Decision Science Letters 9 (2020) 59–76

Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

A new non-dominated sorting ions motion algorithm: Development and applications

Hitarth Buch[a,b*] and Indrajit N Trivedi[c]

[a]*Gujarat Technological University, Visat Gandhinagar Road, Ahmedabad, 382424, India*
[b]*Government Engineering College, Mavdi Kankot Road, Rajkot, 360005, India*
[c]*Government Engineering College, 382028, Gandhinagar, 382028, India*

C H R O N I C L E

Article history:
Received March 23, 2019
Received in revised format: August 12, 2019
Accepted August 12, 2019
Available online August 12, 2019

Keywords:
Multi-objective Optimization
Non-dominated Sorting
Ions Motion algorithm

A B S T R A C T

This paper aims a novel and a useful multi-objective optimization approach named Non-Dominated Sorting Ions Motion Algorithm (NSIMO) built on the search procedure of Ions Motion Algorithm (IMO). NSIMO uses selective crowding distance and non-dominated sorting method to obtain various non-domination levels and preserve diversity amongst the best set of solutions. The suggested technique is employed to various multi-objective benchmark functions having different characteristics like convex, concave, multimodal, and discontinuous Pareto fronts. The recommended method is analyzed on different engineering problems having distinct features. The results of the proposed approach are compared with other well-regarded and novel algorithms. Furthermore, we present that the projected method is easy to implement, capable of producing a nearly true Pareto front and algorithmically inexpensive.

1. Introduction

Optimization process helps us find the best value or optimum solution. The optimization process looks for finding the minimum or maximum value for single or multiple objectives. Multi-objective optimization (MOO) refers to optimizing various objectives which are often conflicting in nature. Every day we see such problems in engineering, mathematics, economics, agriculture, politics, information technology, etc. Also sometimes, indeed, the optimum solution may not be available at all. In such cases, compromise and estimates are frequently required. Multi-objective optimization is much more complicated than single-objective optimization because of the existence of multiple optimum solutions. At large, all solutions are conflicting, and hence, a group of non-dominated solutions is required to be found out to approximate the true pareto front.

Heuristic algorithms are derivative-free solution approaches. This is because heuristic approaches do not use gradient descent to determine the global optimal. Metaheuristic approaches treat the problem as a black box for given inputs and outputs. Problem variables are inputs, while objectives are outputs. Many competent metaheuristic approaches were proposed in the past to solve the multi-objective optimization problem. A heuristic approach starts problem optimization by creating an arbitrary group of initial solutions. Every candidate solution is evaluated, objective values are observed, and based on the outputs, the candidate solutions are modified/changed/combined/evolved. This process is sustained until the end criteria are met.

* Corresponding author.
E-mail address: hitarth_buch_020@gtu.edu.in (H. Buch)

doi: 10.5267/j.dsl.2019.8.001

There are various difficulties associated while solving the problem using the heuristics. Even optimization problems have diverse characteristics. Some of the challenges are constraints, uncertainty, multiple and many objectives, dynamicity. Over a while, global optimum value changes in dynamic problems. Hence, the heuristic approach should be furnished with a suitable operator to keep track of such changes so that global optimum is not lost. Heuristic approaches should also be fault-tolerant to deal with uncertainty effectively. Constraints restrict the search space leading to viable and unviable solutions. The heuristic approach should be able to discard the unsustainable solution and ultimately discover the best optimum solution. Researchers have also proposed surrogate models to reduce computational efforts for computationally expensive functions. The idea of Pareto dominance operator is introduced to compare more than one objectives. The heuristic approach should be able to find all the best Pareto solutions. The proper mechanism should be incorporated with heuristic approaches to deal with multi-objective problems. Storage of non-dominated solutions is necessary through the optimization process. Another desired characteristics of multi-objective heuristic approach are to determine several solutions. In other words, the Pareto solutions should binge uniformly across all the objectives.

Majority of the novel single-objective algorithms have been furnished with appropriate mechanisms to deal with multi-objective problems (MOP) also. Few of them are Non-sorting Genetic Algorithm (Deb et al., 2000), Strength Pareto Evolutionary Algorithm (SPEA-II) (Zitzler et al., 2001), Multi-objective Particle Swarm Optimization (MOPSO) (Coello & Lechuga, 2002), Dragonfly Algorithm (Mirjalili, 2016), Multi-objective Jaya Algorithm (Rao et al., 2017), Multi-objective improved Teaching-Learning based Algorithm (MO-iTLBO) (Rao & Patel, 2014), Multi-objective Bat Algorithm (MOBA) (Yang, 2011), Multi-objective Ant Lion Optimizer (MOALO) (Mirjalili et al., 2017), Multi-objective Bee Algorithm (Akbari et al., 2012), Non-dominated sorting MFO (NSMFO) (Savsani & Tawhid, 2017), Multi-objective Grey Wolf Optimizer (MOGWO) (Mirjalili et al., 2016), Multi-objective Sine Cosine Algorithm (MOSCA) (Tawhid & Savsani, 2017), Multi-objective water evaporation algorithm (MOWCA) (Sadollah & Kim, 2015) and so forth.

The No Free Lunch (Wolpert & Macready, 1997) theorem (NFL) motivates to offer novel algorithms or advance the present ones since it rationally demonstrates that there is no optimization procedure which solves all problems at its best. This concept applies equally to single as well as multi-objective optimization approaches. In an exertion to solve the multi-objective optimization problem, this article suggests a multi-objective variant of the newly proposed Ions Motion Algorithm (IMO). Though the existing approaches can solve a diversity of problems, conferring to the No Free Lunch theory, current procedures may not be capable of addressing an entire range of optimization problems. This theory motivated us to offer the multi-objective IMO with the optimism to solve same or novel problems with improved efficiency.

The remaining paper is organized as follows: Section 2 discusses the existing literature. Section 3 presents the concepts of NSIMO. Section 4 discusses the current single-objective Ions Motion Algorithm (IMO) and its proposed non-sorted version. Section 5 presents deliberates and examines the results of the benchmark functions and engineering design problems. Section 6 shows a brief discussion, and finally, Section 7 accomplishes the work and offers future direction.

2. Review of Literature

In the single-objective optimization, there is a global optimum unique solution. This fact is owing to the only objective in single-objective optimization problems and the presence of the most excellent unique solution. Evaluation of solutions is simple when seeing one goal and can be completed by the relational operators: $\geq$, $>$, $\leq$, $<$, or $=$. Such problems permit optimization issues to suitably relate the aspirant solutions and ultimately determine the finest one. While in multi-objective issues, though,

solutions should be equated with multiple criteria. Multi-objective minimization problem can be expressed as follows:

Optimize (Minimize/maximize):

$$F(\vec{x}) = \left\{ f_1(\vec{x}), f_2(\vec{x}), ..., f_n(\vec{x}) \right\} \tag{1}$$

subject to:

$$a_i(x) \geq 0 \quad i = 1, 2, 3, ..., m \tag{2}$$

$$b_i(x) = 0 \quad i = 1, 2, 3, ..., p \tag{3}$$

$$L_{ib} \leq x_i \leq U_{ib} \quad i = 1, 2, 3, ..., q \tag{4}$$

Here, q presents some variables, $f_1(\vec{x}), f_2(\vec{x}), ..., f_n(\vec{x})$ introduces some objective functions, m and p shows some inequality and equality constraints. L_{ib} and U_{ib}, give lower and upper limits of the variable. The kind of such problems foils us from equating results using the relational operators as there are multiple criteria to evaluate solutions. In a single objective optimization problem, we can indeed say which solution is better using a relational operator, but with various objectives, we need some other operator(s). The primary operator to equate two solutions bearing in mind multiple goals is called Pareto optimal dominance and is described as:

The definition I: Pareto Dominance:

Assuming two vectors $\vec{a} = (a_1, a_2, ..., a_k)$ and $\vec{b} = (b_1, b_2, ..., b_k)$

Vector a dominates b ($a \prec b$) if and only if:

$$\forall i \in \{1, 2, ..., k\} : \left[f_1(\vec{a}) \leq f_1(\vec{b}) \right] \wedge \exists i \in \{1, 2, ..., k\} : \left[f_1(\vec{a}) \prec f_1(\vec{b}) \right] \tag{5}$$

By inspecting Eq. (5) it may be concluded that a solution is improved than another solution if it has equal and nonetheless, one improved value in the objectives. Under such a situation, it is said that a solution dominates another solution. If this situation does not stand good for two solutions, then they are called Pareto optimal or non-dominated solutions. The solutions to the multi-objective problem are Pareto optimal solutions. Hence, the Pareto optimality is defined as follows.

Definition II: Pareto efficiency

Assuming $a' \in A, a'$ is Pareto optimal solution if and only if:

$$\left\{ \nexists \vec{b} \in A \middle| \vec{b} \prec \vec{a} \right\} \tag{6}$$

Pareto optimality or Pareto efficiency is a state of distribution of solutions from which it is not possible to budge to achieve any one single or preference criterion improved without forming nonetheless one specific or preference criterion worse off. Such a solution set is known as the Pareto optimal set. The prognosis of the Pareto optimal solutions in the objective search space is called Pareto optimal front.

The definition III: Pareto optimal set

The Pareto optimal set comprises a set of Pareto optimal solutions.

$$PS := \left\{ \vec{a}, \vec{b} \in A \middle| \nexists \vec{b} \prec \vec{a} \right\} \tag{7}$$

The definition IV: Pareto optimal front

This set comprises objective values for the solutions in the Pareto solutions set:

$$\forall i \in \{1,2,3,...,n\}, PF := \left\{ f_i(\vec{a}) \middle| \vec{a} \in PS \right\} \tag{8}$$

Quick and easy comparison of solutions of multi-objective optimization can be made with above four equations. The group of variables, constraints, and objectives create a search landscape. Considering difficulties associated with the representation of search space for problems with more than goals, researchers consider two search space: goal and parameter space. Similarly, to single-objective optimization, the range of variables regulate the limits of the search space in each dimension while restraints divulge them. The overall outlines of all population-based multi-objective algorithms (MOAs) are nearly matching (Mirjalili et al., 2017). They begin the optimization procedure with several random candidate solutions. Such random solutions are equated utilizing the Pareto dominance operator. The algorithm attempts to enhance non-dominated solutions in the subsequent iteration(s). Different search approaches distinct one algorithm from another to augment the non-dominated solutions. Two perspectives are essential for enhancing the non-dominated solutions using stochastic algorithms: coverage (distribution) and convergence (accuracy) (Kaußler & Schmeck, 2001). The convergence denotes the procedure of refining the exactness of the non-dominated solutions. The eventual aim is to bargain estimations very near to the actual Pareto optimum solutions. Coverage presents that MOAs should attempt to increase the uniform distribution of the non-dominated solutions over the complete range of true Pareto front. For appropriate decision making, a wide range of solutions is desirable, and hence, higher coverage is a critical feature in posteriori methods. The main challenge in the stochastic multi-objective approach is the conflict between the coverage and the convergence. If an approach only focuses on enhancing the correctness of non-dominated solutions, then the resulting coverage will be weak. Or a little importance to the coverage adversely affects the efficiency of the non-dominated solutions. Majority of the existing approaches continually balance coverage and convergence to identify an exact estimate of the true solutions with a uniform spread of solutions across all objectives. The coverage can be improved by employing an archive and leader selection-based method, non-dominated sorting and niching as proposed in (Nafpliotis & Goldberg, 1994; Horn & Nafpliotis, 1993; Mahfoud, 1995).

3. Ions Motion Algorithm

This section first introduces the single-objective IMO algorithm. The next section presents the multi-objective form of single objective IMO.

3.1 Ions Motion Algorithm

In 1834, Michael Faraday coined the Greek term "ion." Typically, the charged particles are known as ions and can be separated into two categories: cations: ions with positive charge and anions: ions with a negative charge. Fig. 1 presents a conceptual model of force between cations and anions. The primary stimulus of Ions Motion Algorithm is a force of attraction and repulsion between unlike and like charges, respectively. Javidy et al. (Javidy et al., 2015) proposed the population-based IMO approach stimulated from these characteristics of ions in nature.

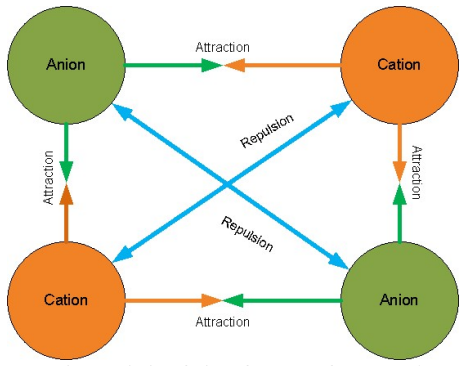


Fig. 1. Conceptual model of the force of attraction and repulsion

In IMO algorithm, anions and cations form the candidate solutions for a given optimization problem. The force of attraction/repulsion move the ions (i.e., candidate solutions) around the search space. The ions are assessed based on the fitness value of the objective function. Anions tend to move towards best cations, while cations tend to move towards best anions. This movement depends upon the force of attraction/repulsion between them. Such an approach guarantees improvement over iterations but does not guarantee the required exploration and exploitation of search space. Two different phases of ions, i.e., liquid, and crystal phase, are assumed to ensure necessary exploitation and exploration of search space.

3.1.1 Liquid phase

Liquid phase provides more freedom to the movement of ions, and hence, in the liquid stage, the ions can pass quickly. Also, the force of attraction is much more than the force of repulsion. Thus, the force of repulsion can be neglected to explore the search space. The distance between two ions is the only key factor considered to compute the force of attraction. So the resulting mathematical model can be proposed as:

$$Pf_{i,j} = \frac{1}{1+e^{-0.1/Pd_{i,j}}} \tag{9}$$

$$Qf_{i,j} = \frac{1}{1+e^{-0.1/Qd_{i,j}}} \tag{10}$$

where, $Pd_{i,j} = \left|P_{i,j} - Qbest_j\right|$ and $Qd_{i,j} = \left|Q_{i,j} - Pbest_j\right|$, i and j presents ion index and dimension respectively, $Pd_{i,j}$ is the distance between ith anion from the best cation in jth dimension, $Qd_{i,j}$ calculates the distance between ith the cation from the best anion in jth dimension. As presented in Eq. (9) and Eq. (10), force is inversely proportional to distances among ions. Larger the distance, lesser is the force of attraction. In other words, the force of attraction becomes less when the distance grows higher from the best ion with the opposite charge.

According to Eq. (9) and Eq. (10), the value of force varies between 0.5 to 1. $Pf_{i,j}$ and $Qf_{i,j}$ are the resultant attraction force of anions and cations respectively. After force calculation, the position of positive and negative ions is updated as per the following equations:

$$P_{i,j} = P_{i,j} + Pf_{i,j}\left(Qbest_j - P_{i,j}\right) \tag{11}$$

$$Q_{i,j} = Q_{i,j} + Qf_{i,j}\left(Pbest_j - Q_{i,j}\right) \tag{12}$$

$Pf_{i,j}$ and $Qf_{i,j}$ are the resulting attraction forces between opposite ions while $Qbest_j$ and $Pbest_j$ present best cations and anions respectively. The attraction force between ions guarantees exploration. Referring to Eq. (9)-(12), we can conclude that in the liquid phase, there is no involvement of the random component. Fig. 2 presents an abstract model of the movement of ions in the liquid stage. With an increasing number of iterations, more and more ions start interacting, converging towards best ion with opposite charge and hence, exploration gradually decreases. This phenomenon is precisely like the conversion from liquid to crystal state observed in nature. The search agents, i.e., ions also enter crystal state, finally converging towards the best solution in search space.

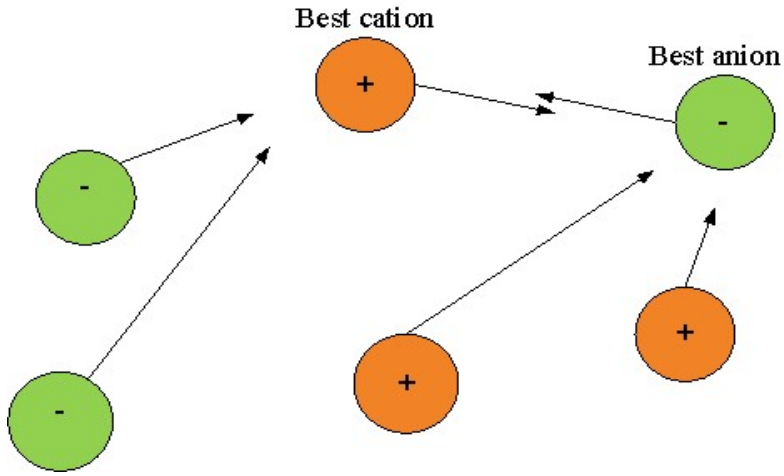


Fig. 2. Ions movement towards the best ions in the liquid phase

3.1.2 Crystal phase

In this stage, the ions congregate to the optimal solution, and convergence has already taken place. Since the search space has unknown form, occasionally convergence gets trapped into local minima. A separate mechanism is proposed at the crystal stage to avoid trapping of solutions in local minima. The cations and anions in the crystal phase are organized to maximize their force of attraction. When an outside force is applied to the same charges in the crystal phase, the resultant repulsion force cracks the crystal apart. Mathematically, the mechanism to overcome local optimum trapping can be demonstrated as below:

if (QbestFit $\geq$ QworstFit/2 and PbestFit $\geq$ PworstFit/2)
if rand () > 0.5
$P_i = P_i + \varphi_1 \times (Qbest - 1)$
else
$P_i = P_i + \varphi_1 \times (Qbest)$
end if
if rand () > 0.5
$Q_i = Q_i + \varphi_2 \times (Pbest - 1)$
else (13)
$Q_i = Q_i + \varphi_2 \times (Pbest)$
end if
if rand () < 0.5
Re-initialize P_i and Q_i
end if
end if

where, φ_1 and φ_2 are random numbers between $[-1,1]$ and $rand(\ )$ is a random number between $[0,1]$. $QbestFit$ and $QworstFit$ present fitness of the best and worst cations. $PbestFit$ and $PworstFit$ are the fitness of best and worst anions. The best fitness of anions and cations should be better than or equal to the average competence of the worst anions and cations. If this situation is met, ions are arbitrarily navigated in search space to circumvent stagnation adjacent to local minima. Again, ions enter the liquid state until termination criteria are met.

It should be noticed here that Eq. (13), four instead of two conditions are proposed to achieve different behavior of the proposed algorithm. These four conditions are presented below.

1. Both first if-else statements are met:
   $P_i = P_i + \varphi_1 \times (Qbest - 1)$

   $Q_i = Q_i + \varphi_2 \times (Pbest - 1)$

2. Only the first if-else statement is met:
   $P_i = P_i + \varphi_1 \times (Qbest - 1)$
   $Q_i = Q_i + \varphi_2 \times (Pbest)$

3. Only the second if-else statement is met:
   $P_i = P_i + \varphi_1 \times (Qbest)$
   $Q_i = Q_i + \varphi_2 \times (Pbest - 1)$

4. Both conditions are not met:
   $P_i = P_i + \varphi_1 \times (Qbest)$
   $Q_i = Q_i + \varphi_2 \times (Pbest)$

In contrast, merging the first two if-else statements will result in only two conceivable combinations:

5. Collective if-else sentences are fulfilled:
   $P_i = P_i + \varphi_1 \times (Qbest - 1)$
   $Q_i = Q_i + \varphi_2 \times (Pbest - 1)$

6. Combined if-else sentences are not fulfilled:
   $P_i = P_i + \varphi_1 \times (Qbest)$
   $Q_i = Q_i + \varphi_2 \times (Pbest)$

Thus, splitting two conditions into four provide different behavior for the IMO, which helps to avoid local optimal entrapment. Fig. 3 presents the standard steps of the Ions Motion Algorithm. The IMO starts with a random group of solutions. The arbitrary collection of solutions during initialization are generated using $r(ub_i - lb_i) + lb_i$ where r is a random number uniformly distributed between $(0,1)$. ub_i and lb_i represent upper and lower bound respectively of ith variable. At this phase, ions are equally separated into a set of anions and cations, respectively. The fitness of each anion and cation is calculated, and according to fitness, best and worst anions/cations are selected and saved. The attraction forces and positions are updated using Eq. (9) - (12) . During each iteration, if the condition of the crystal phase is met, the ions go into the crystal phase. Till the satisfaction of termination criteria, ions keep going between solid and liquid phases. In the end, the best ion is reported as the best approximation of the global solution.

4. Non-Sorted Ions Motion Algorithm (NSIMO)

In a paper on NSGA-II (Deb et al., 2002), elitist non-dominated sorting and diversity preserving crowding distance approach were introduced. The same procedure is integrated into the suggested algorithm for categorization of the population in different non-domination stages with calculated crowded distance. An elitist non-dominated sorting for finding distinct non-domination phases is defined first, and then crowding distance method to maintain the variety amongst the optimum set of solutions has been elucidated. Fig. presents population fronts established on their non-domination ranking. The green-colored solutions form the first front of non-dominated solutions as they are non-dominated by any other solutions. The orange-colored solutions form the second front as the first front

dominates them. On similar lines, all the solutions are sorted built on their non-domination level. Fig. presents a schematic representation of the non-dominated sorting-based approach for the multi-objective optimization algorithm.

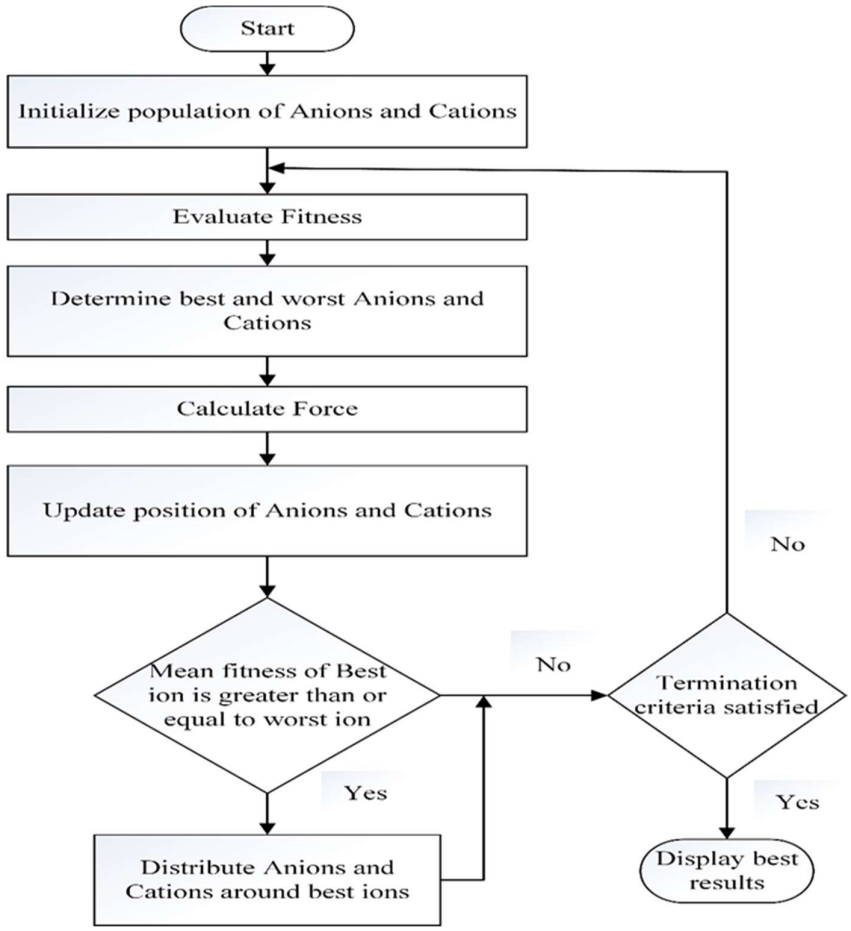


Fig. 3. Standard procedure of IMO

4.1 Diversity maintenance

The Crowding distance method is employed to maintain diversity among the acquired solutions. Initially, the population is grouped corresponding to the fitness of objective function. The boundary solutions are assigned an infinite crowding distance. In Fig. 5, solution a and b are attached infinite crowding distance. Except for boundary solutions, all other solutions are assigned crowding distance as:

$$CD_j^i = \frac{f_j^{i+1} - f_j^{i-1}}{f_j^{\max} - f_j^{\min}} \tag{14}$$

$f_j^{\min}$ and $f_j^{\max}$ are minimum and maximum values of jth objective function. Once, all the population members are assigned crowding distance; any two solutions can be compared for their extent of proximity with other solutions. A solution with a smaller value of CD, in some sense, is more surrounded by other solutions. For uniform spread out of Pareto optimal front, crowded comparison operator $(\prec_n)$ is used in the different stage of the algorithm to guide the selection process. As mentioned earlier, every solution i has two attributes:

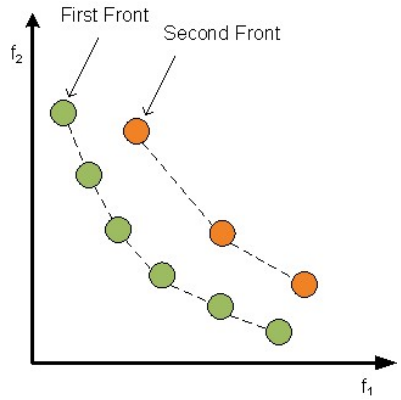


Fig. 4. Diagram of non-dominated sorting

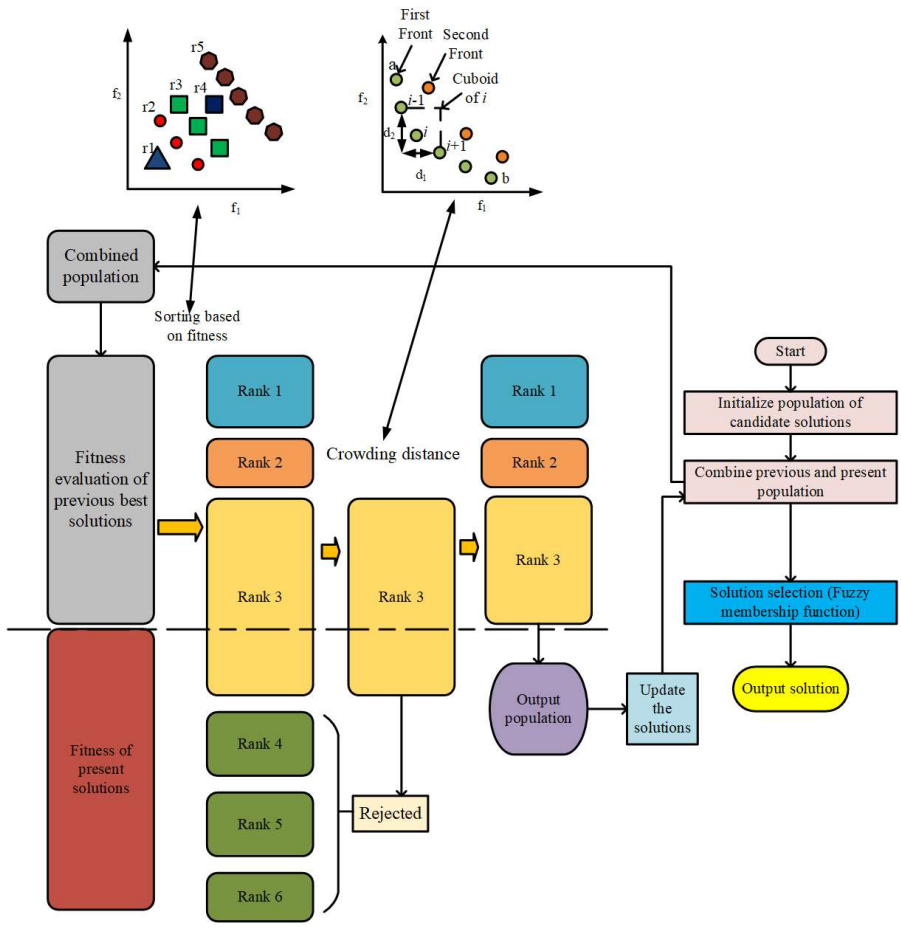


Fig. 5. Schematic representation of non-dominated sorting based algorithm

- Non-domination rank
- Crowding distance

For solutions having a different non-domination level, we prefer solutions with better (lower) rank. If both the solutions belong to the same front, the solution located in the less crowded region is selected.

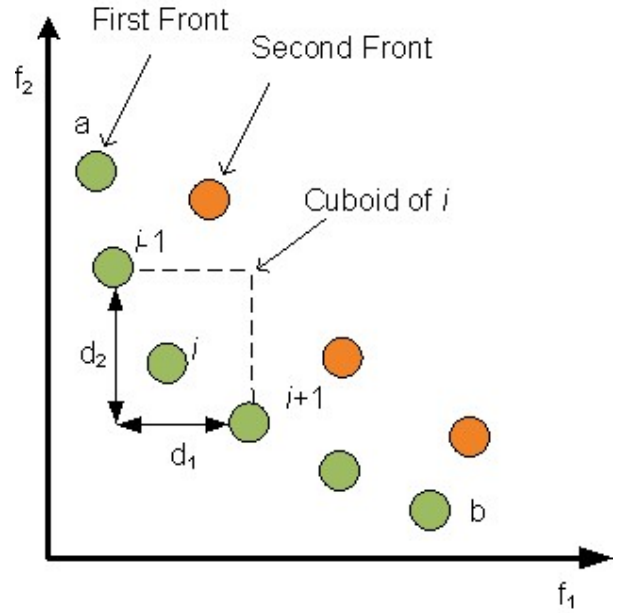


Fig. 6. Diagram of crowding distance approach

5. Simulation Results

This section describes the performance of the suggested algorithm on 20 case studies considering eight unconstrained, six constrained, and six engineering design problems (Mirjalili et al., 2017). The performance of NSIMO is tested on benchmark functions having different Pareto optimal front, i.e., diverse characteristics. To further check the performance of the algorithm, more challenging real-time engineering design problems are also considered.

For results confirmation, two recently developed non-sorted algorithms, such as MOSCA (Tawhid & Savsani, 2017) and NSMFO (Savsani & Tawhid, 2017), are used. The findings are gathered and discussed quantitatively and qualitatively in this section. Each algorithm is run 30 times. Note that we have used 500 iterations and 200 search agents. Best Pareto fronts obtained by the algorithms are compared for the qualitative findings. For the quantifiable results, we have used a variety of performance metrics: Generational Distance (GD) (Veldhuizen & Lamont, 1998), Inverted Generational Distance (IGD) (Sierra & Coello, 2005), metric of spread (Deb, 2001), and metric of spacing (Schott, 1995).

5.1 Findings on unconstrained benchmark test problems

Eight different unconstrained benchmark functions, i.e., KUR, FON, ZDT1, ZDT2, ZDT3, ZDT6, SCHN1, and SCHN2 are employed to assess the performance of the NSIMO. Table 1 and Fig. gives a quantitative and qualitative assessment of results obtained by different algorithms. Results suggest that the NSIMO performs better or competitive as compared to the rest of the algorithms under consideration.

The statistical results present that NSIMO algorithm performs better than NSSCA algorithm significantly on many unconstrained test functions. These results display the superiority of NSIMO showing higher accuracy and better robustness. For the rest of the benchmark functions, it performs competitively if not better. The NSIMO algorithm, however, presents incredibly competitive outcomes in parallel with the NSMFO algorithm and occasionally outperforms it.

The shape of the best Pareto front achieved by the three procedures on different unconstrained benchmark functions is illustrated in Fig. . Reviewing these figures, NSSCA presents poor

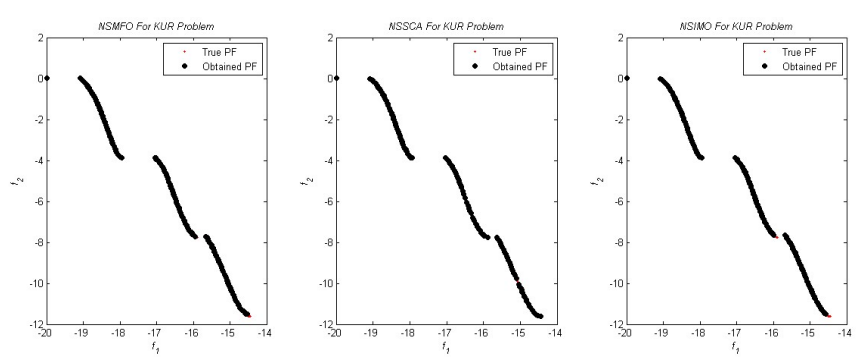
performance, notwithstanding its good coverage in a few cases. However, NSMFO and NSIMO both provide a perfect convergence toward all true Pareto fronts. Performance of NSIMO is better than the rest of the algorithms for ZDT6. This shows that NSIMO can outclass the rest of the algorithms in getting Pareto optimal front with non-convex, non-uniform regions.

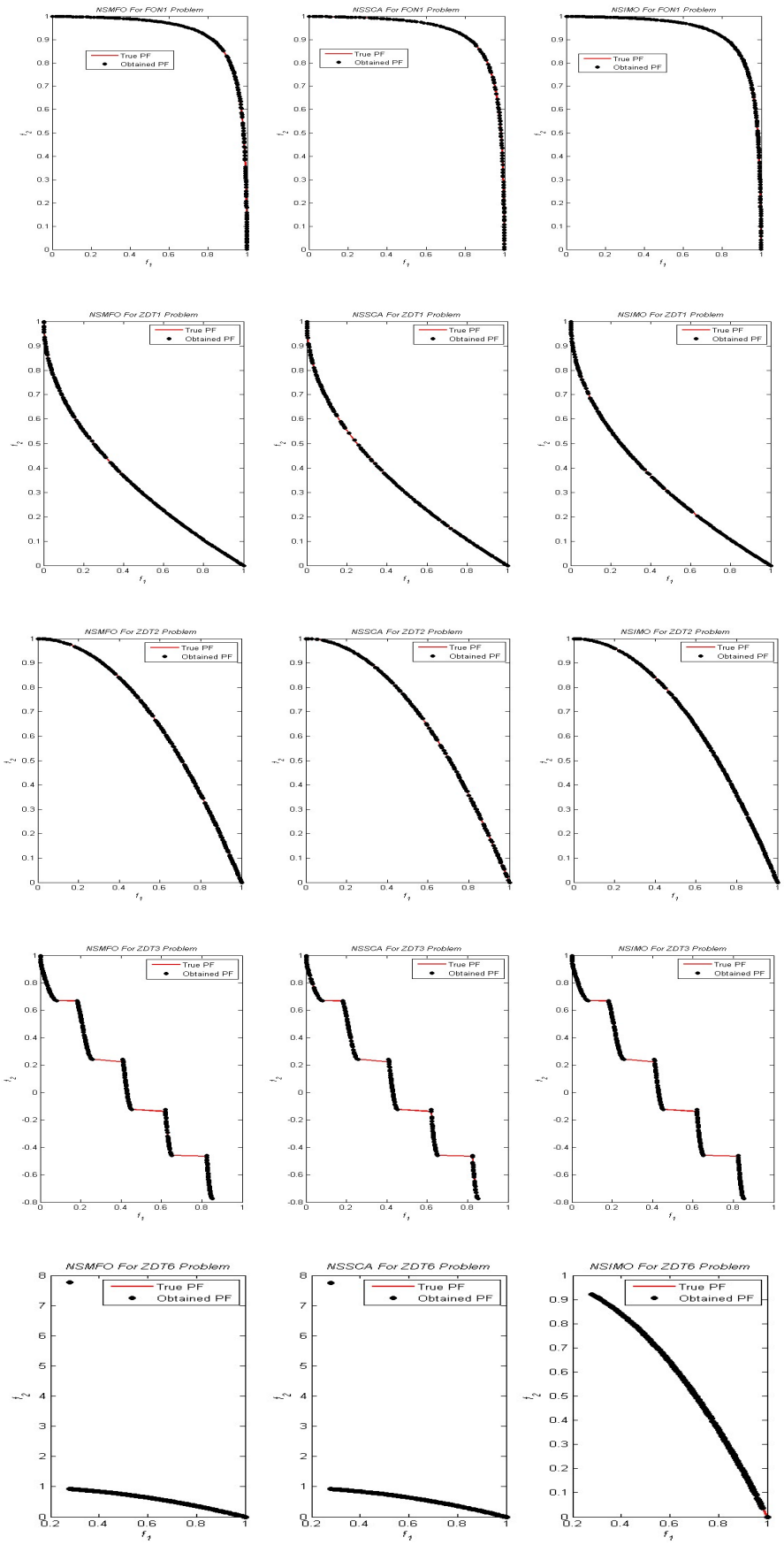
5.2 Results on constrained test problems

The next set of test function comprises constrained benchmark functions. We need to arm NSIMO with a constraint handling method to be able to solve constrained problems. Identifying an appropriate constraint handling method is beyond the scope of this effort, and we use a death penalty function (Mirjalili et al., 2017) to punish search agents that infringe any of the restraints at any level. For equating algorithms, we have applied four metrics in this research: GD, IGD, metric of spread, and metric of space. These performance indicators allow us to enumerate and equate algorithms regarding convergence and coverage.

Table 1
Results (IGD, Metric of Spread and GD) on Unconstrained Multi-Objective Problems

Algorithm→ **Functions ↓**	**PMs**	**NSMFO MEAN±SD**	**NSSCA MEAN±SD**	**NSIMO MEAN±SD**
KUR	GD	**0.0001074±1.5063e-05**	0.00010377±4.7693e-06	0.00011763±1.8943e-05
	Δ	0.44714±**0.016241**	0.66507±0.017925	**0.43762**±0.019637
	IGD	**0.00012776±2.766e-05**	0.00011033±4.3825e-06	0.00012798±4.4222e-05
FON	GD	8.8423e-05±3.8647e-06	**8.5355e-05**±2.9712e-06	8.7132e-05±**1.5134e-06**
	Δ	0.31999±0.022165	0.3955±**0.0068**	**0.31494**±0.018985
	IGD	0.00014599±**5.9036e-06**	0.00019748±9.581e-06	**0.00014514**±6.1866e-06
ZDT-1	GD	0.00016787±1.4753e-05	**0.00015976**±1.7683e-05	0.00016875±**1.4248e-05**
	Δ	**0.36628**±0.01978	0.64984±0.022358	0.37601±**0.019666**
	IGD	**0.00013269**±8.5614e-06	0.00017428±1.0992e-05	0.0001368±**7.7013e-06**
ZDT-2	GD	6.7703e-05±2.1246e-06	**6.6127e-05**±2.5689e-06	6.6268e-05±2.287e-06
	Δ	0.37941±0.017908	0.6569±0.023443	**0.37587**±0.02307
	IGD	0.00014449±1.2209e-05	0.00017878±1.4167e-05	**0.00013973±8.3648e-06**
ZDT-3	GD	**0.00025146**±5.9607e-06	0.00024917±7.1797e-06	0.00024978±**5.8043e-06**
	Δ	**0.55962**±0.012548	0.70387±0.020547	0.56258±**0.012206**
	IGD	0.00016883±1.017e-05	0.00022276±1.6369e-05	**0.00016749±7.7892e-06**
ZDT-6	GD	0.010417±**0.016865**	0.0090416±0.019254	**0.005628**±0.017609
	Δ	0.65378±0.47981	0.81997±**0.29371**	**0.46549**±0.31668
	IGD	**0.0001273±8.0205e-06**	0.00017561±1.4624e-05	0.00013057±1.0637e-05
SCHN-1	GD	0.0001638±**6.9129e-06**	0.00016785±7.8278e-06	**0.00016383**±7.159e-06
	Δ	0.62451±0.034387	0.59813±0.02761	**0.37422±0.017477**
	IGD	**0.00023106**±1.1999e-05	0.00028302±**1.5164e-05**	0.00023399±1.1998e-05
SCHN-2	GD	**1.854e-05**±5.6619e-07	1.8588e-05±7.3537e-07	1.8666e-05±**5.4382e-07**
	Δ	1.0071±0.013382	1.1039±0.017301	**1.0056±0.011871**
	IGD	**6.2055e-05±8.415e-07**	8.7834e-05±7.0331e-06	6.2688e-05±2.5361e-06




NSMFO For FON1 Problem
NSSCA For FON1 Problem
NSIMO For FON1 Problem
NSMFO For ZDT1 Problem
NSSCA For ZDT1 Problem
NSIMO For ZDT1 Problem
NSMFO For ZDT2 Problem
NSSCA For ZDT2 Problem
NSIMO For ZDT2 Problem
NSMFO For ZDT3 Problem
NSSCA For ZDT3 Problem
NSIMO For ZDT3 Problem
NSMFO For ZDT6 Problem
NSSCA For ZDT6 Problem
NSIMO For ZDT6 Problem
True PF
Obtained PF
f_1
f_2

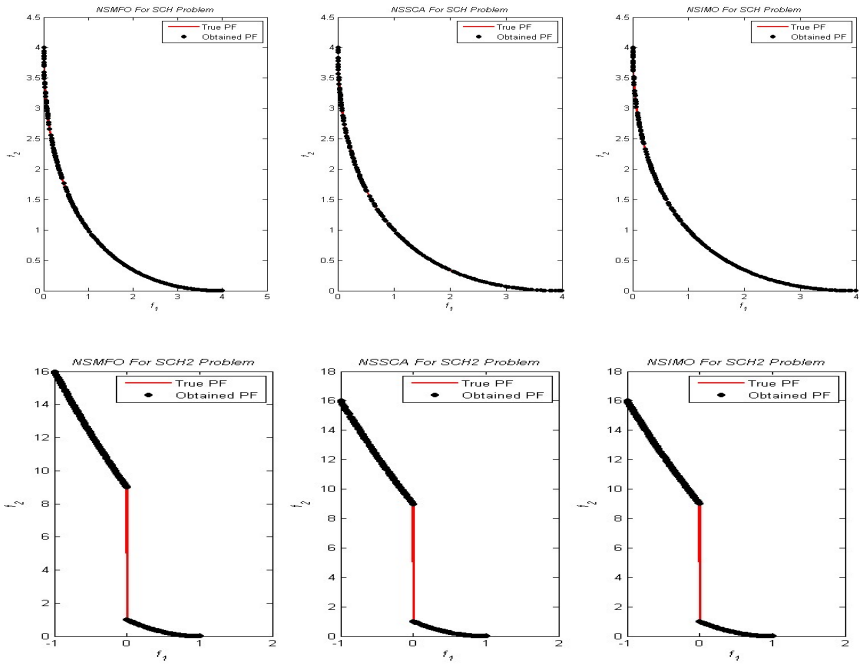


Fig. 7. Best Pareto optimal front of KUR, FON, ZDT1, ZDT2, ZDT3, ZDT4, SCHN1 and SCHN2 obtained by the NSMFO, NSSCA, and NSIMO

Table 2
Results (IGD, Metric of Spread, GD, and Metric of Spacing) on Constrained Multi-Objective Problems

Algorithm→ Functions ↓	PMs	NSMFO MEAN±SD	NSSCA MEAN±SD	NSIMO MEAN±SD
SRN	GD	6.5352e-05±1.522e-05	**5.1191e-05**±1.5714e-05	5.8776e-05±**9.8695e-06**
	IGD	4.6713e-05±**1.881e-06**	6.4209e-05±2.6541e-06	**4.5476e-05**±1.9791e-06
	Δ	0.33905±0.017063	0.68601±**0.01411**	**0.32752**±0.015569
	MoS	1.7525±0.27734	**1.8534**±**0.26276**	1.6097±0.33944
CONSTR	GD	0.00015872± 8.4222e-06	0.00016676± 1.3434e-05	**0.0001568**± **9.417e-06**
	IGD	0.00013559± 1.2016e-05	0.00016462± **6.6251e-06**	**0.0001353**± 9.3247e-06
	Δ	**0.58274**± 0.019573	0.83056± **0.019114**	0.58807± 0.021356
	MoS	**2.5963**±**0.16907**	2.3987±0.25002	2.5092±0.23515
OSY	GD	0.0066976± 0.00085347	**0.0044044**±**0.00045**	0.0063532±0.0010256
	IGD	0.0065997± 0.00091147	**0.0054844**±**0.00084982**	0.0068945±0.0013187
	Δ	0.79921±0.0754	0.85176±**0.024854**	**0.76545**±0.06933
	MoS	**42.2379**±5.2723	41.6241±**3.9097**	41.5859±3.9256
CF1	GD	0.0014345±**1.4957e-05**	0.0014326±3.5153e-05	**0.0014272**±1.809e-05
	IGD	**0.00056284**±6.5769e-05	0.00083561±0.00027776	0.00057587±**6.4474e-05**
	Δ	**0.26101**±**0.017055**	0.70089±0.031555	0.26582±0.019673
	MoS	0.0018384±0.00022823	0.00095134±0.00020729	**0.0018453**±**0.00017062**
BEL	GD	**2.9344e-05**±**2.4679e-06**	3.3158e-05±5.4287e-06	3.005e-05±3.1562e-06
	IGD	**8.8872e-05**±**2.8416e-06**	0.00012607±5.1487e-06	9.0167e-05±4.6455e-06
	Δ	0.3676±**0.012431**	0.71218±0.02448	**0.36587**±0.024505
	MoS	0.0011919±**0.0016248**	**0.0036968**±0.0035206	0.0016076±0.0019559
BNH	GD	**0.00021099**±6.0144e-06	0.0002143±**5.8956e-06**	0.00021325±1.1498e-05
	IGD	0.00023224±**1.189e-05**	0.00029761±2.208e-05	**0.00022943**±1.298e-05
	Δ	0.45494±**0.021187**	0.66419±0.028024	**0.45656**±0.027193
	MoS	**31.8377**±**1.4852**	30.8488±2.4398	31.1807±2.1131

Table 2 demonstrates that the NSIMO outperforms the other two algorithms on most of the constrained test functions used. Results of GD and IGD presents competitive convergence. The results collected

shows that the NSIMO algorithm beats the NSMFO and NSSCA algorithm. The best Pareto optimal fronts in Fig. also substantiates this claim since all the Pareto optimal solutions found by NSIMO are positioned on the front. The results for coverage performance metrics in Table 2 also prove that NSIMO shows better results in many instances. Though they all are very competitive with each other. Fig. demonstrates that some of the constrained test functions have very different Pareto fronts compared to unconstrained test functions, for example, BNH, CONSTR, and OSY. CONSTR has non-convex front along with a linear front. Results for NSIMO achieved to estimate both these parts efficiently. NSIMO almost wholly obtains SRN, BEL, and BNH. NSSCA performs improved than the rest of the two algorithms for OSY function. OSY function is quite like CONSTR function with multiple linear regions in between.

5.3 Results on Engineering Design Problems

The final group of benchmark functions is the most stimulating and comprises six real engineering design problems. These problems have varied constrained characteristics. Table 3 quantitatively compares different algorithms for a comparable group of performance metrics. Table 3 presents that NSIMO algorithm can very well solve the engineering design problems. The best Pareto optimal fronts in Fig. 9, however, offer different conduct from the other two test suites. Fig. 9 presents that the convergence of the NSIMO algorithm is not 100% near to the true Pareto front in the welded beam, speed reducer, and Isolated safety transformer design problems. This is owing to the multi-modal and highly constrained nature of the problem. Despite this, the convergence is rational, and the coverage is exceptional and practically even.

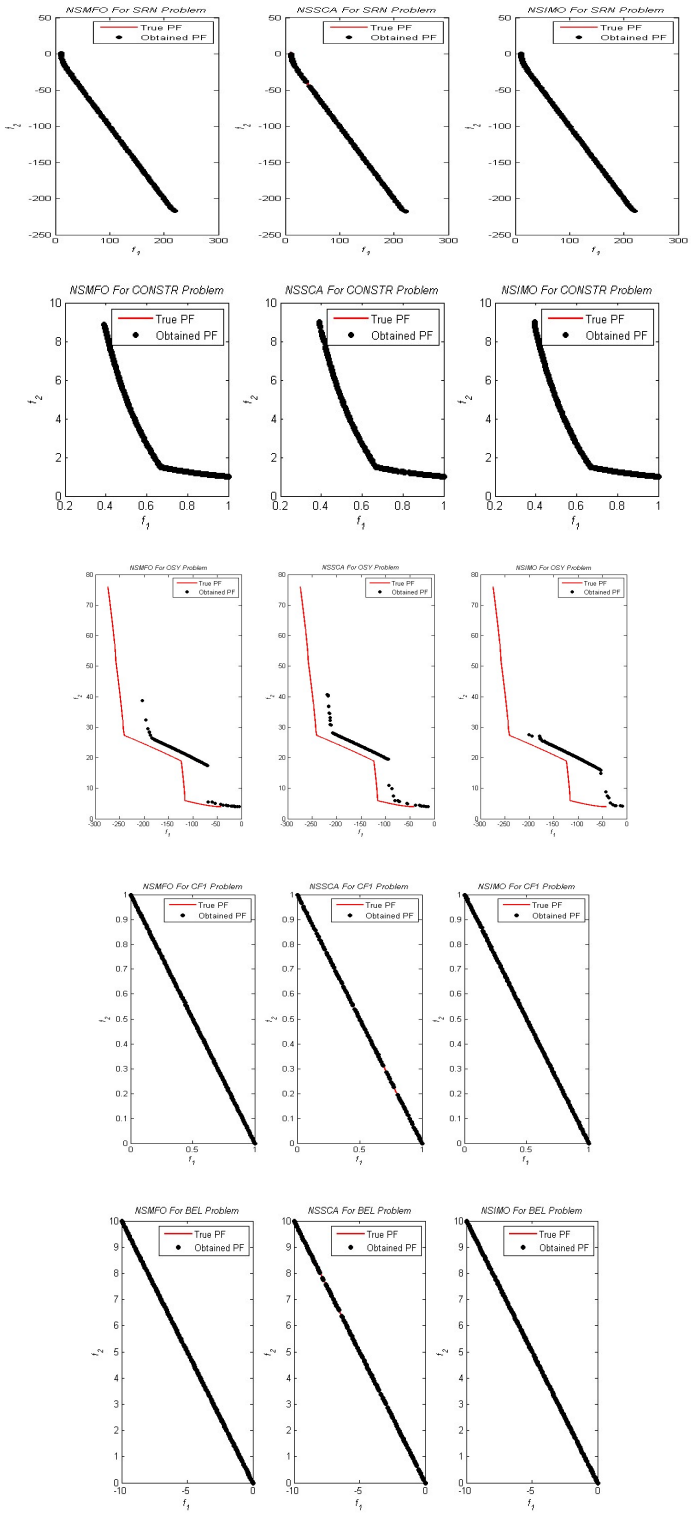
Table 3
Results on Constrained Real-World Engineering Multi-Objective Problems

Algorithm→ **Functions ↓**	**PMs**	**NSMFO** **MEAN±SD**	**NSSCA** **MEAN±SD**	**NSIMO** **MEAN±SD**
Disk Brake	GD	0.00021217±9.2288e-06	0.00021557±**8.665e-06**	**0.00018101**±3.4774e-05
	IGD	0.00019922±2.4627e-05	**0.00018747**±**8.7678e-06**	0.00027207±0.00016832
	Δ	0.67008±**0.032321**	0.76739±0.034974	**0.6318**±0.041105
	MoS	3.6165±0.34099	**4.0017**±**0.21541**	3.4855±0.32762
Four Bar Truss	GD	0.0024977± 0.0030875	**0.00022869**± **1.0134e-05**	0.001247± 0.0016354
	IGD	**0.00021388**± **8.1162e-06**	0.00026617± 1.3e-05	0.0002165± 1.1761e-05
	Δ	0.70854± 0.26493	0.68294± **0.024525**	**0.60075**± 0.21382
	MoS	**182.5304**± 11.4529	168.7437± **10.66**	182.1619± 11.2816
Isolated Transformer	GD	**0.00069302**± **0.00014258**	0.00083383±0.00050994	0.0012147±0.00079664
	IGD	0.0013891± 0.00049654	**0.00037493**±**0.00023764**	0.0021309±0.0008391
	Δ	**0.57726**±0.069078	0.76167±**0.032098**	0.64833±0.10209
	MoS	3.9591±0.6307	**5.4163**±**0.50189**	3.3525±0.67109
Pressure Vessel	GD	0.0002808±2.8963e-05	**5.1166e-05**±**8.9434e-06**	0.0011584±0.00087485
	IGD	9.5019e-05±7.2848e-06	**8.7491e-05**±**3.9046e-06**	0.0001398±3.0382e-05
	Δ	0.34083±0.025803	**0.67829**±**0.021022**	0.37779±0.02608
Speed Reducer	GD	0.00038333±0.00011586	**0.00033068**±**9.4532e-05**	0.00048365±0.0001909
	IGD	0.0025719±0.0035815	**0.00036263**±**3.232e-05**	0.0020127±0.001675
	Δ	0.54322±0.11131	**0.76304**±**0.026699**	0.54283±0.096924
Welded Beam	GD	0.0011713±0.00091419	0.0033673±0.0039845	**0.0010877**±**0.00081558**
	IGD	0.007178±0.0026728	**0.0031542**±**0.0026112**	0.0078412±0.0030566
	Δ	**0.50927**±0.062032	0.80993±0.0749	0.54345±**0.060318**
	MoS	10.3228±**0.60164**	**10.4591**±0.6809	10.1706±0.93415

6. Discussion

The quantitative and qualitative outcomes presented that the NSIMO algorithm aids from competitive convergence and coverage. Competitive convergence of NSIMO is congenital from the IMO algorithm.

The principal mechanism that assures convergence in IMO and NSIMO is owing to the proposed liquid phase where the ions tend to get attracted towards best anions and cations. It was also observed and demonstrated that competitive coverage is another gain of the NSIMO algorithm. The competitive coverage initiates from the exit from local optima by the repulsion forces in crystal phase and crowding selection mechanism. Non-dominated sorting approach and crowding distance mechanism guaranteed the selection of the best solution and desired spread during each iteration. These mechanisms emphasize improving the coverage and convergence of the Pareto optimal front achieved throughout the optimization process.



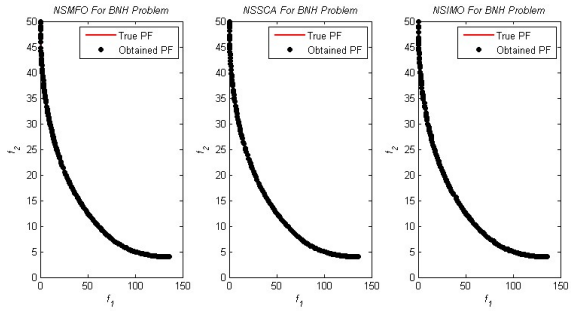
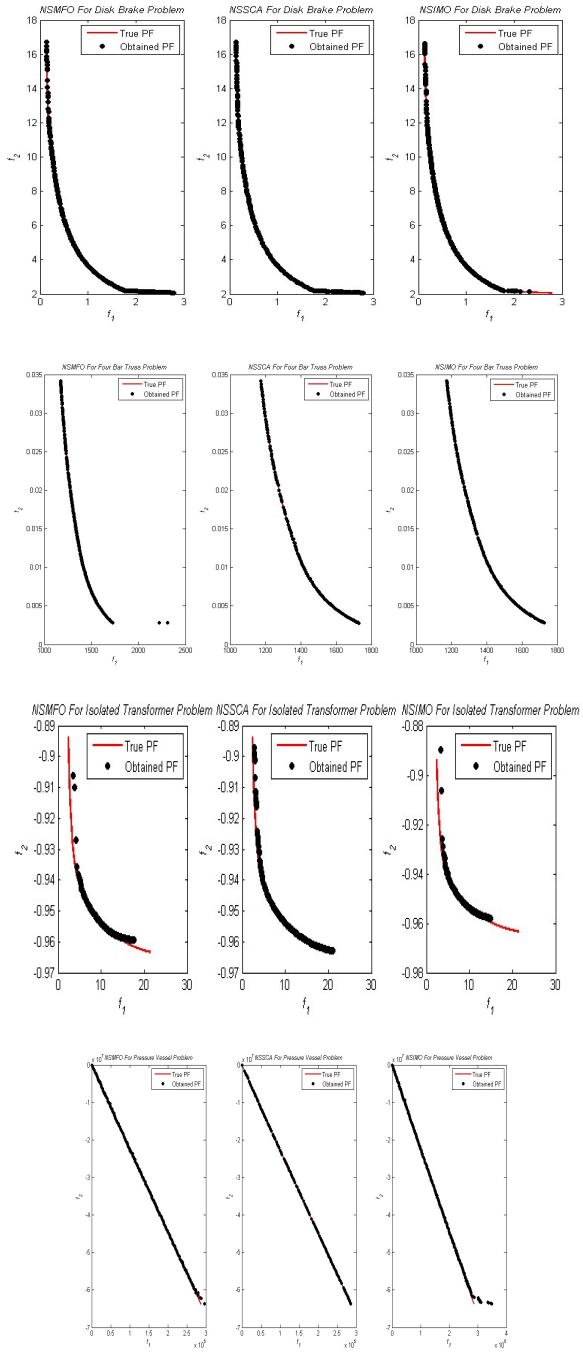


Fig. 8. Best Pareto optimal front of SRN, CONSTR, OSY, CF1, BEL, and BNH obtained by the NSMFO, NSSCA, and NSIMO



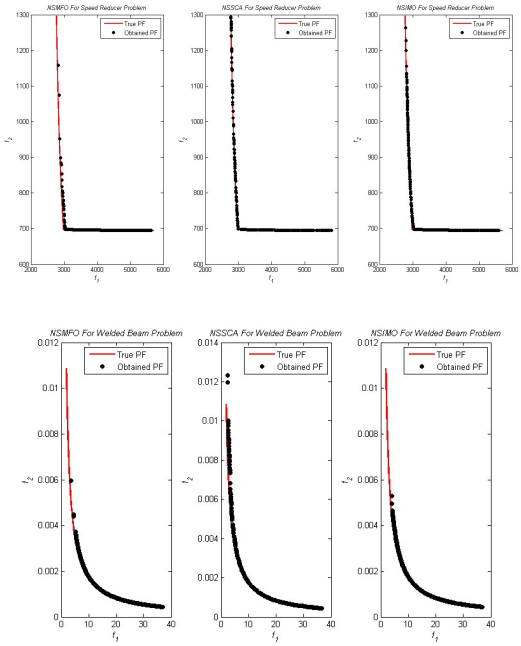


Fig. 9. Best Pareto optimal front of Real World Engineering Problems obtained by the NSMFO, NSSCA, and NSIMO

7. Conclusion

This work suggested the multi-objective variant of the newly introduced IMO algorithm named NSIMO. With preserving the critical search mechanism of IMO, NSIMO was intended with arming IMO with non-dominated sorting mechanism and crowding distance mechanism. The algorithm was examined on 20 case studies, including eight unconstrained, six constrained, and six engineering design benchmark problems. The measurable results were compared using four performance parameters: Generational Distance (GD), Inverse Generational Distance (IGD), Metric of Spread, and Metric of Spacing. Likewise, qualitative results were testified as for the best Pareto optimal obtained in 30 runs. For results validation, the projected algorithm is compared to the recently introduced algorithms: MOSCA and NSMFO. The results presented that the NSIMO can outperform/provide very competitive findings compared to the rest of the algorithms. It was found that NSIMO has high coverage and convergence as well. The test functions used are diverse and have varied Pareto optimal fronts. The outcomes exhibited that NSIMO could achieve Pareto optimal front almost of any form. Lastly, the findings of engineering design problems validated that NSIMO is proficient in solving stimulating problems with numerous constraints. Hence, we accomplish that the projected approach has virtues amongst the contemporary multi-objective algorithms and recommend it as a substitute for solving multi-objective optimization problems. For forthcoming works, it is recommended to apply NSIMO to other engineering design problems, development of archive-based IMO, and its comparison with NSIMO. Also, it is worth to examine and identify the best-constrained handling method for this approach.

Funding: The authors declare that they have not received any funding.

Conflict of Interest: The authors declare that they have no conflict of interest.

References

Akbari, R., Hedayatzadeh, R., Ziarati, K., & Hassanizadeh, B. (2012). A multi-objective artificial bee colony algorithm. *Swarm and Evolutionary Computation*, *2*, 39–52.

Branke, J., Kaußler, T., & Schmeck, H. (2001). Guidance in evolutionary multi-objective optimization. *Advances in Engineering Software*, *32*(6), 499–507.

Coello Coello, C. A., & Lechuga, M. S. (2002). MOPSO: A proposal for multiple objective particle swarm optimization. *Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002*, *2*, 1051–1056.

Deb, K. (2001). Multi-Objective Optimization Using Evolutionary Algorithms. *John Wiley & Sons, LTD*, p. 497.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Parallel Problem Solving from Nature PPSN VI*, 849–858.

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

Horn, J, Nafpliotis, N., & Goldberg, D. E. (1994). A niched Pareto genetic algorithm for multiobjective optimization. *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference On*, *1*, 82–87.

Horn, Jeffrey, & Nafpliotis, N. (1993). Multiobjective Optimization Using The Niched Pareto Genetic Algorithm. *IlliGAL Report No. 93005*, 82–87.

Javidy, B., Hatamlou, A., & Mirjalili, S. (2015). Ions motion algorithm for solving optimization problems. *Applied Soft Computing Journal*, *32*, 72–79.

Mahfoud, S. (1995). Niching methods for genetic algorithms. *Urbana*, (95001), 251.

Mirjalili, S. (2016). Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, *27*(4), 1053–1073.

Mirjalili, S., Jangir, P., Mirjalili, S. Z., Saremi, S., & Trivedi, I. N. (2017). Optimization of problems with multiple objectives using the multi-verse optimization algorithm. *Knowledge-Based Systems*. https://doi.org/10.1016/j.knosys.2017.07.018

Mirjalili, S., Jangir, P., & Saremi, S. (2017). Multi-objective ant lion optimizer: A multi-objective optimization algorithm for solving engineering problems. *Applied Intelligence*, *46*(1), 79–95.

Mirjalili, S. M., Saremi, S., Mirjalili, S. M., & Coelho, L. D. S. (2016). Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. *Expert Systems with Applications*, *47*, 106–119.

Rao, R. V., Rai, D. P., & Balic, J. (2017). A multi-objective algorithm for optimization of modern machining processes. *Engineering Applications of Artificial Intelligence*, *61*, 103–125.

Sadollah, A., Eskandar, H., & Kim, J. H. (2015). Water cycle algorithm for solving constrained multi-objective optimization problems. *Applied Soft Computing*, *27*, 279–298.

Savsani, V., & Tawhid, M. A. (2017). Non-dominated sorting moth flame optimization (NS-MFO) for multi-objective problems. *Engineering Applications of Artificial Intelligence*, *63*, 20–32.

Schott, J. R. (1995). Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization.

Sierra, M. R., & Coello Coello, C. A. (2005). *Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and ∈-Dominance*. https://doi.org/10.1007/978-3-540-31880-4_35

Tawhid, M. A., & Savsani, V. (2017). Multi-objective sine-cosine algorithm (MO-SCA) for multi-objective engineering design problems. *Neural Computing and Applications*, 1–15.

Veldhuizen, D. A. Van, Van Veldhuizen, D. A., & Lamont, G. B. (1998). *Multiobjective Evolutionary Algorithm Research: A History and Analysis*.

Venkata Rao, R., & Patel, V. (2014). A multi-objective improved teaching-learning based optimization algorithm for unconstrained and constrained optimization problems. *International Journal of Industrial Engineering Computations*, 1–22.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82.

Yang, X.-S. (2011). Bat algorithm for multi-objective optimisation. *International Journal of Bio-Inspired Computation*, *3*(5), 267–274. https://doi.org/10.1504/IJBIC.2011.042259

Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-Report*, *103*.