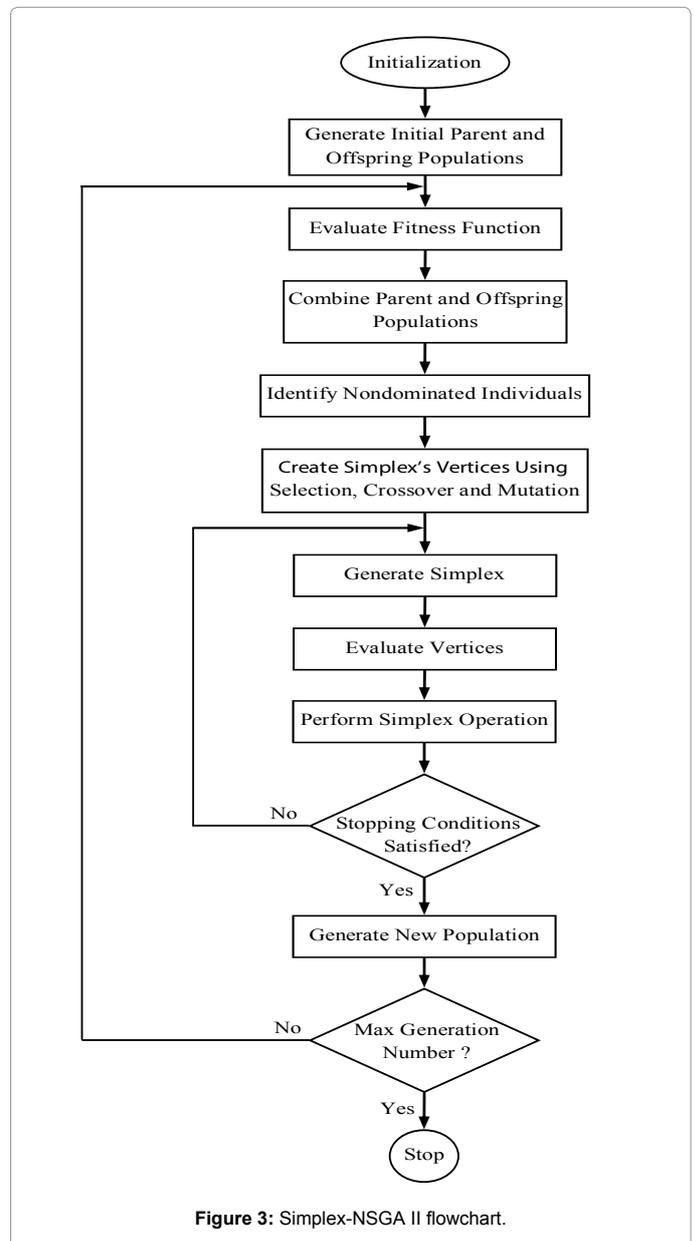


Figure 3: Simplex-NSGA II flowchart.

exploitation to arrange the member of Pareto optimal solutions is the main form of a multi-objective optimization algorithm. The proposed Simplex-NSGA II consists of two loops. The outer loop generates new population using the simplex method at each generation. The inner loop initially propagates the vertices of simplex using reflection, expansion and contraction operations to reshape and move the simplex toward likelihood regions of the search space and is terminated when the maximum number of iterations ( $i_{max}$ ) is reached. Table 1 represents the pseudo-code of the Simplex-NSGA II.

### Initialization

Simplex-NSGA II has nine parameters that must be set before the execution of the algorithm. The parameters are listed in Table 2 and are introduced in the following subsections.

### Generation initial population

In the proposed method, the initial parent population ( $P_0$  of size  $N_{pop}$ ) is generated randomly with uniform distribution. The population is sorted based on non-domination and each solution is assigned a fitness value equal to its non-domination level. An initial offspring population ( $Q_0$ ) with the same size as the parent population is created through recombination based on binary tournament selection and inducing variations via mutation operators.

### Population update

From the first generation onward, a combined population ( $R_{gen} = P_{gen} \cup Q_{gen}$ ) is formed that is sorted according to fast non-domination procedure. The new parent population is formed by the inner loop. Therefore, during any iteration of the inner loop, the

population will be updated using reflection, expansion and contraction operators for the next generation. In this respect, any individual that is generated by genetic operators such as selection, crossover and mutation is utilized as vertices of the simplex to generate the initial simplex in order to create the new population.

### Stop condition

The inner loop stops when the maximum number of iterations ( $i_{max}$ ) is reached. The outer loop is stopped when the maximum number of generation is reached.

### Numerical Results

In this section, the performance of Simplex-NSGA II is evaluated using ten benchmarks that are proposed in CEC 2009 [41], listed in Table 3 and 4. The results are compared to well-known algorithms of Non-dominated Sorting Genetic Algorithm II (NSGA II) [24] and Multi-Objective Particle Swarm Optimization (MOPSO) [28]. For the performance metric [32], Inverted Generational Distance (IGD) [42], Spacing (SP) [43] and Maximum Spread (MS) [18] criteria are employed to measure convergence, quantity and coverage respectively. The mathematical formulation of IGD is as follows:

$$IGD = \frac{\sqrt{\sum_{i=1}^N d_i^2}}{N} \quad (8)$$

Where  $N$  is the number of true Pareto optimal solutions and  $d_i$  indicates the Euclidean distance between the  $i$ -th true Pareto optimal solution and the closet obtained Pareto optimal solutions in the reference set. It should be noted that  $IGD=0$  indicates that all members of the non-dominated solutions are in the true Pareto Front.

The mathematical formulation of the SP and MS are as follows:

$$SP = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N-1} (\bar{d} - d_i)^2} \quad (9)$$

where  $\bar{d}$  is the average of all  $d_i$ ,  $N$  is the number of Pareto optimal solutions obtained, and  $d_i = \min_j (|f_1^i(x) - f_1^j(x)| + |f_2^i(x) - f_2^j(x)|)$  for all  $i, j = 1, 2, 3, \dots, N$ .

Initialization	
Set: Genetic Parameters ( $N_{gen}, N_{pop}, P_{cr}, P_{mut}$ ),	
Simplex Parameters ( $a, \alpha, \gamma, \beta, i_{max}$ )	
Generate initial population ( $P_0$ and $Q_0$ )	
gen=1	
Repeat	
Evaluate fitness function	
Combine $Q_{gen}$ and $Q_{gen}$	
Classify the population ( $R_{gen} = P_{gen} \cup Q_{gen}$ )	
Create simplex's vertices	
Selection	
Crossover	
Mutation	
$i=1$	
Repeat	
Generate simplex	
Compute cost function for each vertex	
Update simplex	
Reflection	
Contraction	
Expansion	
$i=i+1$	
<b>Until</b> $i \geq i_{max}$	
Create new population	
gen=gen+1	
<b>Until</b> gen $\geq N_{gen}$	

Table 1: Simplex-NSGA II pseudo-code.

Parameter	Description	Value
$N_{pop}$	Population size	50
$N_{gen}$	Maximum generation	100
$P_{crs}$	Probability of crossover	0.80
$P_{mut}$	Probability of mutation	0.01
$a$	Side of simplex	22
$\alpha$	Reflection coefficient	0.9
$\gamma$	Expansion coefficient	2
$\beta$	Contraction coefficient	0.005
$i_{max}$	Maximum iteration of simplex method	60

Table 2: Parameters of simplex-NSGA II.

Name	Mathematic Formulation
UF1	$f_1 = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} \left[ x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right) \right]^2, f_2 = 1 - \sqrt{x_1} + \frac{2}{ J_2 } \sum_{j \in J_2} \left[ x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right) \right]^2$ $J_1 = \{j   j \text{ is odd and } 2 \leq j \leq n\}, J_2 = \{j   j \text{ is even and } 2 \leq j \leq n\}$
UF2	$f_1 = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} y_j^2, f_2 = 1 - \sqrt{x_1} + \frac{2}{ J_2 } \sum_{j \in J_2} y_j^2$ $J_1 = \{j   j \text{ is odd and } 2 \leq j \leq n\}, J_2 = \{j   j \text{ is even and } 2 \leq j \leq n\}$ $y_j = \begin{cases} x_j - [0.3x_1^2 \cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1] \cos(6\pi x_1 + \frac{j\pi}{n}) & \text{if } j \in J_1 \\ x_j - [0.3x_1^2 \cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1] \sin(6\pi x_1 + \frac{j\pi}{n}) & \text{if } j \in J_2 \end{cases}$
UF3	$f_1 = x_1 + \frac{2}{ J_1 } \left( 4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} \cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2 \right), f_2 = \sqrt{x_1} + \frac{2}{ J_2 } \left( 4 \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} \cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2 \right)$ $J_1 \text{ and } J_2 \text{ are the same as those of UF1}$ $y_j = x_j - x_1^{0.5(1+\frac{3(j-2)}{n-2})}, j = 2, 3, \dots, n$
UF4	$f_1 = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} h(y_j), f_2 = 1 - x_2 + \frac{2}{ J_2 } \sum_{j \in J_2} h(y_j)$ $J_1 \text{ and } J_2 \text{ are the same as those of UF1}$ $y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, 3, \dots, n, h(t) = \frac{ t }{1 + e^{2 t }}$
UF5	$f_1 = x_1 + \left(\frac{1}{2N} + \varepsilon\right)  \sin(2N\pi x_1)  + \frac{2}{ J_1 } \sum_{j \in J_1} h(y_j), f_2 = 1 - x_1 + \left(\frac{1}{2N} + \varepsilon\right)  \sin(2N\pi x_1)  + \frac{2}{ J_2 } \sum_{j \in J_2} h(y_j)$ $J_1 \text{ and } J_2 \text{ are the same as those of UF1}, \varepsilon > 0$ $y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, 3, \dots, n, h(t) = 2t^2 - \cos(4\pi t) + 1$
UF6	$f_1 = x_1 + \max \left\{ 0, 2 \left( \frac{1}{2N} + \varepsilon \right)  \sin(2N\pi x_1)  \right\} + \frac{2}{ J_1 } \left( 4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} \cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 1 \right)$ $f_2 = 1 - x_1 + \max \left\{ 0, 2 \left( \frac{1}{2N} + \varepsilon \right)  \sin(2N\pi x_1)  \right\} + \frac{2}{ J_2 } \left( 4 \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} \cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 1 \right)$ $J_1 \text{ and } J_2 \text{ are the same as those of UF1}, \varepsilon > 0$ $y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, 3, \dots, n$
UF7	$f_1 = \sqrt[5]{x_1} + \frac{2}{ J_1 } \sum_{j \in J_1} y_j^2, f_2 = 1 - \sqrt[5]{x_1} + \frac{2}{ J_2 } \sum_{j \in J_2} y_j^2$ $J_1 \text{ and } J_2 \text{ are the same as those of UF1}$ $y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, 3, \dots, n$

Table 3: Bi-objective benchmark functions.

$$MS = \sqrt{\sum_{i=1}^M \max[d(a_i, b_i)]} \quad (10)$$

where  $d$  is a function to calculate the Euclidean distance,  $b_i$  is the maximum value in the  $i$ th objective,  $b_i$  is the minimum in the  $i$ th objective, and  $M$  is the number of objectives.

In addition to utilizing the performance metrics, the best set of Pareto optimal solution obtained by Simplex-NSGA II on both the parameter space as well as the search space is shown in Figures 4-6. These figures show the performance of Simplex-NSGA II in comparison with true Pareto front. For a comparative evaluation, all the algorithms are run 30 times on the test problems and the statistics results of the 30

runs are provided in Tables 5-7. Note that 900,000 function evaluations are used for each algorithm and the number of parameters for each benchmark is 30.

Statistical results of the algorithm for IGD, SP and MS are provided in Tables 5-7, respectively. The IGD shows that the proposed hybrid algorithm (Simplex-NSGA II) is able to provide the best results on all statistical metrics for bi-objective test problems. IGD is the performance metric that shows the accuracy and convergence of an algorithm. Thus it can be stated that the proposed Simplex-NSGA II algorithm is able to provide superior convergence on the bi-objective benchmarks.

The resulting Pareto optimal solution of Simplex-NSGA II on each benchmark is also depicted in Figures 4 and 5. These figures demonstrate

Name	Mathematic Formulation
UF8	$f_1 = \cos(0.5x_1\pi)\cos(0.5x_2\pi) + \frac{2}{ J_1 } \sum_{j \in J_1} (x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $f_2 = \cos(0.5x_1\pi)\sin(0.5x_2\pi) + \frac{2}{ J_2 } \sum_{j \in J_2} (x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $f_3 = \sin(0.5x_1\pi) + \frac{2}{ J_3 } \sum_{j \in J_3} (x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $J_1 = \{j   3 \leq j \leq n, \text{ and } j-1 \text{ is a multiplication of } 3\}$ $J_2 = \{j   3 \leq j \leq n, \text{ and } j-2 \text{ is a multiplication of } 3\}$ $J_3 = \{j   3 \leq j \leq n, \text{ and } j \text{ is a multiplication of } 3\}$
UF9	$f_1 = 0.5[\max\{0, (1 + \varepsilon)(1 - 4(2x_1 - 1)^2)\} + 2x_2]x_2 + \frac{2}{ J_1 } \sum_{j \in J_1} (x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $f_2 = 0.5[\max\{0, (1 + \varepsilon)(1 - 4(2x_1 - 1)^2)\} + 2x_2]x_2 + \frac{2}{ J_2 } \sum_{j \in J_2} (x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $f_3 = 1 - x_2 + \frac{2}{ J_3 } \sum_{j \in J_3} (x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $J_1 = \{j   3 \leq j \leq n, \text{ and } j-1 \text{ is a multiplication of } 3\}$ $J_2 = \{j   3 \leq j \leq n, \text{ and } j-2 \text{ is a multiplication of } 3\}$ $J_3 = \{j   3 \leq j \leq n, \text{ and } j \text{ is a multiplication of } 3\}, \varepsilon > 0.1$
UF10	$f_1 = \cos(0.5x_1\pi)\cos(0.5x_2\pi) + \frac{2}{ J_1 } \sum_{j \in J_1} (4y_j^2 - \cos(8\pi y_j) + 1)$ $f_2 = \cos(0.5x_1\pi)\sin(0.5x_2\pi) + \frac{2}{ J_2 } \sum_{j \in J_2} (4y_j^2 - \cos(8\pi y_j) + 1)$ $f_3 = \sin(0.5x_1\pi) + \frac{2}{ J_3 } \sum_{j \in J_3} (4y_j^2 - \cos(8\pi y_j) + 1)$ $J_1 = \{j   3 \leq j \leq n, \text{ and } j-1 \text{ is a multiplication of } 3\}$ $J_2 = \{j   3 \leq j \leq n, \text{ and } j-2 \text{ is a multiplication of } 3\}$ $J_3 = \{j   3 \leq j \leq n, \text{ and } j \text{ is a multiplication of } 3\}$

Table 4: Tri-objective benchmark functions.

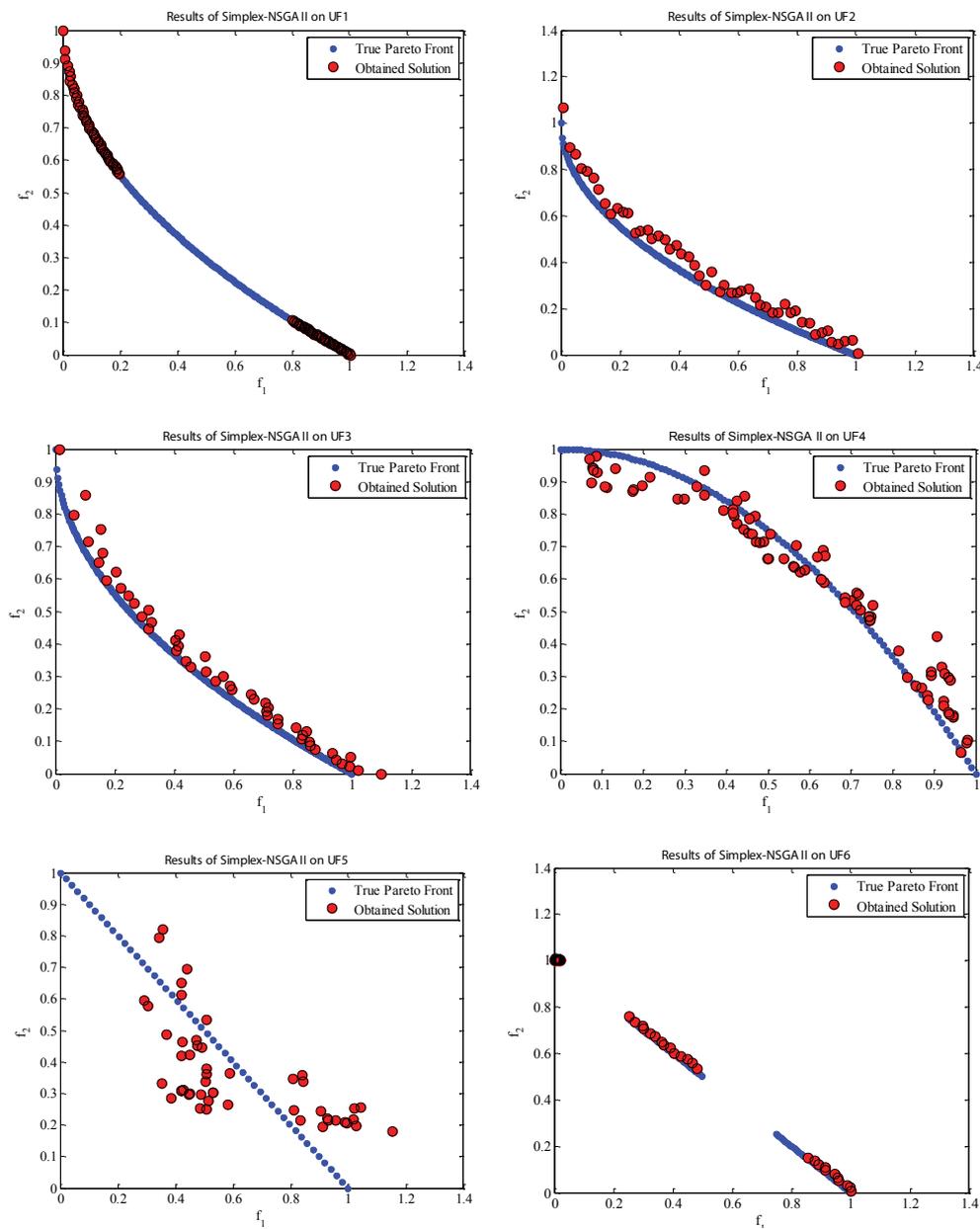


Figure 4: Pareto optimal solutions for UF1 to UF6.

the convergence and coverage characteristics (Tables 6 and 7) of the Simplex-NSGA II are also better than the other algorithms. In addition, Simplex-NSGA II Pareto optimal fronts are closer to the true Pareto front and highly distributed along both objectives. The numerical results show that the performance of the Simplex-NSGA algorithm is more stable than MOPSO and NSGA II.

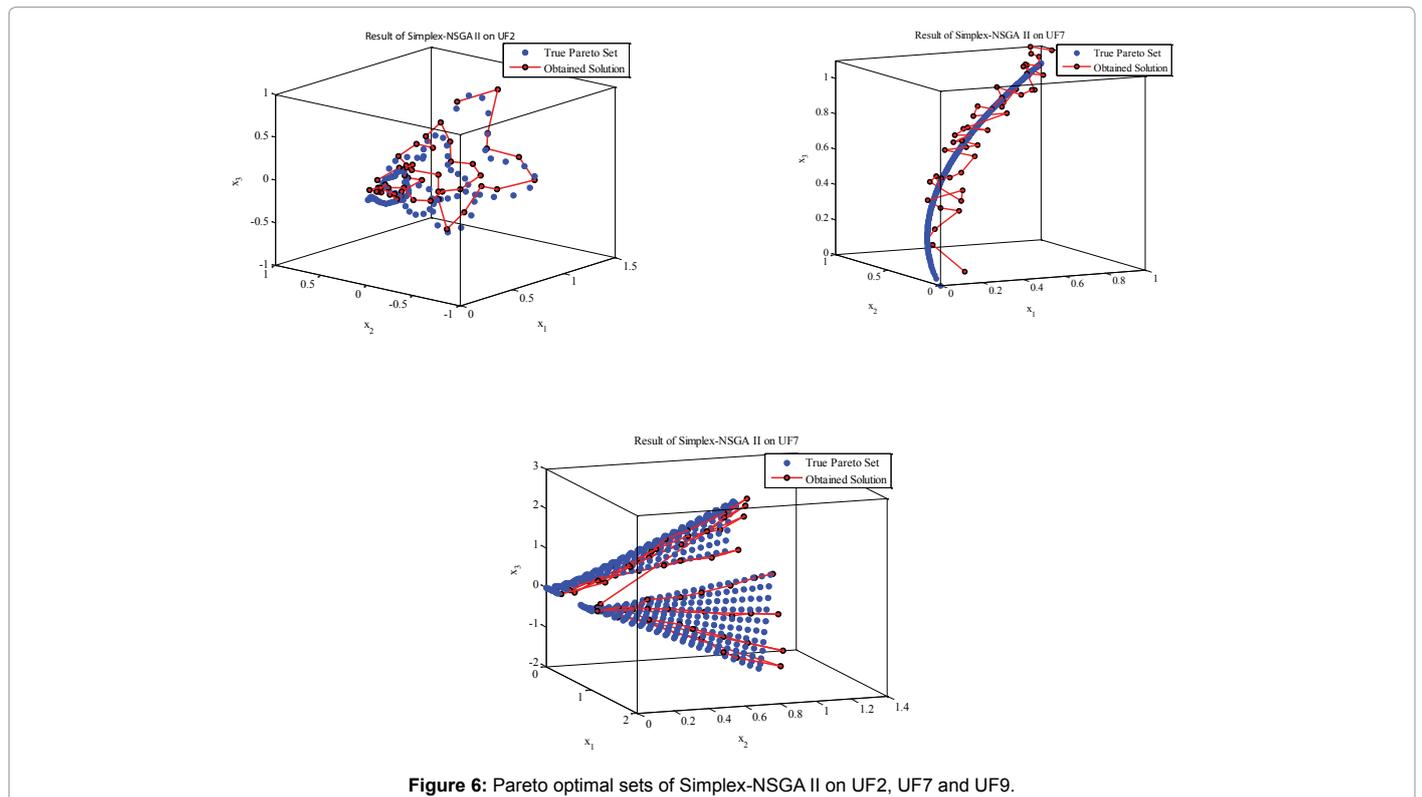
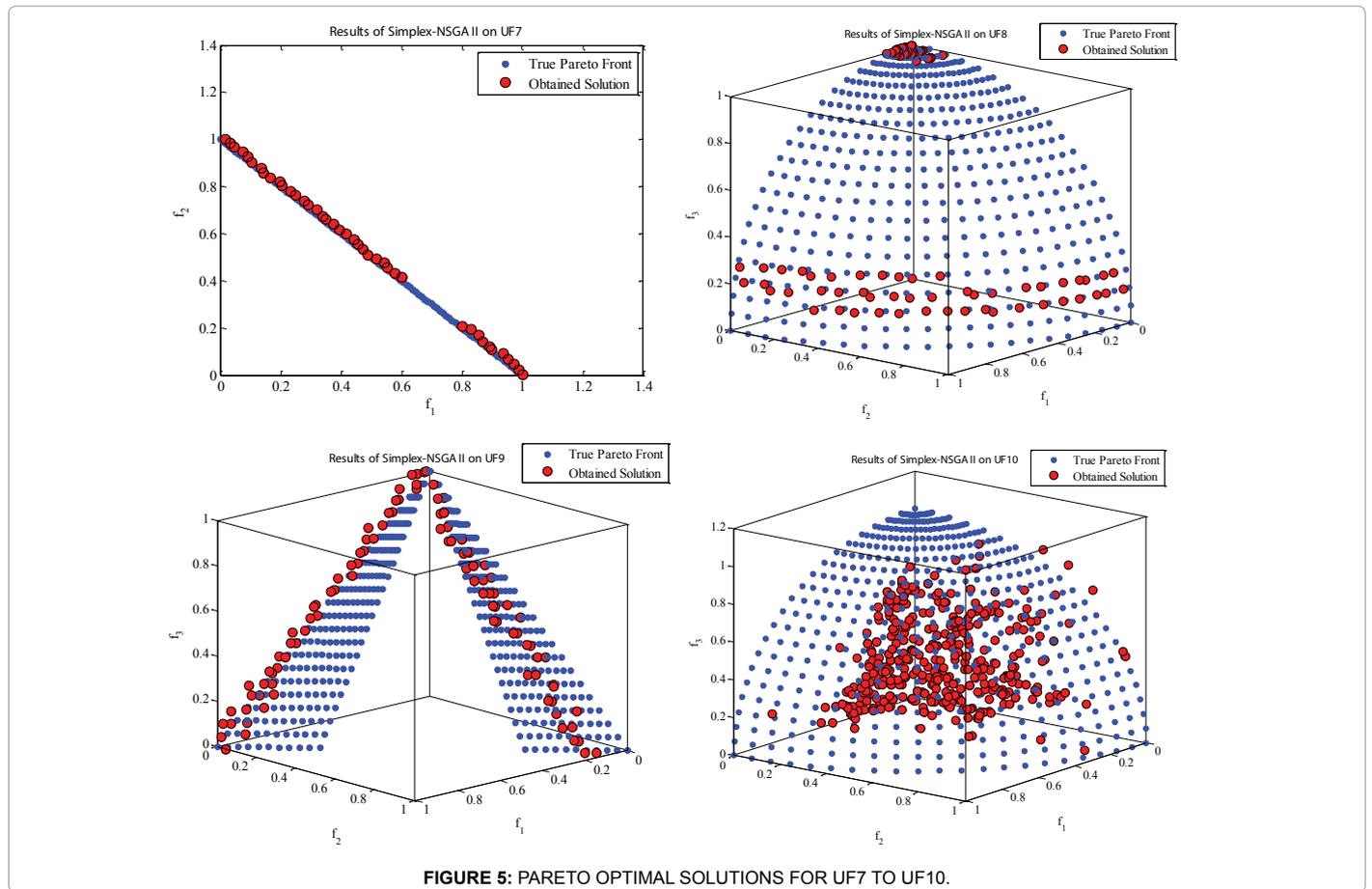
Problems UF8, UF9 and UF10 are three-objective that make them more challenging against the two-objective problems. The results of the first three-objective problem show that Simplex-NSGA II has competitive and even better performance in solving these problems as compared to NSGA II.

The Pareto optimal fronts obtained by the proposed algorithm for UF8 are shown in Figure 5. The Pareto optimal solutions have

converged towards the true Pareto optimal front despite their low diversity. Therefore, the proposed Simplex-NSGA II algorithm shows high convergence but low coverage for the UF8 in the best case.

The UF9 test problem has a separated Pareto front and its pertinent results indicate that the proposed algorithm has been effective on this challenging test function. The statistical results both quantitatively and qualitatively indicate that the coverage of the proposed Simplex-NSGA II is very good for all objectives. Finally, the results of the last UF10 test problem are similar to that of UF8, in which NSGA II provides better statistical results on the average, median and standard deviation, and the worst result is for IGD. However, the proposed Simplex-NSGA II, provides the best coverage among the 30 independent runs.

To get a better image of the Pareto optimal sets obtained via the



IGD	UF1			UF2			UF3		
	Simplex-NSGAI	MOPSO	MOEA/D	Simplex-NSGAI	MOPSO	NSGA II	Simplex-NSGAI	MOPSO	NSGA II
Average	0.04695	0.137005	0.18713	0.01359	0.06040	0.12224	0.07361	0.31399	0.28875
Median	0.03886	0.13174	0.18285	0.01225	0.04835	0.1231	0.07265	0.30802	0.28989
STD. Dev.	0.03319	0.04407	0.05073	0.00289	0.02762	0.01040	0.01108	0.04473	0.015931
Worst	0.09301	0.22786	0.24642	0.01845	0.13051	0.14389	0.09167	0.37773	0.31284
Best	0.00674	0.08990	0.12652	0.01060	0.03699	0.10456	0.05764	0.25648	0.26322
	UF4			UF5			UF6		
	Simplex-NSGAI	MOPSO	NSGA II	Simplex-NSGAI	MOPSO	NSGA II	Simplex-NSGAI	MOPSO	NSGA II
Average	0.02434	0.13604	0.06813	0.45289	2.20238	1.29155	0.15713	0.64752	0.688412
Median	0.02415	0.13432	0.06846	0.45486	2.12574	1.33731	0.13523	0.55073	0.69811
STD. Dev.	0.00138	0.00739	0.00214	0.08449	0.55304	0.13479	0.09019	0.26612	0.05543
Worst	0.02760	0.15189	0.07037	0.53701	3.03836	1.46726	0.35279	1.24281	0.74031
Best	0.02295	0.12733	0.06466	0.25741	1.46479	1.12326	0.07954	0.37933	0.55255
	UF7			UF8		UF9		UF10	
	Simplex-NSGAI	MOPSO	NSGA II	Simplex-NSGAI	NSGA II	Simplex-NSGAI	NSGA II	Simplex-NSGAI	NSGA II
Average	0.02669	0.35395	0.45534	0.19992	0.53681	0.31183	0.48840	1.70149	1.63729
Median	0.02641	0.38730	0.43776	0.20740	0.53650	0.32593	0.41441	1.54063	1.59123
STD. Dev.	0.01986	0.20442	0.18993	0.06217	0.18267	0.08044	0.14459	0.55163	0.29849
Worst	0.06600	0.61512	0.67711	0.33433	0.79647	0.48522	0.72230	3.02835	2.16250
Best	0.00558	0.05402	0.02910	0.12553	0.24530	0.22217	0.33365	1.12806	1.22028

Table 5: Statistical results for IGD on UF1 to UF10.

SP	UF1			UF2			UF3		
	Simplex-NSGAI	MOPSO	NSGA II	Simplex-NSGAI	MOPSO	NSGA II	Simplex-NSGAI	MOPSO	NSGA II
Average	0.03811	0.00898	0.00384	0.02770	0.00829	0.00876	0.01991	0.00699	0.28875
Median	0.03761	0.00855	0.00382	0.02654	0.00814	0.00859	0.02007	0.00677	0.28939
STD. Dev.	0.01516	0.00247	0.00151	0.01289	0.00168	0.00076	0.00276	0.00170	0.01582
Worst	0.06146	0.01464	0.00665	0.06199	0.01245	0.01042	0.02369	0.01007	0.31294
Best	0.01399	0.00670	0.00213	0.01640	0.00624	0.00797	0.01461	0.00476	0.26342
	UF4			UF5			UF6		
	Simplex-NSGAI	MOPSO	NSGA II	Simplex-NSGAI	MOPSO	NSGA II	Simplex-NSGAI	MOPSO	NSGA II
Average	0.02543	0.00666	0.00730	0.12049	0.00479	0.00278	0.06150	0.02084	0.00640
Median	0.02386	0.00662	0.00728	0.12758	0.00487	0.00007	0.05482	0.01235	0.00000
STD. Dev.	0.00475	0.00091	0.00059	0.02505	0.00408	0.00553	0.04126	0.03258	0.01227
Worst	0.03384	0.00809	0.00836	0.15826	0.01206	0.01615	0.13353	0.11140	0.03010
Best	0.02055	0.00546	0.00610	0.07457	0.00006	0.00000	0.01222	0.00215	0.00000
	UF7			UF8		UF9		UF10	
	Simplex-NSGAI	MOPSO	NSGA II	Simplex-NSGAI	NSGA II	Simplex-NSGAI	NSGA II	Simplex-NSGAI	NSGA II
Average	0.02541	0.00670	0.00540	0.23364	0.02682	0.25104	0.02343	1.06788	0.01994
Median	0.02379	0.00655	0.00441	0.23699	0.02639	0.24302	0.02349	0.89524	0.02066
STD. Dev.	0.01286	0.00285	0.00301	0.03859	0.00827	0.03695	0.00405	0.41661	0.00348
Worst	0.05257	0.01240	0.01168	0.27492	0.04473	0.30891	0.03087	1.81766	0.02665
Best	0.00351	0.00325	0.00084	0.16389	0.01531	0.19916	0.01716	0.67219	0.01536

Table 6: Statistical results for SP on UF1 to UF10.

proposed Simplex-NSGA II, the first three variables of the true Pareto optimal set and determined sets are illustrated in Figure 6 showing a very good convergence and coverage characteristics.

In summary, numerical results demonstrate that the proposed Simplex-NSGA II has better performance for bi-objective benchmark functions with regards to convergence and coverage. However, for some tri-objective benchmark functions, even though the proposed algorithm shows high convergence, it has a low coverage. Thus one could say the key advantages of the proposed Simplex-NSGA II algorithm compared to NSGA II and MOPSO are high convergence

and coverage characteristics. In addition, the results of the proposed Simplex-NSGA II were in almost all cases better than those of MOPSO. Thus the results show that the proposed Simplex-NSGA II outperforms MOPSO as it utilizes simplex operators for generating new population obtained from the non-dominated solution. In addition, MOPSO updates gBest in each iteration. So all particles are attracted by the same (or a similar set of) gBest(s) in each iteration [44], while the individuals of Simplex-NSGA II, are updated for each generation thus assisting search agents to explore the search space more extensively.

MS	UF1			UF2			UF3		
	Simplex-NSGAI	MOPSO	NSGA II	Simplex-NSGAI	MOPSO	NSGA II	Simplex-NSGAI	MOPSO	NSGA II
Average	0.96922	0.64538	0.51774	0.89222	0.91205	0.87201	0.99659	0.61030	0.23994
Median	0.91629	0.66322	0.59541	0.88328	0.91636	0.87437	1.00000	0.61612	0.22943
STD. Dev.	0.13919	0.19292	0.16609	0.05448	0.02560	0.00560	0.08165	0.10575	0.12129
Worst	0.79407	0.26592	0.31478	0.82385	0.86654	0.85986	0.87763	0.38172	0.08975
Best	0.99536	0.95226	0.74128	0.93883	0.95301	0.87794	1.00000	0.77145	0.47863
	UF4			UF5			UF6		
	Simplex-NSGAI	MOPSO	NSGA II	Simplex-NSGAI	MOPSO	NSGA II	Simplex-NSGAI	MOPSO	NSGA II
Average	0.95759	0.81275	0.88320	0.83746	0.27926	0.29215	0.98393	0.27435	0.09677
Median	0.96323	0.81321	0.88131	0.83556	0.28654	0.29165	1.00000	0.22917	0.00005
STD. Dev.	0.01762	0.01367	0.01812	0.06986	0.09575	0.03470	0.21202	0.11285	0.20715
Worst	0.92252	0.79441	0.85324	0.69914	0.15574	0.23834	0.79702	0.15436	0.00000
Best	0.97665	0.83449	0.91394	0.95996	0.43827	0.34380	1.00000	0.52516	0.59484
	UF7			UF8		UF9		UF10	
	Simplex-NSGAI	MOPSO	NSGA II	Simplex-NSGAI	NSGAI	Simplex-NSGAI	NSGA II	Simplex-NSGAI	NSGA II
Average	0.98737	0.42928	0.56317	0.35198	0.50810	0.87641	0.19160	0.27643	0.13025
Median	1.00000	0.29520	0.63267	0.34883	0.50601	0.94795	0.16566	0.14774	0.10933
STD. Dev.	0.09838	0.27553	0.24209	0.44928	0.16136	0.39021	0.16351	0.38569	0.06243
Worst	0.80289	0.14458	0.14963	0.27879	0.22723	0.30638	0.06771	0.03183	0.06479
Best	1.00000	0.87714	0.99152	0.73412	0.71476	0.96641	0.64242	0.93133	0.25414

Table 7: Statistical results for MS on UF1 to UF10.

## Conclusion

A new hybrid heuristic optimization algorithm is proposed for multi-objective optimization problems. The proposed hybrid algorithm formulates the optimization problems via a stochastic search method. The new hybrid algorithm is based on Nelder-Mead simplex and Non-dominated Sorting Genetic Algorithm II (NSGA II) developed to find the global optimum. The proposed hybrid scheme is noted as Simplex-NSGA II whose performance is evaluated using a series of bi-objective and tri-objective benchmarks test functions. The results are compared with those of other multi-objective optimization methods such as MOPSO and NSGA II. The numerical results demonstrate that the proposed hybrid algorithm has a greater performance in solving multi-objective optimization problems with a much better convergence and coverage capability.

## References

- Mirjalili S, Saremi S, Mirjalili SM, Coelho LD (2016) Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. *Expert Syst Appl* 47: 106-119.
- Coello CAC (2006) Evolutionary multi-objective optimization: A historical view of the field. *Comput Intell Mag IEEE* 1: 28-36.
- Kipouros T, Jaeggi DM, Dawes WN, Parks GT, Savilla M, et al. (2008) Biobjective design optimization for axial compressors using tabu search. *AIAA J* 46: 701-711.
- Luh GC, Chueh CH (2004) Multi-objective optimal design of truss structure with immune algorithm. *Comput Struct* 82: 829-844.
- Cunha AG, Covas JA (2002) RPSGAe - A multiobjective genetic algorithm with elitism: Application to polymer extrusion. *Work Mult Object Metaheuristics*.
- Rangaiah GP (2008) Multi-objective optimization: Techniques and applications in chemical engineering. *World Scientific* 1: 1.
- Coello CAC, Lamont GB (2004) Applications of multi-objective evolutionary algorithms. *World Scientific* 1.
- Le DA, Vo DN (2016) Cuckoo search algorithm for minimization of power loss and voltage deviation. *Int J energy Optim Eng* 5: 23-34.
- Vo DN (2015) Pseudo-gradient based particle swarm optimization with constriction factor for multi objective optimal power flow. *Glob J Technol Optim*.
- AIRashidi MR, El-Hawary ME (2007) Hybrid particle swarm optimization approach for solving the discrete opf problem considering the valve loading effects. *IEEE Trans Power Syst* 22: 2030-2038.
- Deb K (2001) Multi-objective optimization using evolutionary algorithms. John Wiley & Sons p: 16.
- Fonseca CM, Fleming PJ (1993) Genetic algorithms for multi-objective optimization: formulation discussion and generalization. *Icga Citeseer* 93: 416-423.
- Horn J, Nafploitis N, Goldberg D (1994) A niched pareto genetic algorithm for multi-objective optimization. *Proc. 1st IEEE Conf. Evol Comput* IEEE 82-87.
- Zitzler E, Thiele L (1998) Multi-objective optimization using evolutionary algorithms—A comparative case study. *Lect Notes Comput Sci Parallel Probl Solving from Nature—PPSN V, Springer* 1498: 292-301.
- Coello CC, Lamont GB, Van Veldhuizen DA (2007) Evolutionary algorithms for solving multi-objective problems. Springer Science & Business Media.
- Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang Q (2011) Multi-objective evolutionary algorithms: A survey of the state of the art. *Swarm Evol Comput* 1: 32-49.
- Srinivas N, Deb K (1994) Multiobjective optimization using non-dominated sorting in genetic algorithms. *Evol Comput* 2: 221-248.
- Zitzler E, Thiele L (1999) Multi-objective evolutionary algorithms: A comparative case study and the strength Pareto approach. *Evol Comput IEEE Trans* 3: 257-271.
- Zitzler E (1999) Evolutionary algorithms for multi-objective optimization: Methods and applications. *Citeseer* 63.
- Knowles JD, Corne DW (2000) Approximating the non-dominated front using the Pareto archived evolution strategy. *Evol Comput* 8: 149-172.
- Abbass HA, Sarker R, Newton C (2001) PDE: A pareto-frontier differential evolution approach for multi-objective optimization problems. *Proc Congr Evol Comput* 2: 971-978.
- Gebreslassie UMDBH (2016) Efficient ant colony optimization (EACO) algorithm for deterministic optimization. *Int J Swarm Intell Evol Comput*.
- Allah RMR (2016) Hybridization of fruit fly optimization algorithm and firefly algorithm for solving nonlinear programming problems. *Int J Swarm Intell Evol Comput*.
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6: 182-197.

25. Coello CAC, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimization. *Evol Comput IEEE Trans* 8: 256-279.
26. Zhang Q, Li H (2007) MOEA/D: A multi-objective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11: 712-731.
27. Coello A, Lechuga M (2002) MOPSO: A proposal for multi-objective swarm optimization. *Evol Comput* 2: 1051-1056.
28. Coello CA, Pulido GT, Lechuga M (2004) Handling multiple objectives with particle swarm optimization. *IEEE Trans Evol Comput* 8: 256-279.
29. Reyes-Sierra M, Coello CAC (2005) Fitness inheritance in multi-objective particle swarm optimization. *Swarm Intell Symp* 116-123.
30. Goldberg D (1989) Genetic algorithms in optimization search and machine learning. Addison Wesley 905: 205-211.
31. Goldberg D, Holland J (1988) Genetic algorithms and machine learning. *Mach Learn* 3: 95-99.
32. Pradhan PM, Panda G (2012) Solving multi-objective problems using cat swarm optimization. *Expert Syst Appl* 39: 2956-2964.
33. Velazquez OJ, Coello CC, Arias-Montano A (2014) Multi-objective compact differential evolution. *Differ Evol IEEE Symp* 1-8.
34. Hemmatian H, Fereidoon A, Assareh E (2014) Optimization of hybrid laminated composites using the multi-objective gravitational search algorithm (MOGSA). *Eng Optim* 46: 1169-1182.
35. Shi X, Kong D (2015) A multi-objective ant colony optimization algorithm based on elitist selection strategy.
36. Hancer E, Xue B, Zhang M, Karaboga D, Akay B (2015) A multi-objective artificial bee colony approach to feature selection using fuzzy mutual information. *Evol Comput (CEC) 2015 IEEE Congr* 2420-2427.
37. Lin W, Yu DY, Zhang C, Liu X, Zhang S, Tian Y, et al. (2015) A multi-objective teaching-learning-based optimization algorithm to scheduling in turning processes for minimizing make span and carbon footprint. *J Clean Prod* 101: 337-347.
38. Kitagawa G (1998) A self-organizing state-space model. *J Am Stat Assoc* 1203-1215.
39. Rao SS (1996) Engineering optimization, theory and practice 3rd. edition.
40. Zhang C, Ma X (2015) NSGA-II Algorithm with a local search strategy for multi-objective optimal design of dry-type air-core reactor. *Math Probl Eng* 839035.
41. Zhang Q, Zhou A, Zhao S (2008) Multi-objective optimization test instances for the CEC 2009 special session and competition. *Univ Essex* 1-30.
42. Sierra MR, Coello CCA (2005) Improving PSO-based multi-objective optimization using crowding, mutation and E-dominance. *Evol Multi-Criterion Optim* 505-519.
43. Schott JR (1995) Fault tolerant design using single and multi-criteria genetic algorithm optimization. DTIC Document.
44. Nebro AJ, Durillo JJ, Coello CCA (2013) Analysis of leader selection strategies in a multi-objective particle swarm optimizer. *2013 IEEE Congr Evol Comput* 3153-3160.