Global Journal of Researches in Engineering: F
Electrical and Electronics Engineering
Volume 20 Issue 2 Version 1.0 Year 2020
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 2249-4596 & Print ISSN: 0975-5861

MONSDA: - A Novel Multi-Objective Non-Dominated Sorting Dragonfly Algorithm

By Pradeep Jangir

Abstract- This novel article presents the multi-objective version of the recently proposed Dragonfly Algorithm (DA) known as Non-Dominated Sorting Dragonfly Algorithm (NSDA). This proposed NSDA algorithm works in such a manner that it first collects all non-dominated Pareto optimal solutions in achieve till the evolution of last iteration limit. The best solutions are then chosen from the collection of all Pareto optimal solutions using a crowding distance mechanism based on the coverage of solutions and swarming strategy to guide dragonflies towards the dominated regions of multi-objective search spaces. For validate the efficiency and effectiveness of proposed NSDA algorithm is applied to a set of standard unconstrained, constrained and engineering design problems. The results are verified by comparing NSDA algorithm against Multi objective Colliding Bodies Optimizer (MOCBO), Multi objective Particle Swarm Optimizer (MOPSO), non-dominated sorting genetic algorithm II (NSGA-II) and Multi objective Symbiotic Organism Search (MOSOS).The results of proposed NSDA algorithm validates its efficiency in terms of Execution Time (ET) and effectiveness in terms of Generalized Distance (GD), Diversity Metric (DM) on standard unconstraint, constraint and engineering design problem in terms of high coverage and faster convergence.

Keywords: non-dominated; crowing distance; NSDA algorithm; multi-objective algorithm; economic constrained emission dispatch.

GJRE-F Classification: FOR Code: 290901p

MONSDAANOVELMULTIOBJECTIVENONDOMINATEDSORTINGDRAGONFLYALGORITHM

Strictly as per the compliance and regulations of:

MONSDA: - A Novel Multi-Objective Non-Dominated Sorting Dragonfly Algorithm

Pradeep Jangir

Abstract- This novel article presents the multi-objective version of the recently proposed Dragonfly Algorithm (DA) known as Non-Dominated Sorting Dragonfly Algorithm (NSDA). This proposed NSDA algorithm works in such a manner that it first collects all non-dominated Pareto optimal solutions in achieve till the evolution of last iteration limit. The best solutions are then chosen from the collection of all Pareto optimal solutions using a crowding distance mechanism based on the coverage of solutions and swarming strategy to guide dragonflies towards the dominated regions of multi-objective search spaces. For validate the efficiency and effectiveness of proposed NSDA algorithm is applied to a set of standard unconstrained, constrained and engineering design problems. The results are verified by comparing NSDA algorithm against Multi objective Colliding Bodies Optimizer (MOCBO), Multi objective Particle Swarm Optimizer (MOPSO), non-dominated sorting genetic algorithm II (NSGA-II) and Multi objective Symbiotic Organism Search (MOSOS).The results of proposed NSDA algorithm validates its efficiency in terms of Execution Time (ET) and effectiveness in terms of Generalized Distance (GD), Diversity Metric (DM) on standard unconstraint, constraint and engineering design problem in terms of high coverage and faster convergence.

Keywords: non-dominated; crowing distance; NSDA algorithm; multi-objective algorithm; economic constrained emission dispatch.

I. Introduction

Optimization is a work of achieving the best result under given limitation or constraints. Now a day, optimization is used in all the fields like construction, manufacturing, controlling, decision making, prediction etc. The final target is always to get feasible solution with minimum use of resources. In this field computers make a revolutionary impact on every field as it provides the facility of virtual testing of all parameters that are involved in a particular design with less involvement of human efforts, benefits in less time consuming, human efforts and wealth as well.

Today we use computer-aided design where a designer designs a virtual system on computer and gives only command to test all parameters involved in that design without even the need for a single prototype. A designer only to design and simulate a system and set all the parameter limitation for the computer.

Computer-aided design technique becomes more effective with the additional feature of auto-generation of solutions after it's mathematically formulation of any system or design problem. Auto generation of solution, this feature is come into nature with the development of algorithms. In past years, real world designing problems are solved by gradient descent optimization algorithms. In gradient descent optimization algorithm, the solution of mathematically formulated problem is achieved by obtaining its derivative. This technique is suffered from local minima stagnation [1, 2] more time consuming and their solution is highly dependent on their initial solution.

The next stage of development of optimization algorithms is population based stochastic algorithms. These algorithms had number of solutions at a time so embedded with a unique feature of local minima avoidance. Later population based algorithms are developed to solve single objective at a time either it may be maximization or minimization on accordance the problems objective function. Some popular algorithms for single objective problems are Moth-Flame optimizer (MFO) [3], Bat algorithm (BA) [4], Particle swarm optimization (PSO) [5], Ant colony optimization (ACO) [6], Genetic algorithm (GA) [7], Cuckoo search (CS)[8], Mine blast algorithm (MBA) [9], Krill Herd (KH) [10], Interior search algorithm (ISA) [11] etc. These algorithms have capabilities to handle uncertainties [12], local minima [13], misleading global solutions [14], better constraints handling [15] etc. To overcome these difficulties different algorithms are enabled with different powerful operators. As mention above here is only objective then it is easy to measure the performance in terms of speed, accuracy, efficiency etc. with the simple operational operators.

In general, real world problems are nonlinear and multi-objective in nature. In multi-objective problem there may be some objectives are consisting of maximization function while some are minimization function. So now a day, multi-objective algorithms are in firm attention.

Let's take an example of buying a car, so we have many objectives in mind like speed, cost, comfort level, space for number of people riding, average fuel consumption, pick up time required to gain particular speed, type of fuel requirement either it is diesel driven, petrol driven or both etc. To simply understand multi-objective problem, from Fig. 1, we consider two objectives, first cost and second comfort level. So we go for sole objective of minimum cost possible then we

Author: Junior Engineer Rajasthan Rajya Vidyut Prasaran Nigam, 132 KV GSS, Losal-332025, Sikar, Rajasthan, India.
e-mail: pkjmtech@gmail.com

have to deny comfort level objective and vice-versa. It means real word problems are with conflicting objectives. So as, we are disabled to find an optimal solution like single objective problems. About multi-objective algorithm and its working is detailed described in next portion of the article.

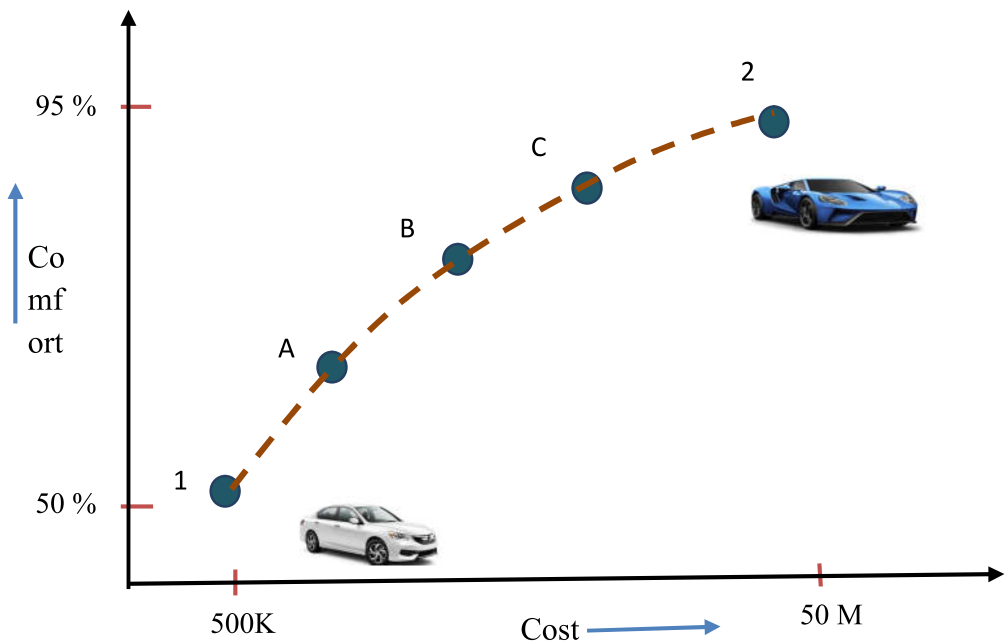


Fig. 1: Car-buying decision-making problem (Hypothetical Optimal solutions)

The No free launch [16] theorem that logically proves that none of the only algorithm exists equally efficient for all engineering problem. This is the main reason that it allows all researcher either to propose new algorithm or improve the existing ones. This paper proposed the multi-objective version of the well-known dragonfly algorithm (DA) [17]. In this paper non-sorted DA (NSDA) is tested on the standard un-constraint and constraint test function along with some well-known engineering design problem, their results are also compared with contemporary multi-objective algorithms Multi objective Colliding Bodies Optimizer (MOCBO)[18], Multi objective Particle Swarm Optimizer (MOPSO)[19-20], Non-dominated Sorting Genetic Algorithm (NSGA) [21-23], non-dominated sorting genetic algorithm II (NSGA-II)[24] and Multi objective Symbiotic Organism Search (MOSOS)[25]that are widely accepted due to their ability to solve real world problem.

The structure of the paper can be given as follows: - Section 2 consists of literature; Section 3 includes the proposed novel NSDA algorithm; Section 4 consists of competitive results analysis of standard test functions as well as engineering design problem and section 5 includes real world application, finally conclusion based on results and future scope of work is drawn.

II. Literature Review

As the name describes, multi-objective optimization handles simultaneously multiple objectives. Mathematically minimize/maximize optimization problem can be written as follows:

$$Minimize/maximize: \quad Fn(\vec{x}) = \{fn_1(\vec{x}), fn_2(\vec{x}), \ldots, f_{no}(\vec{x})\} \tag{2.1}$$

$$Subject\ to: \quad p_i(\vec{x}) \geq 0, \quad i = 1,2, \ldots, q \tag{2.2}$$

$$t_i(\vec{x}) = 0, \quad i = 1,2, \ldots, r \tag{2.3}$$

$$L_i^{lb} \leq x_i \leq U_i^{ub}, \quad i = 1,2,\ldots,k \tag{2.4}$$

Where q is the number of inequality constraints, *r* is the number of equality constraints, *k* is the number of variables, p_i is the i^{th} inequality constraints, *no* is the number of objective functions, t_i indicates the i^{th} equality constraints, and $[L_i^{lb}, U_i^{ub}]$ are the boundaries of i^{th} variable.

Obviously, relational operators are ineffective in comparing solutions with respect to multiple objectives.

The most common operator in the literate is Pareto optimal dominances, which is defined as follows for minimization problems:

$$\forall n \in \{1,2,\dots,k\}: f_n(\vec{x}) \le f_n(\vec{y}) \quad \wedge \quad \exists n \in \{1,2,\dots,k\}: f_n(\vec{x}) < f_n(\vec{y}) \tag{2.5}$$

where $\vec{x} = (x_1, x_2, \dots, x_k)$ and $\vec{y} = (y_1, y_2, \dots, y_k)$.

For maximization problems, Pareto optimal dominance is defined as follows:

$$\forall n \in \{1,2,\dots,k\}: f_n(\vec{x}) \ge f_n(\vec{y}) \quad \wedge \quad \exists n \in \{1,2,\dots,k\}: f_n(\vec{x}) > f_n(\vec{y}) \tag{2.6}$$

where $\vec{x} = (x_1, x_2, \dots, x_k)$ and $\vec{y} = (y_1, y_2, \dots, y_k)$.

These equations show that a solution is better than another in a multi-objective search space if it is equal in all objective and better in at least one of the objectives. Pareto optimal dominance is denoted with $\prec$ and $\succ$. With these two operator's solutions can be easily compared and differentiated.

Population based multi-objective algorithm's solution consists of multiple solution. But with multi-objective algorithm we cannot exactly determine the optimal solution because each solution is bounded by other objectives or we can say there is always conflict between other objectives. So the main function of stochastic/population based multi-objective algorithm is to find out best trade-offs between the objectives, so called Pareto optimally set [26-28].

The principle of working for an ideal multi-objective optimization algorithm is as shown in Fig. 2.

Step No. -1 Find maximum number of non-dominated solution according to objective, it expresses the number of Pareto optimal set so as shows higher coverage

Step No. -2 Choose one of the Pareto optimal solution using crowding distance mechanism that fulfills the objectives.

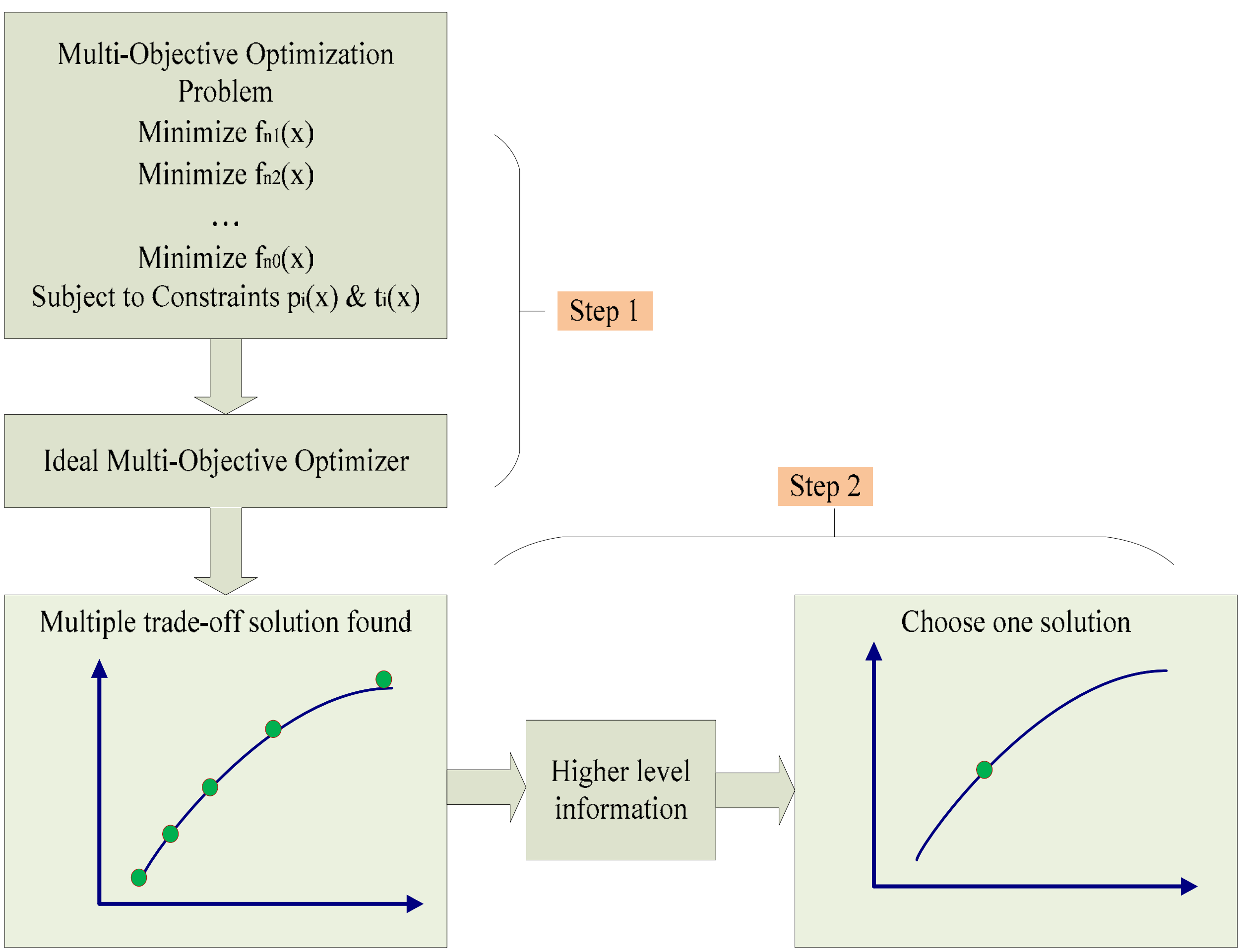


Fig. 2: Multi-objective optimization (Ideal) procedure.

Now a day recently proposed sole objective algorithms are equipped with powerful operators to provide them a capability to solve multi-objective problems as well. In the same manner we proposed NSDA algorithm in a hope that it will perform efficiently for multi-objective problems. These are: Multi-objective GWO [29], Multi-objective Bat Algorithm [30], Multi-objective Bee Algorithm [31],Pareto Archived Evolution Strategy (PAES) [32], Pareto-frontier Differential Evolution (PDE) [33], Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [34], Strength-Pareto Evolutionary Algorithm (SPEA) [35, 36] and Multi-objective water cycle algorithm with unconstraint and constraint standard test functions [37][38].Performance measurement for approximate robustness to Pareto front of multi-objective optimization algorithms in terms of coverage, convergence and success metrics.

The computational complexity of NSDA algorithm is order of $O(mn^2)$where N is the number of individuals in the population and M is the number of objectives. The complexity for other good algorithms in this field: NSGA-II, MOPSO, SPEA2 and PAES are $O(mn^2)$. However, the computational complexity is much better than some of the algorithms such as NSGA and SPEA which are of $O(mn^3)$.

III. Non-Dominated Sorting Dragonfly Algorithm (NSDA)

Dragonfly Algorithm (DA) with sole objective was proposed by Mirjalili Seyedali in 2015 [17]. It is basically a stochastic population based, nature inspired algorithm. In this algorithm the basic strategy based on swarming nature of dragonflies for exploration and exploitation. DA algorithm originated from the static and dynamic swarming behaviors of dragonflies. These two swarming behaviors are similar to the basic stage of working of any optimization algorithm in all meta-heuristic algorithms as: exploration and exploitation. Dragonflies build small number of group and fly in different directions in search of food is known as static swarm, this function is very similar to exploration phase in meta-heuristic techniques. Whereas, dragonflies make a big group and fly in only direction for either attacking to prey or migration to other place is known as dynamic swarm, this function is very similar to exploitation phase.

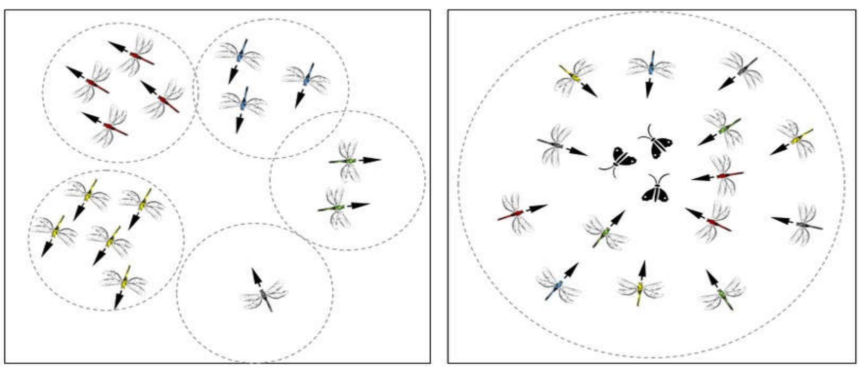


Fig. 3: Representation of static and dynamic swarming behavior of Dragonflies

Mathematical modelling of Dragonfly Algorithm:

Each portion of Dragonfly Algorithm is formulated by algebraic equations are:

1. For Separation part formulating equation:

$$\text{SEP.}_j = -\sum_{i=1}^{N} L - L_i \tag{3.1}$$

2. For Alignment part formulating equation:

$$\text{Alig.}_j = \frac{\sum_{i=1}^{N} L_i}{N} \tag{3.2}$$

3. For cohesion part formulating equation:

$$\text{Coh.}_j = \frac{\sum_{i=1}^{N} L_i}{N} - L \tag{3.3}$$

Global Journal of Researches in Engineering (F) Volume XX Issue II Version I Year 2020

4. For Attraction towards a food source part formulating equation:

$$\mathrm{F}_j = L^{+} - \mathrm{L} \tag{3.4}$$

5. For Attraction towards a food source part formulating equation:

$$\mathrm{E}_{\mathrm{j}} = L^{-} + \mathrm{L} \tag{3.5}$$

6. Step vector is formulating equation:

$$\Delta L_{\mathrm{t+1}} = \left(\mathrm{sSep.}_{\ \mathrm{j}} + \mathrm{aAlig.}_{\ j} + \mathrm{cCoh.}_{\ j} + \mathrm{fF}_j + \mathrm{eE}_j\right) + \mathrm{w}\Delta L_{\mathrm{t}} \tag{3.6}$$

7. Position vector is calculated using equation

$$L^{d}{}_{\mathrm{t+1}} = L_{\mathrm{t}} + \Delta L_{\mathrm{t+1}} \tag{3.7}$$

8. Position of dragonfly updated using equation

$$L_{\mathrm{t+1}} = L_{\mathrm{t}} + \mathrm{Levy}(L) * L_{\mathrm{t}} \tag{3.8}$$

Where:

L=Location of the current individuals, N= Neighboring individuals, L^{+}=positions of food source, L^{-}=positions of enemy, s=separation weight, a=alignment weight, c=cohesion weight, f=food weight, e=enemy weight, w=inertia weight, t=iteration counter and d=dimension of position vectors that levy flight step calculated.

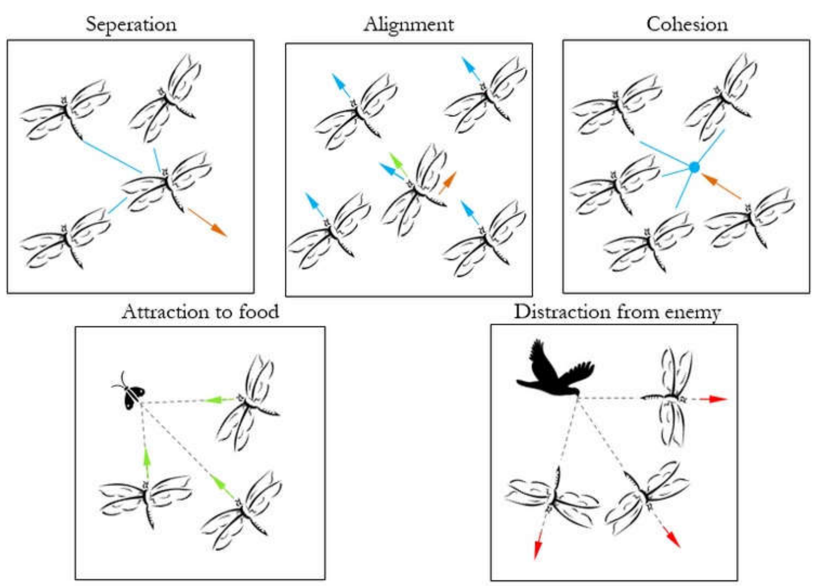


Fig. 4: Dragonfly Algorithm principle

Basic working of NSDA algorithm is as follows:

- Stage 1
  - First of all, initialize the population of dragonflies
  - Randomly generated sets of dragonflies & position vectors are represented in matrix for convenience to understand
  - Then fitness of step vector & position vectoris calculated on an according as objective function
- Stage 2
  - Position of dragonflies are updated as a function of levy flight motion and so as value of position vector is decided
  - The value of absolute distance is achieved which is basically a distance between the current best solution to the final optimal solution
  - Step vector is a function of both static and dynamic swarming behavior of dragonflies where some constant weight is assigned to the step vector function according to their swarming nature
- Step 3
  - Termination counter in integrated to limit/forcefully stop the search in uncertain search space (max. iteration counter to forcefully converge the search to optimal one)

- Size of the position vector matrix is continuously reduced over the course of iteration due to directed search to find global best solution
- Continuously position of the dragonflies are updated towards the optimal one via a Levy function or position vector equation for each iteration

➢ Step 4

- Likewise, multi-objective optimization the NSDA algorithm is made to capable to store the pare to optimal solutions in a collection set and make it as flexible to change solution over the course of iteration
- Solution is assigned a rank according to their ability as if a solution is not dominated by other solution is assigned rank1, dominated by only solution assigned rank 2 and so on & if collection set is full (archive size) over predefined size then some solutions that are less non-dominated (according to fitness value) in nature are directed to be out from the collection set according to the crowding distance mechanism.

This collection set is similar to the term achieve used in MOSOS and NSGA-II. It is a repository to store the best non-dominated solutions obtained so far. The search mechanism in NSDA is very similar to that of DA, in which solutions are improved using step vectors. Due to the existence of multiple best solutions, however, the best dragon flies position should be chosen from the collection set.

In order to select solutions from the archive to establish tunnels between solutions, we employ a leader selection mechanism. In this approach, the crowding distance between each solution in the archive is first selection and the number of solutions in the neighbourhood is counted as the measure of coverage or diversity. We require the NSDA to select solutions from the less populated regions of the archive using the following equation to improve the distribution of solutions in the archive across all objectives.

This subsection proposes multi-objective version of the DA algorithm called NSDA algorithm. The non-dominated sorting has been of the most popular and efficient techniques in the literature of multi-objective optimization. As its name implies, non-dominated sorting sort Pareto optimal solutions based on the domination level and give them a rank. This means that the solutions that are not dominated by any solutions is assigned with rank 1, the solutions that are dominated by only one solution are assigned rank 2, the solutions that are dominated by only two solutions are assigned rank 3, and so on. Afterwards, solutions are chosen to improve the quality of the population base on their rank. The better rank, the higher probability to be chosen. The main drawback of non-dominated sorting is its computational cost, which has been resolved in NSGA-II.

The success of the NSGA-II algorithm is an evidence of the merits of non-dominated sorting in the field of multi-objective optimization. This motivated our attempts to employ this outstanding operator to design another multi-objective version of the DA algorithm. In the NSDA algorithm, solutions are updated with the same equations presented in equation 3.9. In every iteration, however, the solutions to have optimal position of dragonflies are chosen using the following equation:

$$P_i = {}^{c}/_{Rank_i} \qquad (3.9)$$

where *c* is a constant and should be greater than 1 and $Rank_i$ is the rank number of solutions after doing the non-dominated sorting.

This mechanism allows better solutions to contribute in improving the solutions in the population. It should be noted that non-dominated sorting gives a probability to dominated solutions to be selected as well, which improves the exploration of the NSDA algorithm. Flow chart of NSDA algorithm is represented as Fig. 5.

Constraint Handling Approach:

With the extended literature survey we find that the population based algorithms are the common way to solve the multi-objective problems as they are more commonly provides the global solution and capable of handling both continuous and combinational optimization problem with a very high coverage and convergence. Multi-objective problems are subjected to various type of constraints like linear, non-linear, equality, inequality etc. So with these problems embedded it is very difficult to find simple and good strategy to achieve considerable solutions in the acceptable criterion. So in this paper NSDA algorithm uses a very simple approach to get feasible solutions. In this mechanism, after generating number of solutions at each generation, all the desirable constraint checked and then some solution that fulfills the criterion of acceptable solution are selected and collected them in achieve. Afterward non dominated solutions added in archive as we find more suitable solution to get acceptable solution. So as if achieve is full then less dominated solutions are removed. Finally according to crowing distance mechanism all these solutions (more suitable position of dragonflies) from archive is selected to get desired solution.

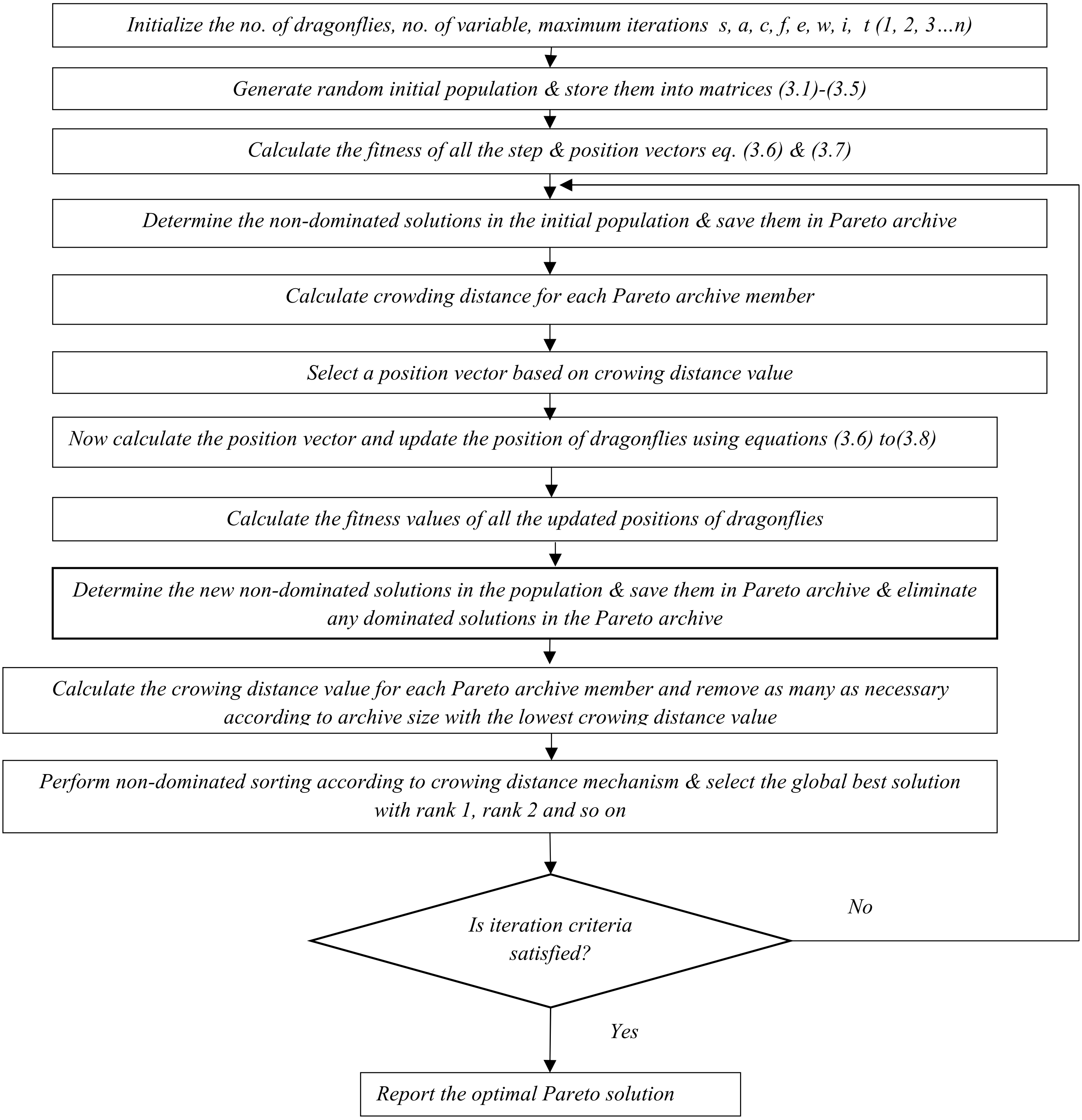


Fig. 5: Flow chart of NSDA algorithm

IV. Results Analysis on Test Functions

For determine the performance of proposed NSDA algorithm is applied to:

- A set of unconstraint and constraint standard multi-objective test functions
- Tested on well-known engineering design problems
- Non-linear, highly complex practical application known as formulation of economic constrained emission dispatch (ECED) with stochastic integration of wind power (WP) in the next section

NSDA algorithm is tested on seventeen different multi-objective case studies, including eight unconstrained test functions, five constrained test functions, and four real world engineering design problem, later algorithm is applied to the main application economic constrained emission dispatch with wind power (ECEDWP). These can be classified into four groups given below:

- Standard multi-objective unconstrained test functions (KUR, FON, ZDT1, ZDT2, ZDT3, ZDT4, SCHN1, and SCHN2)
- Standard multi-objective constrained test functions (TNK, OSY, BNH, SRN, and CONST)
- Real world engineering multi-objective design problem (Four bar truss design, welded beam design, speed reducer and disk brake design problem)
- Modeling of ECEDWP problem

Mathematical representation of these standard test functions are given in Appendix 1. (Multi-objective unconstrained test functions), 2. (Multi-objective constrained test functions), 3. (Engineering multi-objective design problem) with distinct characteristics like non-linear, non-convex, discrete pareto fronts and convex etc. are selected to measure the performance of proposed NSDA algorithm. To deal with real world engineering design problem is really a typical task with unknown search space, in this article we includes four different engineering problems are considered and performance is compared with various well known algorithms like MOWCA, NSGA-II, MOPSO, PAES and μ-GA multi-objective algorithms. Each algorithm is separately runs fifteen times and numeric results are listed in tables below. To measure the quality of obtained results we match their coverage of obtained true pareto front with respect to their original or true pareto fronts.

For numeric as well as qualitative performance of purposed NSDA algorithm on various case studies we consider Generational Distance (GD) given by Veldhuizen in 1998 [39]for measuring the deviation of the distance between true pareto front and obtained pareto front, Diversity matric (Δ) also known as matrix of spread to measure the uniformly distribution of non-dominated solution given by Deb [24]and Metric of spacing (S) to represent the distribution of non-dominated distribution of obtained solutions by purposed algorithm given by Schott [40].

The mathematical representation of these performance indicating metric are as follows:

$$GD = \frac{\sqrt{\int_{i=1}^{n_{PFs}} (d_i)^2}}{n} \tag{4.1}$$

where d_i shows the Euclidean distance (calculated in the objective space) between the i^{th} Pareto optimal solution achieved and the nearest true Pareto optimal solution in the reference set, n_{PFs} is the total number of achieved Pareto optimal solutions.

$$\Delta = \frac{\left|d_l+d_m+\int_{i=1}^{n_{PFs}}|d_i-d|\right|}{d_l+d_m+(n-1)d} \tag{4.2}$$

where, d_l , d_m are Euclidean distances between extreme solutions in true pareto front and obtained pareto front. d_i shows the Euclidean distance between each point in true pareto front and obtained pareto front. n_{PFs} and 'd' are the total number of achieved Pareto optimal solutions and averaged distance of all solutions.

$$S = \sqrt{\frac{1}{n_{PFs}-1}\int_{i=1}^{n_{PFs}}(d_i-d)^2} \tag{4.3}$$

where "d" is the average of all d_i , n_{PFs} is the total number of achieved Pareto optimal solutions, and $d_i = \min_j\left(|f_1^i(\vec{x}) - f_1^j(\vec{x})| + |f_2^i(\vec{x}) - f_2^j(\vec{x})\right)$ for all *i,j=1,2,…,n*. Smallest value of *"S"* metric gives the global best non-dominated solutions are uniformly distributed, thus if numeric value of d_i and d are same then value of *"S"* metric is equal to zero.

a) Results on unconstrained test problems

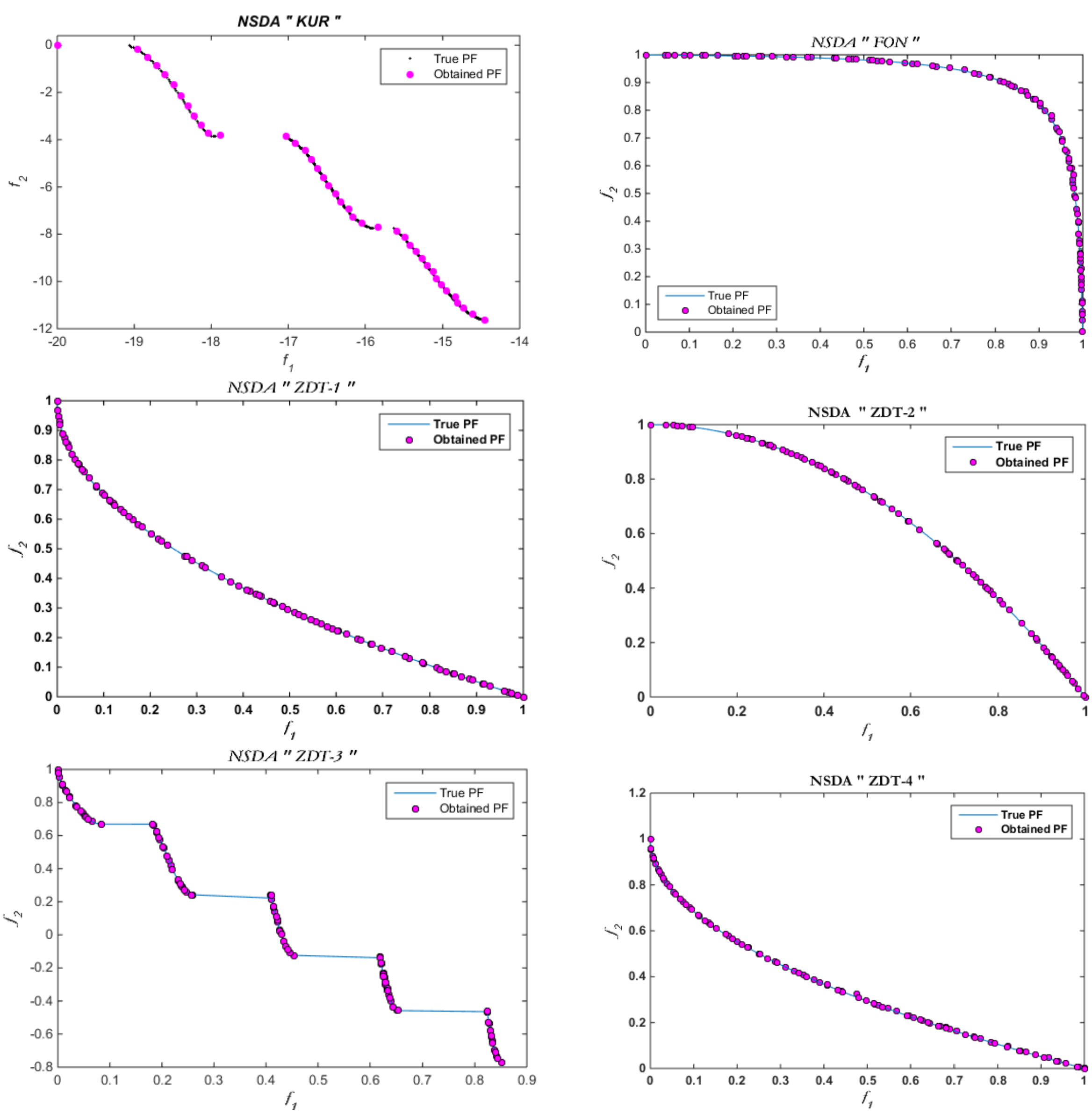
Like as above mentioned, the first set of test problems consist of unconstrained standard test functions. All the standard unconstrained test functions mathematical formulation is shown in Appendix A. Later, the numeric results are represented in Table 1 and best optimal pareto front is shown in Fig. 6.

All the statistical results are shown Table 1 suggests that the NSDA algorithm effectively outperforms with most of the unconstraint test functions compare to the MOSOS, MOCBO, MOPSO and NSGA-II algorithm. The effectiveness of proposed non-dominated version of DA (NSDA algorithm) can be seen in the Table 1, represents a greater robustness and accuracy of NSDA algorithm in terms of mean and standard deviation with the help of GD, diversity matrix along with computational time. However, proposed NSDA algorithm shows very competitive results in comparison with the MOPSO, MOCBO and MOSOS algorithms and in some cases these algorithms performs better than proposed one. Pareto front obtained by proposed NSDA algorithm shows almost complete coverage with respect to true pareto front.

Table 1: Results of the multi-objective NSDA algorithms (using GD, Δ, CT) on the unconstrained test functions employed

Algorithm→ Function ↓	PFs	NSDA MEAN±SD	MOSOS MEAN±SD	MOCBO MEAN±SD	MOPSO MEAN±SD	NSGA-II MEAN±SD
	GD	0.00729±0.00241	0.0075±0.0042	0.0083±0.0062	0.015±0.0075	0.0301±0.0043
KUR	Δ	0.02704±0.01025	0.0295±0.0122	0.0357±0.0236	0.0991±0.031	0.0362±0.0240
	CT	7.65853±0.44369	10.7413±0.822	7.9531±0.5823	8.0532±0.621	20.4368±3.102
	GD	0.00173±0.00032	0.0019±0.0002	0.0022±0.0003	0.0042±0.000	0.0026±0.0003

FON	Δ	0.29805±0.03758	0.3875±0.0062	0.3955±0.0068	0.4158±0.008	0.3987±0.0082
	CT	09.6681±0.55567	11.4013±1.140	8.6606±0.8862	8.732±0.9134	22.0323±4.522
	GD	0.32751±0.06748	0.3325±0.0256	0.3337±0.0319	0.3348±0.035	0.3352±0.038
ZDT-1	Δ	0.35589±0.00875	0.3803±0.0122	0.3825±0.0125	0.3876±0.024	0.3905±0.0220
	CT	6.59987±0.00381	8.2351±0.0204	3.1435±0.0193	3.7533±0.006	11.2681±0.364
	GD	0.07104±0.00066	0.0731±0.0010	0.0729±0.0005	0.0733±0.001	0.0725±0.0004
ZDT-2	Δ	0.04239±0.06687	0.4307±0.0007	0.4316±0.0007	0.4321±0.001	0.431±0.00075
	CT	4.66875±0.02005	8.2345±0.0457	3.1502±0.0130	3.6113±0.014	11.2811±0.024
	GD	0.07146±0.03847	0.1022±0.5187	0.0982±0.5007	0.1235±0.009	0.1147±0.0039
ZDT-3	Δ	0.69874±0.23568	0.6537±0.0052	0.65325±0.002	0.8234±0.108	0.7386±0.0474
	CT	8.78546±0.34789	13.4567±0.129	6.2846±0.1059	8.3764±0.231	14.3406±0.144
	GD	0.49878±0.00020	0.5015±0.0006	0.5078±0.0013	0.5146±0.001	0.5204±0.0019
ZDT-4	Δ	0.35879±0.01478	0.4585±0.0073	0.4795±0.0079	0.6543±0.024	0.7003±0.0089
	CT	7.87956±0.12275	13.9022±0.121	6.6922±0.1440	8.8203±0.218	14.8102±0.170
	GD	0.00904±0.00070	0.0028±0.0024	0.0031±0.0032	0.0032±0.003	0.0034±0.0042
SCHN-1	Δ	0.50078±0.01578	0.5295±0.1312	0.5302±0.1356	0.8582±0.164	0.5502±0.1360
	CT	11.7805±1.23254	8.2135±1.121	5.4845±1.1320	5.5721±1.133	17.9121±2.162
	GD	0.04478±0.00189	0.0705±0.0215	0.0932±0.0228	0.1497±0.022	0.3096±0.0217
SCHN-2	Δ	0.66587±0.02458	0.7821±0.0512	0.801±0.08326	0.8652±0.060	0.9562±0.0921
	CT	5.79974±0.14058	8.7015±0.4532	5.9751±0.2821	6.0272±0.582	18.421±2.1802



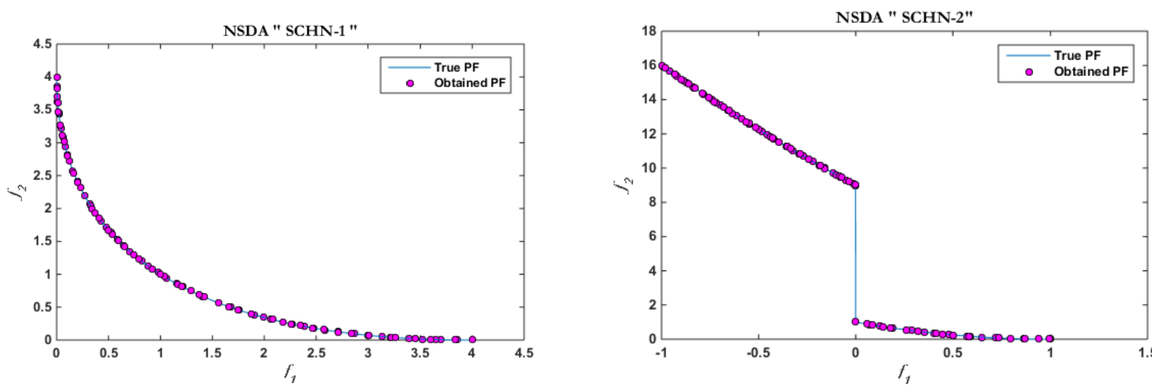


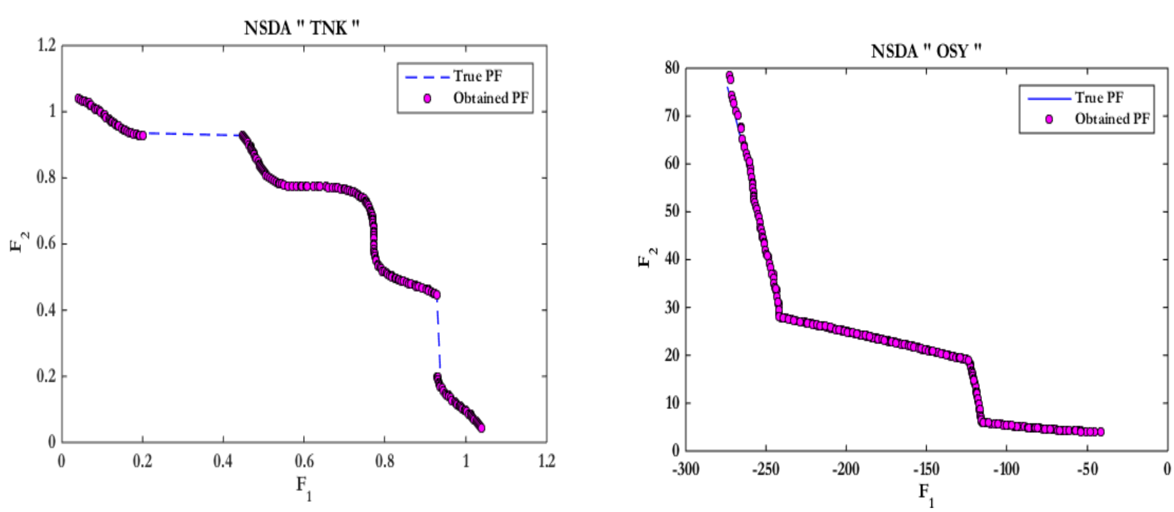
Fig. 6: Best Pareto optimal front of KUR, FON, ZDT1, ZDT2, ZDT3, ZDT4, SCHN1 and SCHN2 obtained by the NSDA algorithm

b) Results on constrained test problems

The next set of standard test functions consisting of constrained functions. For constrained test function it should be necessary that NSDA algorithm has a capability of handling constraints so algorithm is equipped with a death penalty function to search agents that violate any of the constraints at any level [41]. For comparing the results of different algorithms, we have utilized GD and Δ metrics.

Table 2: Results of the multi-objective NSDA algorithms on constrained test problems

Algorithm→ Function ↓	PFs	NSDA MEAN±SD	MOSOS MEAN±SD	MOCBO MEAN±SD	MOPSO MEAN±SD	NSGA-II MEAN±SD
	GD	0.14466±0.00210	0.1508±0.0040	0.1528±0.0051	0.1576±0.0062	0.1542±0.0072
TNK	Δ	0.57896±0.05587	0.1206±0.0423	0.1242±0.0512	0.1286±0.0522	0.126±0.06242
	CT	10.7895±0.04748	15.1286±0.063	11.0104±0.052	12.0212±0.054	17.4204±0.055
	GD	0.10054±0.00020	0.1196±0.0031	0.1210±0.0041	0.1282±0.0042	0.1242±0.0043
OSY	Δ	0.54789±0.05679	0.5354±0.0616	0.5422±0.0712	0.5931±0.0721	0.5682±0.0751
	CT	15.5578±0.02047	20.2124±0.032	12.2104±0.030	14.6420±0.042	24.2204±0.039
	GD	0.14458±0.00375	0.1436±0.0062	0.1498±0.0076	0.1644±0.0078	0.1566±0.0042
BNH	Δ	0.44587±0.03789	0.4288±0.0625	0.4798±0.0721	0.4975±0.0632	0.4892±0.0832
	CT	07.5254±0.04587	16.2664±0.054	9.1544±0.0420	9.7452±0.0464	19.652±0.0511
	GD	0.05001±0.01478	0.0988±0.0014	0.1018±0.0015	0.1125±0.0026	0.1024±0.0032
SRN	Δ	0.20458±0.00090	0.2295±0.0017	0.2352±0.0019	0.2730±0.0023	0.2468±0.0018
	CT	7.24456±0.00102	12.3254±0.012	7.3251±0.0082	9.2134±0.0083	17.0231±0.023
	GD	0.32145±0.04002	0.5162±0.0021	0.5202±0.0034	0.5854±0.0036	0.5532±0.0041
CONST	Δ	0.7056±0.000706	0.7122±0.0072	0.7235±0.0083	0.7344±0.0084	0.8126±0.0087
	CT	16.8556±0.00054	10.0112±0.003	5.2252±0.0028	6.4766±0.0035	14.0892±0.003



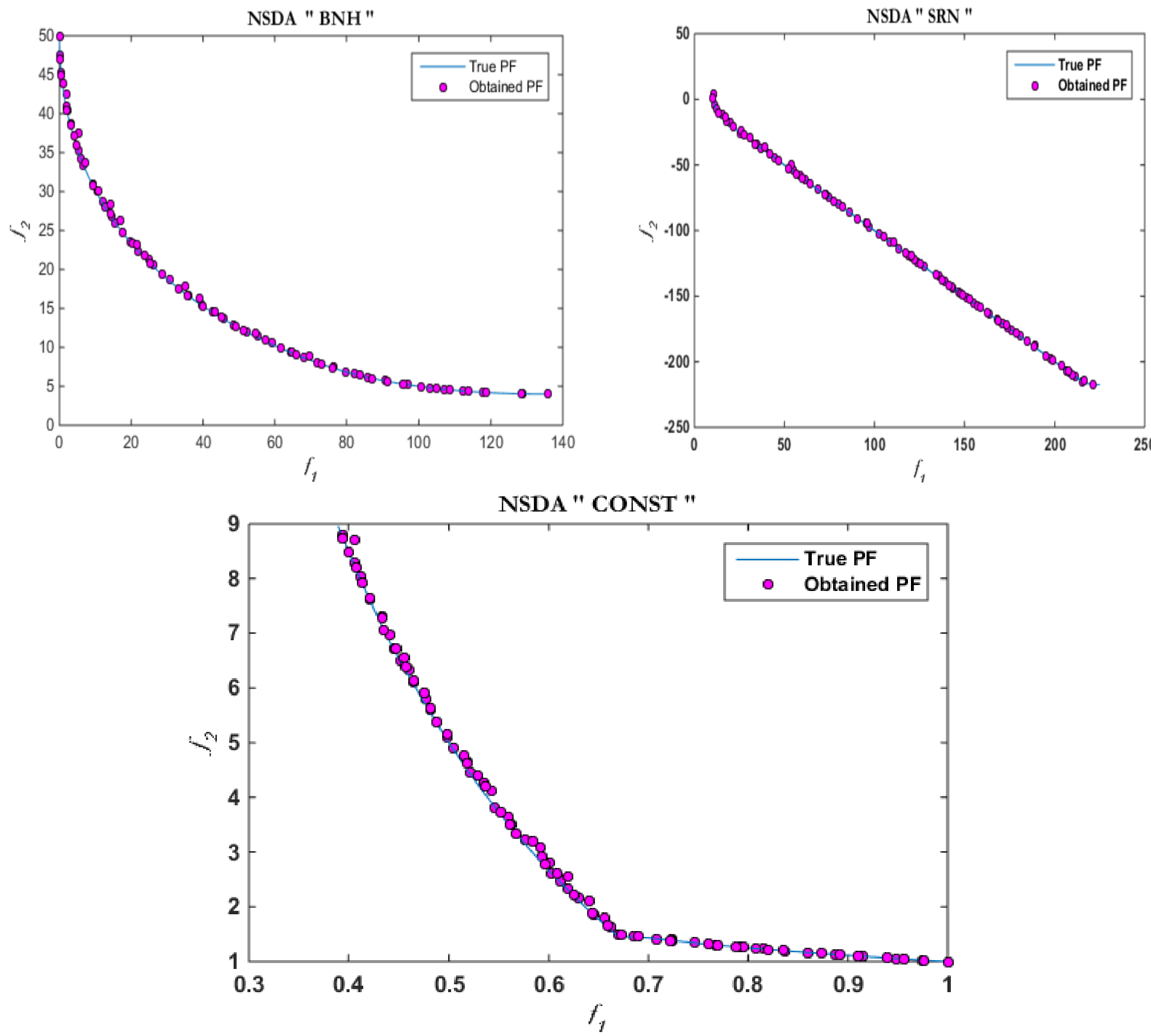


Fig. 7: Best Pareto optimal front TNK, OSY, BNH, SRN and CONST obtained by NSDA algorithm

Table 2 suggests that the NSDA algorithm comparatively performs better than other four algorithms for most of the standard constrained test functions employed. The best Pareto optimal fronts in Fig. 7 also helps in proving since all the Pareto optimal solutions exactly follow the true pareto fronts obtained from by NSDA algorithm.

CONST function consists of concave front with linear front, OSY is similar to CONST but consists of many linear regions with different slops while TNK almost similar to wave shaped. These also suggests that NSDA algorithm has a capability to solve various type of constraint problem. All the constraint test functions are mathematically given in Appendix B.

c) Results on constrained engineering design problems

The third set of test functions is the most complicated one and consists of four real engineering design problems. Mathematical model of all the four engineering design problem are given in Appendix C. Same as before both GD and diversity matrix is employed to measure the performance of NSDA algorithm with respect to other algorithms to solve them, numeric results are given in Tables and Figure respectively shows the best optimal front obtained by NSDA algorithm.

i. *Four-bar truss design problem*

The statistical results of four bar truss design problem [42] in given in Table 3 and best optimal front is given in Fig. 8. It consists of two minimization objectives displacement and volume with four design control variable mathematically given in Appendix C.

Table 3: Results of the multi-objective NSDA algorithm on four-bar truss design problem in terms mean and standard deviation

PFs→	GD	S
Methods ↓	MEAN±SD	MEAN±SD
NSDA	0.1756±0.0235	1.8717±0.1205
MOWCA	0.2076±0.0055	2.5816±0.0298
NSGA-II	0.3601±0.0470	2.3635±0.2551
MOPSO	0.3741±0.0422	2.5303±0.2275
μ- GA	0.9102±1.7053	8.2742±16.831
PAES	0.9733±1.8211	3.2314±5.9555

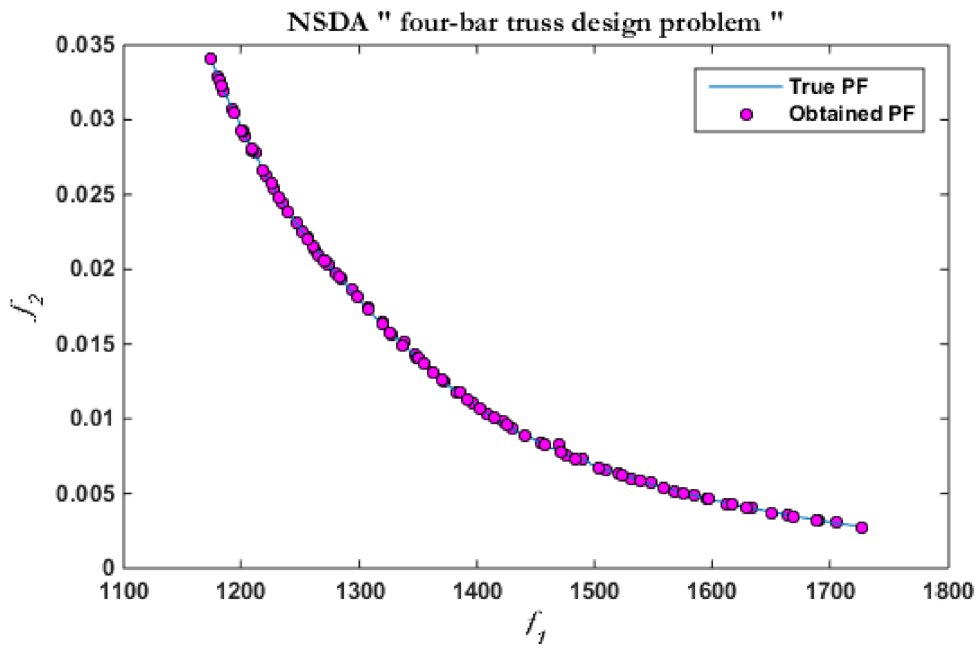


Fig. 8: Pareto optimal front obtained by the NSDA Algorithm for "Four –bus truss design problem"

ii. *Speed-reducer design problem*

The statistical results of speed reducer design problem[43] is given in Table 4 and best optimal front is given in Fig. 9. It is a well-known mechanical design problem consists of two minimization objectives stress and weight with seven design control variable mathematically given in Appendix C.

Table 4: Results of the multi-objective NSDA algorithm on speed-reducer design problem in terms mean and standard deviation

PFs→ Methods ↓	GD	S
	MEAN±SD	MEAN±SD
NSDA	0.95578±0.32458780	1.578354±05.947475
MOWCA	0.98831±0.17894217	16. 68520±2.6969443
NSGA-II	9.843702±7.0810303	02.7654494±3.534978
μ- GA	3.117536±1.6781086	47.80098±32.8015157
PAES	77.99834±4.2102608	16.20129±4.26842769

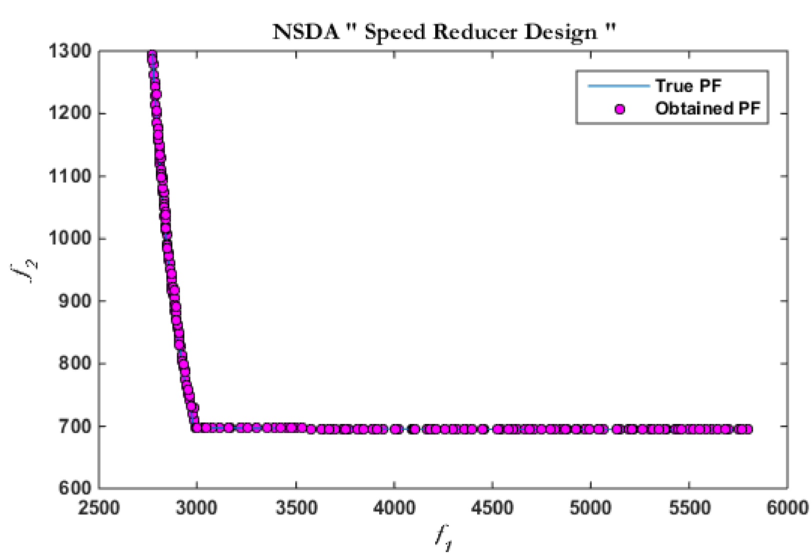


Fig. 9: Pareto optimal front obtained by the NSDA Algorithm for "Speed Reducer design problem"

iii. *Welded-beam design problem*

The statistical results of welded beam design problem [44] is given in Table 5 and best optimal front is given in Fig. 10. It is a well-known mechanical design problem consists of two minimization objectives fabrication cost and deflection of beam with four design control variable mathematically given in Appendix C.

Table 5: Results of the multi-objective NSDA algorithms on welded-beam design problem in terms mean and standard deviation

PFs→ Methods ↓	GD	Δ
	MEAN±SD	MEAN±SD
NSDA	0.03325±0.01693	0.75844±0.03770
MOWCA	0.04909±0.02821	0.22478±0.09280
NSGA-II	0.16875±0.08030	0.88987±0.11976
paε-ODEMO	0.09169±0.00733	0.58607±0.04366

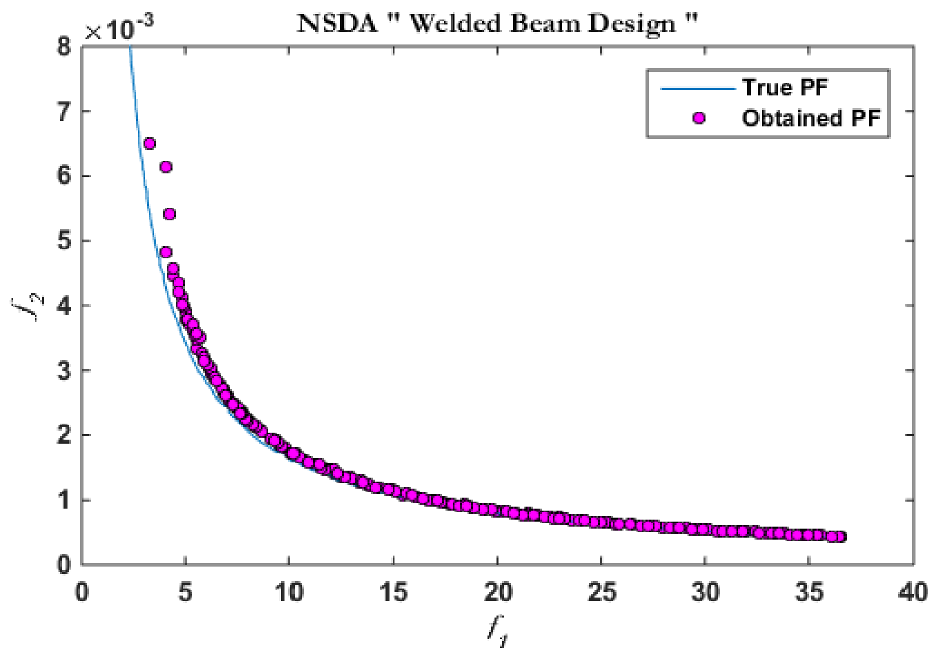


Fig. 10: Pareto optimal front obtained by the NSDA Algorithm for "Welded Beam Design problem"

iv. *Disk brake design problem*

The statistical results of welded beam design problem [44] is given in Table 6 and best optimal front is given in Fig. 11. It is a well-known mechanical design problem consists of two minimization objectives stopping time and mass of brake of a disk brake with four design control variable mathematically given in Appendix C.

Table 6: Results of the multi-objective NSDA algorithms on the Disk brake design problem in terms mean and standard deviation

PFs→ Methods ↓	GD	Δ
	MEAN±SD	MEAN±SD
NSDA	0.0587±0.27810	0.43551±0.08237
paε-ODEMO	2.6928±0.24051	0.84041±0.20085
NSGA-II	3.0771±0.10782	0.79717±0.06608
MOWCA	0.0244±0.12314	0.46041±0.10961

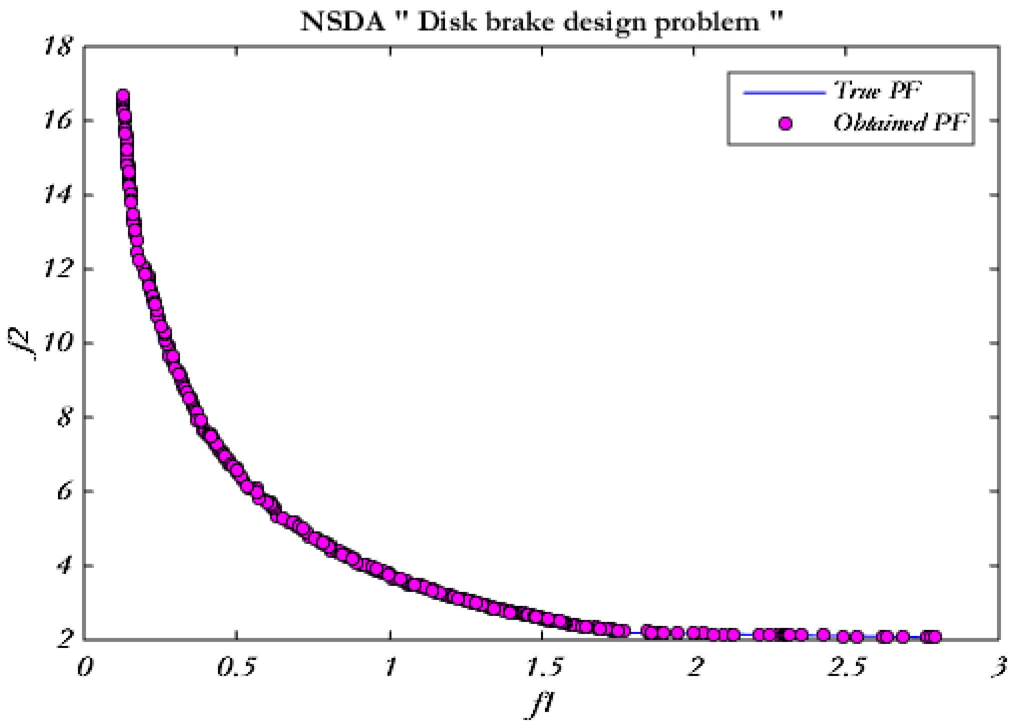


Fig. 11: Pareto optimal front obtained by the NSDA Algorithm for "Disk brake design problem"

Due to high complexity of engineering design problem it is really hard to gain results alike true pareto front but we can clearly see that optimal pareto obtained by NSDA algorithm is covers almost whole solutions that are the actual/true solutions of an engineering design problem. From all above tested function we can conclude that problem either it consists of constraints or unconstraint problem NSDA algorithm shows its capability to solve any kind of linear, non-linear and complex real world problem. So in the next section we attached a highly non-linear complex real problem to show its effectiveness regarding the real world complex application with many objectives.

d) Formulation of Economic Constrained Emission Dispatch (ECED) with integration of Wind Power (WP)

i. *Mathematical formulation of wind power*

In case of wind power generation the output power of wind generator is calculated with the help of a stochastic variable wind speed υ (meter/seconds). Wind speed is a variable function so there probability distribution plays a very important role. Wind speed mathematically formulated as two-parametric Weibull distribution function, probability density function (PDF) and cumulative distribution function (CDF) as follows:

$$s(v) = (k/c)\,(v/c)^{k-1} * \exp\left[-(v/c)^{k}\right],\ v \geq 0 \tag{4.1}$$

$$S(v) = 1 - \exp\left[-(v/c)^{k}\right], v \geq 0 \tag{4.2}$$

Where, S(v) and s(v) are CDF and PDF respectively. Shape factor and scale factor are k and c respectively.

The wind speed and output wind power are related as:

$$P_{wind} = \begin{cases} 0, & v < v_{in}\ or\ v \geq v_{out} \\ P^{rated}\frac{v-v_{in}}{v_{rated}-v_{in}} & v_{in} \leq v < v_{rated} \\ P_{rated} v_{in} & \leq v < v_{out} \end{cases} \tag{4.3}$$

Where, v_{rated} and P_{rated} are the rated speed of wind and rated power output. v_{out} and v_{in} are cut-out and cut-in speed of wind respectively. The CDF of P_{wind} in the boundary of [0, P_{rated}] on an accordance with the speed range of wind can be formulated as:

$$S(P_{wind}) = 1 - exp\begin{cases} -\left[\left(1 + \frac{v_{rated}-v_{in}}{v_{in}*P_{rated}} P_{wind}\right)\frac{v_{in}}{c}\right]^{k} \} + \exp\left[-(v_{out}/c)^{k}\right], \\ 0 \leq P_{wind} < P_{rated} \end{cases} \tag{4.4}$$

Above equation is very meaningful to calculate the ECED problems with speculative wind power with variable speed.

Global Journal of Researches in Engineering (F) Volume XX Issue II Version I Year 2020

ii. *Modeling of ECEDWP problem*

As wind power is formulated as system constraint, so the objective function of economic emission dispatch problem (EEDP) stays on unchanged as classical EEDP:

Fuel cost objective is given by:

$$Minimization \quad S(P_i) = \sum_i^N (a_i + b_i P_i + c_i P_i^2) \tag{4.5}$$

where, the thermal power generators cost coefficients are a_i, b_i, c_i for i-th generator, Sum of the total fuel cost of the system and N is the total number of generators.

Total Emission is calculated by:

$$Minimization \;\; E(P_i) = \sum_i^N [\{(\alpha_i + \beta_i P_i + \gamma_i P_i^2) * 10^{-2}\} + \delta_i * \exp(\varphi_i * P_i)] \tag{4.6}$$

where, $\alpha_i, \beta_i, \gamma_i, \delta_i$ and φ_i are emission coefficients with valve point effect taking into consideration for i-th thermal generator.

iii. *System Constraints*

As wind power generation is considered as system constraint with the summation of stochastic variables the classical power balance constraint changes to fulfill the predefined confidence level.

$$P_r \sum_{i=1}^N (P_i + P_{Wind} \geq P_D + P_{Loss}) \geq \eta_{pbc} \tag{4.7}$$

where, η_{pbc} is confidence level that a power system must follow the load demand and so as it is selected nearer to unity as values lesser than unity represents high operational risk. P_{loss}represents system losses can be calculated by B-coefficient method given below:

$$P_{Loss} = \sum_{i=1}^N \sum_{j=1}^N P_i B_{ij} P_j + \sum_{i=1}^N P_i B_{i0} + B_{00} \tag{4.8}$$

So as to change above described power balance constrained equation into deterministic form can be solved as:

$$P_r\{P_{Wind} < P_D + P_{Loss} - \sum_{i=1}^N P_i\} = F(P_D + P_{Loss} - \sum_{i=1}^N P_i) \leq 1 - \eta_{pbc} \tag{4.9}$$

Assume that the wind turbine have same speed and same direction and combination of Eqs. (4) and (9), the power balance constraint is represented as:

$$P_D + P_{Loss} - \sum_{i=1}^N P_i \leq \frac{cP_{rated}}{v_{rated} - v_{in}} \left| \ln\left[\eta_{pbc} + exp * \left(\frac{v_{out}^k}{c^k}\right)\right]\right|^{\frac{1}{k}} - \frac{v_{in} * P_{rated}}{v_{rated} - v_{in}} \tag{4.10}$$

iv. *Reserve capacity system constraint*

So as to reduce the impact of stochastic wind power on system, up and down spinning reserve needs to be maintained [22]. Such reserve constraints formulated as [15] and [16] respectively:

$$P_r\{\sum_{i=1}^N (P_i^{max} - P_i) \geq P_{sr} + t_u * P_{Wind}\} \geq \eta_{urc} \tag{4.11}$$

$$P_r\{\sum_{i=1}^N (P_i - P_i^{min}) \geq t_d * (P_{rated} - P_{Wind})\} \geq \eta_{drc} \tag{4.12}$$

where, P_{sr} represents the reserve demand of conventional thermal power plant system and it generally keeps the maximum value of thermal unit, P_i^{max} and P_i^{min} are maximum and minimum output level of operational generators of i-th unit, η_{drc} and η_{urc} are predefined down and upper confidence level parameter respectively, t_u and t_d are the demand coefficients of up and down spinning reserves.

v. *Generational capacity constraint*

The real output power is bounded by each generators upper and lower bounds given as:

$$P_i^{Minimum} \leq P_i \leq P_i^{Maximum} \tag{4.13}$$

V. 40-Operational Thermal Generating Unit

a) Case study I- 40 thermal-generator lossless system without wind power

In this case forty operational generating unit is consider without integration of wind power means all the generating units are coal fired. Input parameters like generators operating limit, fuel cost coefficients and emission coefficients are given in Appendix D extracted from [45]. System is considered lossless and its solution is compared with three well known multi-objective algorithms like SMODE [45], NSGA-II [45]and MBFA [46] in terms of various objectives such as best cost, best emission and best compromise between both objectives. Best compromise solution is then obtained by the fuzzy based method [47]. Total power demand for this system is 10500 MW. Results obtained by NSDA algorithm is added to table 7 and best pareto front obtained by NSDA algorithm is represented in Fig. 12.

Table 7: Results of the multi-objective NSDA algorithms for case study I- 40 thermal-generator lossless system without wind power

Case Study I	SMODE [45]			NSGAII [45]			MBFA [46]			NSDA		
	Best emission	Best cost	Best compromise	Best emission	Best Cost	Best compromise	Best emission	Best cost	Best compromise	Best emission	Best cost	Best compromise
Cost ($/h)	156,700	119,650	124,230	128,490	124,380	126,180	129,995	121,415	123,638	127,568	119,310	124,830
Emission (tons/h)	66,799	377,560	96,578	93,002	153,560	99,671	176,682	356,424	188,963	87,124	408,025	94,450

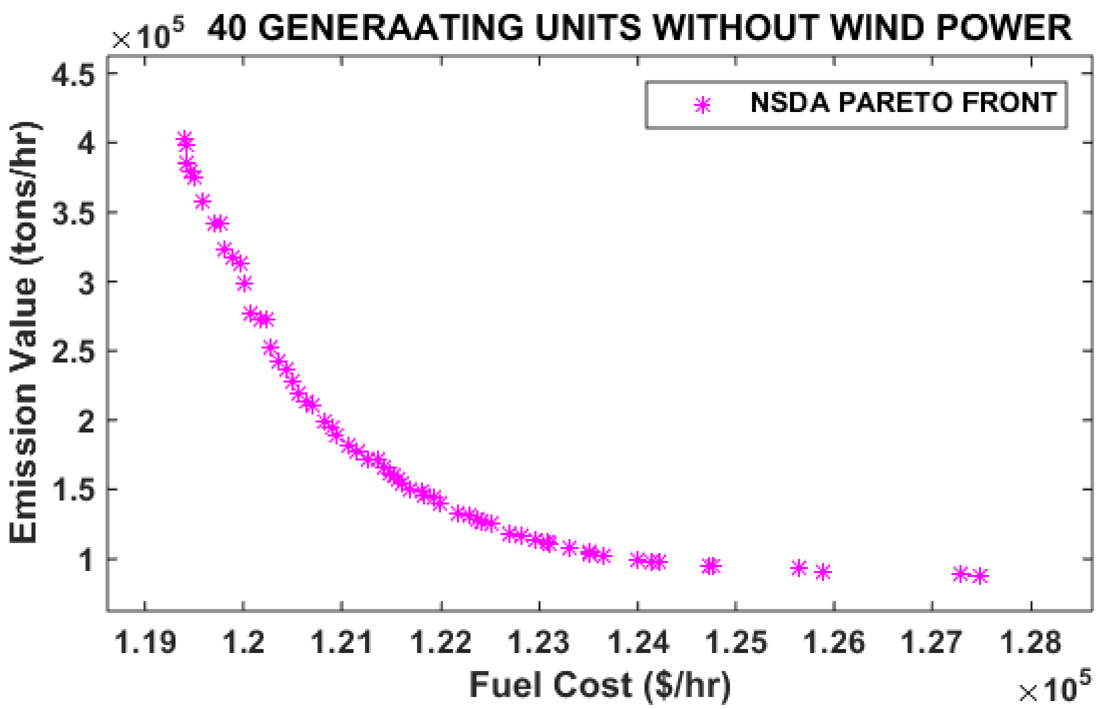


Fig. 12: Pareto optimal front obtained by the NSDA Algorithm for "40 thermal-generator lossless system without wind power"

b) Case study II- 40 thermal-generator lossless system with wind power

All the conditions are remaining same as case study I like input parameters and power demand. While integrating with wind power plant, the total rated output power of wind farm is set to 1000 MW [45, 47].Statistical results obtained by NSDA algorithm is reported in Table 8 and best optimal front is represented in Fig. 13.

Table 8: Results of the multi-objective NSDA algorithms for case study II- 40 thermal-generator lossless system with wind power

Case Study-II	SMODE[45]			NSGAII [45]			MOEA/D[51]			NSDA		
	Best emission	Best cost	Best Compromise point	Best emission	Best cost	Best Compromise Point	Best emission	Best cost	Best Compromise	Best emission	Best cost	Best Compromise Point
$\sum P_G$ P_W Cost Emission	10,245.76 254.24 153,830 54,055	10,177.55 322.45 116,430 385,770	10,225.71 274.29 123,590 68,855	10,241.72 258.28 132,410 73,894	10,242.09 257.91 122,610 121,850	10,241.63 258.37 126,240 78,860	10,244.43 255.568 154,0 0 0 55,754	10,242.71 257.294 115,770 440,240	10,242.8 257.156 120,950 79,485	10,242.7 257.321 146,685 56,509	10,224.18 275.82 118,689 179,099	10,236.58 263.42 123,459 68,801

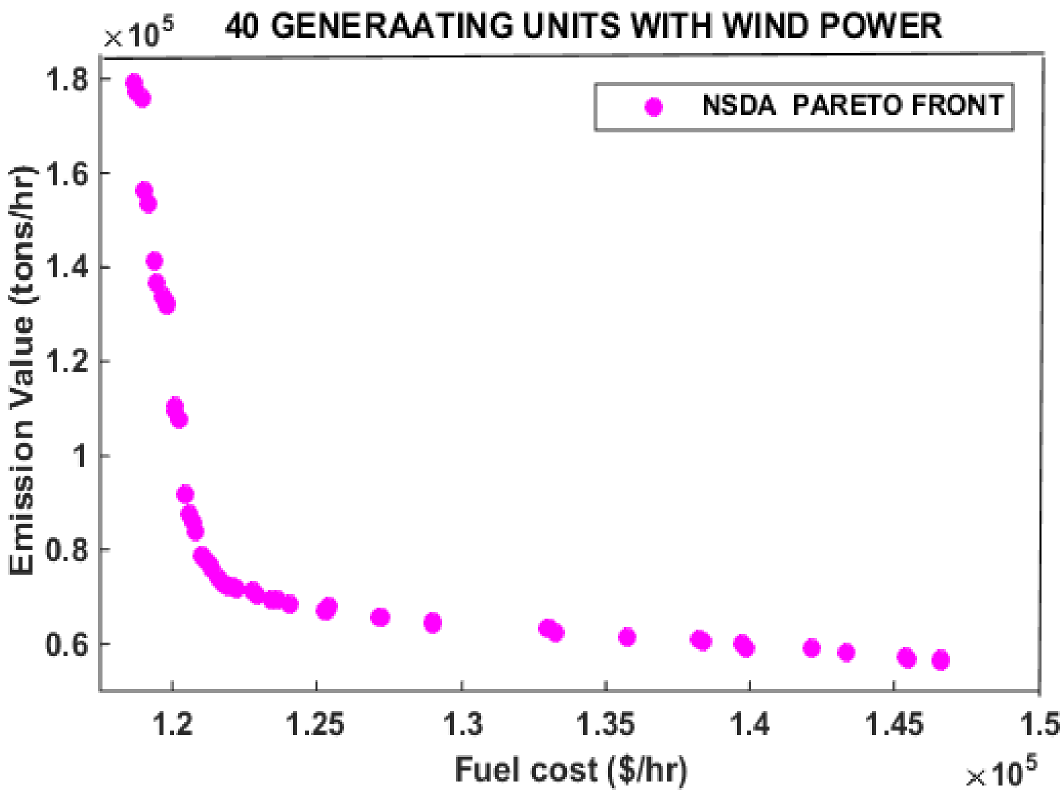


Fig. 13: Pareto optimal front obtained by the NSDA Algorithm for "40 thermal-generator lossless system with wind power"

Table 9: Results of Wilcoxon test and simulation/computational time or speed

		NSDA	NSGAII [45]			NSDA	NSGAII [45]
Case Study I Cost	Best Worst Mean Wilcoxon test (H/P) Simulation speed (s)	119310 127568 124830 1/ 5.40e −10 11.89	124,380 147,760 131,710	Case Study II Cost	Best Worst Mean Wilcoxon test (H/P) Simulation speed (s)	118,689 146,685 123,010 1/ 5.77e −10 09.785	122,610 173,060 134,880
Case Study I Emission	Best Worst Mean Wilcoxon test (H/P) Simulation speed (s)	87,124 408.025 189,284 1/ 5.55e −10 20.57	93,002 194,830 141,800 154.78	Case Study II Emission	Best Worst Mean Wilcoxon test (H/P) Simulationspeed (s)	56,509 179,099 104,258 1/ 5.63e −10 40.04	73,894 158,250 102,120 127.57

VI. Result Discussion

In almost all the cases that we consider in this article where NSDA algorithm proves its effectiveness in both prospective quantitative and qualitative. From plots also evident that NSDA algorithm follows the exact pareto front similar to the true pareto front for all constrained, unconstrained and complex engineering design problem. So as for real world application of economic emission dispatch problem and its integration with stochastic wind power generation. So for this application Wilcoxon test (statistical test) is performed. In Table 9 the signed rank test is presented in third row of each results whereas the calculation time is represented in forth row. For this test null hypothesis cannot be rejected at 5% level for numeric value '0' while null hypothesis is rejected at 5% level with the value of '1'. Where NSDA algorithm performs superior to other algorithms that are considered for comparative purpose. NSDA algorithm shows good performance in both coverage and convergence as main mechanism that guarantee convergence in DA and NSDA algorithms are continuously shrink its virtual limitation using Levy strategy in the movement of dragonflies for their random walk. Both mechanism emphasizes convergence and exploitation proportional to maximum number of generation (iteration). Since this complex task might degrade its performance compare to without limitation or free movement should be a concern. However the numerical results expresses that NSDA algorithm has a little effect of slow convergence at all.

NSDA algorithm has an advantage of high coverage, which is the result of the selection of position of dragonflies and archive selection procedure. All the position are updated according to their fitness value that enable the algorithm to direct the search space in right direction to find the best solution without trapped in local solution. Archive selection criteria follow all the rules of the entrance and exhaust of any value in it for each iteration and updated when its size full. Solutions of higher fitness in archive have higher probability to thrown away first to improve the coverage of the pareto optimal front obtained during the optimization process.

VII. Conclusion

In this paper the non-dominated sorting dragonfly algorithm-multi-objective version of recently proposed dragonfly algorithm (DA) is proposed known as NSDA algorithm. This paper also utilizes the static and dynamic swarming strategy for exploration purpose used in its parent DA version. NSDA algorithm is developed with equipping dragonfly algorithm with crowding distance criterion, an archive and dragonflies position (accordance to ranking) selection method based on Pareto optimal dominance nature. The NSDA algorithm is first applied on 17 standard test functions (including eight unconstraint, five constraint and four engineering design problem) to prove its capability in terms of qualities and quantities showing numerical as well as convergence and coverage of pareto optimal front with respect to true pareto front. Then after NSDA algorithm is applied to real world complex ECEDWP problem where algorithm proves its dominance over other well recognized contemporary algorithms. The numeric results are stored and represented in performance indices: GD, metric of diversity, metric of spacing and computational time. The qualitative results are reported as convergence and coverage in best pareto optimal front found in 15 independent runs. To check effectiveness of proposed version of algorithm the results are verified with SMODE, MOSOS, MOCBO, MOPSO, NSGA-II and other well recognize algorithms in the field of multi-objective algorithms. We can also conclude from the standard test functions results that NSDA algorithm is able to find pareto optimal front of any kind of shape. Finally, the result of complex real world ECEDWP problem validates that NSDA algorithm is capable of solving any kind of non-linear and complex problem with many constraint and unknown search space. Therefore, we conclude that proposed non-dominated version of DA algorithm has various merits among the contemporary multi-objective algorithms as well as provides an alternative for solving multi or many objective problems.

For future works, it is suggested to test NSDA algorithm on other real world complex problems. Also, it is worth to investigate and find the best constrained handling technique for this algorithm.

References Références Referencias

1. C. T. Kelley, "Detection and Remediation of Stagnation in the Nelder--Mead Algorithm Using a Sufficient Decrease Condition," *SIAM Journal on Optimization,* vol. 10, pp. 43-55, 1999.
2. T. P. Vogl, J. Mangis, A. Rigler, W. Zink, and D. Alkon, "Accelerating the convergence of the back-propagation method," *Biological cybernetics,* vol. 59, pp. 257-263, 1988.
3. Seyedali Mirjalili,"Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," Knowledge-Based System, vol. 89, pages 228-249, 2015.
4. Xin-She Yang (2010) The bat algorithm (BA),"A Bio-inspired algorithm".
5. J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 1995, pp. 1942–1948.
6. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1:28–39
7. John H (1992) Holland, adaptation in natural and artificial systems. MIT Press, Cambridge
8. Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29:17–35
9. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput 13:2592–2612
10. A.H. Gandomi, A.H. Alavi, Krill Herd: a new bio-inspired optimization algorithm, Common Nonlinear Sci. Numer. Simul. 17 (12) (2012) 4831–4845.
11. Gandomi A.H. "Interior Search Algorithm (ISA): A Novel Approach for Global Optimization." ISA Transactions, Elsevier, 53(4), 1168–1183, 2014.
12. H.-G. Beyer and B. Sendhoff, "Robust optimization–a comprehensive survey," Computer methods in applied mechanics and engineering,vol. 196, pp. 3190-3218, 2007.
13. J. D. Knowles, R. A. Watson, and D. W. Corne, "Reducing local optima in single-objective problems by multi-objectivization," in *Evolutionary multi-criterion optimization*, 2001, pp. 269-283.
14. K. Deb and D. E. Goldberg, "Analyzing deception in trap functions," *Foundations of genetic algorithms,* vol. 2, pp. 93-108, 1993.

15. C. A. C. Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer methods in applied mechanics and engineering,* vol. 191, pp. 1245-1287, 2002.
16. D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *Evolutionary Computation, IEEE Transactions on,* vol. 1, pp. 67-82, 1997.
17. Seyedali Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems", "The Natural Computing Applications Forum 2015", 30 April 2015.
18. A. Panda, S. Pani, Multiobjective colliding bodies optimization, in:Proceedings of 5th Int. Conf. on Soft Computing for Problem Solving, SocProS,IIT Roorkee, India, 2015.
19. C.A. Coello Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives withparticle swarm optimization, IEEE Trans. Evol. Comput. 8 (3) (2004) 256–279.
20. C.A. Coello Coello, M.S. Lechuga, MOPSO: a proposal for multiple objectiveparticle swarm optimization, in: Proceedings of the IEEE Congress onEvolutionary Computation, 2002, pp. 1051–1056.
21. K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in Parallel problem solving from nature PPSN VI, 2000, pp. 849-858.
22. K. Deb and T. Goel, "Controlled elitist non-dominated sorting genetic algorithms for better convergence," in Evolutionary multi-criterion optimization, 2001, pp. 67-81.
23. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," Evolutionary Computation, IEEE Transactions on, vol. 6, pp. 182-197, 2002.
24. K. Deb, A. Pratap, S. Agarwal, T.A.M.T. Meyarivan, A fast and elitistmultiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2)(2002) 182–197.
25. Arnapurna Panda, Sabyasachi Pani: A Symbiotic Organisms Search algorithm with adaptive penaltyfunction to solve multi-objective constrained optimization problems, Applied Soft Computing 46 (2016) 344–360.
26. P. Ngatchou, A. Zarei, and M. El-Sharkawi, "Pareto multi objective optimization," in Intelligent Systems Application to Power Systems, 2005. Proceedings of the 13th International Conference on, 2005, pp. 84-91.
27. V. Pareto, Cours d'economie politique: Librairie Droz, 1964.
28. F. Y. Edgeworth, Mathematical Physics: P. Keagan, London, England, 1881.
29. S. Mirjalili, S. Saremi, S. M. Mirjalili, and L. d. S. Coelho, "Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization," *Expert Systems with Applications,* vol. 47, pp. 106-119, 2016.
30. X.-S. Yang, "Bat algorithm for multi-objective optimisation," *International Journal of Bio-Inspired Computation,* vol. 3, pp. 267-274, 2011.
31. R. Akbari, R. Hedayatzadeh, K. Ziarati, and B. Hassanizadeh, "A multi-objective artificial bee colony algorithm," *Swarm and Evolutionary Computation,* vol. 2, pp. 39-52, 2012.
32. Knowles JD, Corne DW(2000) Approximating the nondominated front using the Pareto archived evolution strategy. Evol Comput 8(2):149–172
33. H. A. Abbass, R. Sarker, and C. Newton, "PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems," in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, 2001, pp. 971-978.
34. Qingfu Zhang,*Hui Li* :MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition, IEEE transactions on evolutionary computation, vol. 11, no. 6, december 2007.
35. E. Zitzler, *Evolutionary algorithms for multiobjective optimization: Methods and applications* vol. 63: Citeseer, 1999.
36. E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach," *Evolutionary Computation, IEEE Transactions on,* vol. 3, pp. 257-271, 1999.
37. Ali Sadollah, Hadi Eskandar, Ardeshir Bahreininejad, Joong Hoon Kim : Water cycle algorithm for solving multi-objective optimization problems, Soft Comput, 06 september 2014.
38. Ali Sadollaha, Hadi Eskandarb, Joong Hoon Kim : Water cycle algorithm for solving constrained multi-objectiveoptimization problems, Applied Soft Computing 27 (2015) 279–298.
39. D. A. Van Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithm research: A history and analysis," Citeseer1998.
40. J. R. Schott, "Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization," DTIC Document1995.
41. C. A. C. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Computers in Industry,* vol. 41, pp. 113-127, 2000.
42. C. C. Coello and G. T. Pulido, "Multiobjective structural optimization using a microgenetic algorithm," *Structural and Multidisciplinary Optimization,* vol. 30, pp. 388-403, 2005.
43. A. Kurpati, S. Azarm, and J. Wu, "Constraint handling improvements for multiobjective genetic algorithms," *Structural and Multidisciplinary Optimization,* vol. 23, pp. 204-213, 2002.

44. T. Ray and K. M. Liew, "A swarm metaphor for multiobjective design optimization," *Engineering optimization,* vol. 34, pp. 141-153, 2002.
45. B.Y. Qu , J.J. Liang, Y.S. Zhu, Z.Y. Wang, P.N. Suganthan, "Economic emission dispatch problems with stochastic wind power using summation based multi-objective evolutionary algorithm," Information Sciences 351 (2016) 48–66.
46. P.K. Hota , A.K. Barisal , R. Chakrabarti , Economic emission load dispatch through fuzzy based bacterial foraging algorithm, Electr. Power Energy Syst. 32 (2010) 794–803.
47. M.A. Abido , Environmental / economic power dispatch using multiobjective evolutionary algorithms: a comparative study, IEEE Trans Power Syst. 1 (4) (2003) 1529–1537.
48. T. T. Binh and U. Korn, "MOBES: A multiobjective evolution strategy for constrained optimization problems," in *The Third International Conference on Genetic Algorithms (Mendel 97)*, 1997, p. 27.
49. A. Osyczka and S. Kundu, "A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm," *Structural optimization,* vol. 10, pp. 94-99, 1995.
50. N. Srinivasan and K. Deb, "Multi-objective function optimisation using non-dominated sorting genetic algorithm," *Evolutionary Comp,* vol. 2, pp. 221-248, 1994.
51. Y.S. Zhu , J. Wang , Multi-objective economic emission dispatch considering wind power using evolutionary algorithm based on decomposition, Electr. Power Energy Syst. 63 (2014) 434–445.

Appendix A: Unconstrained multi-objective test problems utilized in this work

KUR:

Minimize:

$$f_1(x) = \sum_{i=1}^{2}\left[-10\exp\left(-0.2\sqrt{x_i^2 + x_{i+1}^2}\right)\right]$$

$$f_2(x) = \sum_{i=1}^{2} \left[|x_i|^{0.8} + 5sin(x_i^3)\right]$$

$$-5 \le x_i \le 5$$

$$1 \le i \le 3$$

FON:

$$\min imize = \begin{cases} f_1(x) = 1 - \exp\left[-\sum_{i=1}^{n}(x_i - \frac{1}{\sqrt{n}})^2\right] \\ f_2(x) = 1 - \exp\left[-\sum_{i=1}^{n}(x_i + \frac{1}{\sqrt{n}})^2\right] \end{cases}$$

$$-4 \le x_i \le 4$$

$$1 \le i \le n$$

ZDT1:

Minimise: $f_1(x) = x_1$

Minimise: $f_2(x) = g(x) \times h(f_1(x), g(x))$

Where:

$$G(x) = 1 + \frac{9}{N-1}\sum_{i=2}^{N} x_i\, h(f_1(x), g(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}}$$

$$0 \le x_i \le 1, 1 \le i \le 30$$

ZDT2:

Minimise: $f_1(x) = x_1$

Minimise: $f_2(x) = g(x) \times h(f_1(x), g(x))$

Where: $G(x) = 1 + \frac{9}{N-1} \sum_{i=2}^{N} x_i \; h(f_1(x), g(x)) = 1 - \left(\frac{f_1(x)}{g(x)}\right)^2$

$$0 \leq x_i \leq 1, 1 \leq i \leq 30$$

ZDT3:

Minimise: $f_1(x) = x_1$

Minimise: $f_2(x) = g(x) \times h(f_1(x), g(x))$

Where: $G(x) = 1 + \frac{9}{29} \sum_{i=2}^{N} x_i \; h(f_1(x), g(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}} - \left(\frac{f_1(x)}{g(x)}\right) \sin(10\pi f_1(x))$

$$0 \leq x_i \leq 1, 1 \leq i \leq 30$$

ZDT4:

Minimise: $f_1(x) = x_1$

Minimise: $f_2(x) = g(x) \times h(f_1(x), g(x))$

$$h(f_1(x), g(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}} \quad \mathrm{g(x)} = 91 + \sum_{i=2}^{10} (x_i^2 - 10 * \cos(4\pi x_i))$$

SCHN-1:

Minimize: $f_1(x) = x_i^2$

$f_2(x) = (x-2)^2$ Where: value of can be from 10 to 10^5.

$$-A \leq x \leq A$$

SCHN-2 :

Minimize:

$$\begin{cases} f_1(x) = \begin{cases} -\mathrm{x}, & if \quad x \leq 1 \\ \mathrm{x}-2, & if \quad 1 < x \leq 3 \\ 4-\mathrm{x}, & if \quad 3 < x \leq 4 \\ \mathrm{x}-4, & if \quad x > 4 \end{cases} \\ f_2(x) = (x-5)^2 \end{cases}$$

$$-5 \leq x \leq 10$$

Appendix B: Constrained multi-objective test problems utilised in this work

TNK:

Minimise: $f_1(x) = x_1$

Minimise: $f_2(x) = x_2$

Where: $g_1(x) = -{x_1}^2 - {x_2}^2 + 1 + 0.1Cos(16arctan\left(\frac{x_1}{x_2}\right))$

$$g_2(x) = 0.5 - (x_1 - 0.5)^2 - (x_2 - 0.5)^2 \quad 0.1 \leq x_1 \leq \pi, 0 \leq x_2 \leq \pi$$

BNH:
This problem was first proposed by Binh and Korn [48]:

Minimise: $f_1(x) = 4{x_1}^2 + 4{x_2}^2$

Minimise: $f_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2$

Where: $g_1(x) = (x_1 - 5)^2 + {x_2}^2 - 25$

$g_2(x) = 7.7 - (x_1 - 8)^2 - (x_2 + 3)^2$

$0 \le x_1 \le 5, 0 \le x_2 \le 3$

OSY:
The OSY test problem has five separated regions proposed by Osyczka and Kundu [49]. Also, there are six constraints and six design variables.

Minimise: $f_1(x) = {x_1}^2 + {x_2}^2 + {x_3}^2 + {x_4}^2 + {x_5}^2 + {x_6}^2$

Minimise: $f_2(x) = -[25(x_1 - 2)^2 + (x_2 - 1)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2]$

Where: $g_1(x) = 2 - x_1 - x_2$

$g_2(x) = -6 + x_1 + x_2$

$g_3(x) = -2 - x_1 + x_2$

$g_4(x) = -2 + x_1 - 3x_2$

$g_5(x) = -4 + x_4 + (x_3 - 3)^2$

$g_6(x) = 4 - x_6 - (x_5 - 3)^2$

$0 \le x_1 \le 10, 0 \le x_2 \le 10, 1 \le x_3 \le 5, 0 \le x_4 \le 6, 1 \le x_5 \le 5, 0 \le x_6 \le 10$

SRN:
The third problem has a continuous Pareto optimal front proposed by Srinivas and Deb [50].

Minimise: $f_1(x) = 2 + (x_1 - 2)^2 + (x_2 - 1)^2$

Minimise: $f_2(x) = 9x_1 - (x_2 - 1)^2$

Where: $g_1(x) = {x_1}^2 + {x_2}^2 - 255$

$g_2(x) = x_1 - 3x_2 + 10$

$-20 \le x_1 \le 20, -20 \le x_2 \le 20$

CONSTR:

This problem has a convex Pareto front, and there are two constraints and two design variables.

Minimise: $f_1(x) = x_1$

Minimise: $f_2(x) = (1 + x_2)/(x_1)$

Where: $g_1(x) = 6 - (x_2 + 9x_1)$, $g_2(x) = 1 + x_2 - 9x_1$

$0.1 \le x_1 \le 1, 0 \le x_2 \le 5$

Appendix C: Constrained multi-objective engineering problems used in this work

Four-bar truss design problem:

The 4-bar truss design problem is a well-known problem in the structural optimisation field [42], in which structural volume (*f1*) and displacement (*f2*) of a 4-bar truss should be minimized. As can be seen in the following equations, there are four design variables (*x1-x4*) related to cross sectional area of members 1, 2, 3, and 4.

Minimise: $f_1(x) = 200 * (2 * x(1) + sqrt(2 * x(2)) + sqrt(x(3)) + x(4))$

Minimise: $f_2(x) = 0.01 * (\left(\frac{2}{x(1)}\right) + \left(\frac{2 * sqrt(2)}{x(2)}\right) - ((2 * sqrt(2))/x(3)) + (2/x(1)))$

$$1 \le x_1 \le 3, 1.4142 \le x_2 \le 3, 1.4142 \le x_3 \le 3, 1 \le x_4 \le 3$$

Speed reducer design problem:

The speed reducer design problem is a well-known problem in the area of mechanical engineering [43], in which the weight (*f1*) and stress (*f2*) of a speed reducer should be minimized. There are seven design variables: gear face width (*x1*), teeth module (*x2*), number of teeth of pinion (*x3* integer variable), distance between bearings 1 (*x4*), distance between bearings 2 (*x5*), diameter of shaft 1 (*x6*), and diameter of shaft 2 (*x7*) as well as eleven constraints.

Minimise: $f_1(x) = 0.7854 * x(1) * x(2)^2 * (3.3333 * x(3)^2 + 14.9334 * x(3) - 43.0934) - 1.508 * x(1) *$
$(x(6)\hat{}2 + x(7)\hat{}2) + 7.4777 * (x(6)\hat{}3 + x(7)\hat{}3) + 0.7854 * (x(4) * x(6)\hat{}2 + x(5) * x(7)\hat{}2)$

Minimise: $f_2(x) = ((sqrt(((745 * x(4))/(x(2) * x(3)))\hat{}2 + 16.9e6))/(0.1 * \ldots x(6)\hat{}3))$

Where: $g_1(x) = 27/(x(1) * x(2)\hat{}2 * x(3)) - 1$

$g_2(x) = 397.5/(x(1) * x(2)\hat{}2 * x(3)\hat{}2) - 1$

$g_3(x) = (1.93 * x(4)\hat{}3)/(x(2) * x(3) * x(6)\hat{}4) - 1$

$g_4(x) = (1.93 * x(5)\hat{}3)/(x(2) * x(3) * x(7)\hat{}4) - 1$

$g_5(x) = ((sqrt(((745 * x(4))/(x(2) * x(3)))\hat{}2 + 16.9e6))/(110 * x(6)\hat{}3)) - 1$

$g_6(x) = ((sqrt(((745 * x(5))/(x(2) * x(3)))\hat{}2 + 157.5e6))/(85 * x(7)\hat{}3)) - 1$

$g_7(x) = ((x(2) * x(3))/40) - 1$

$g_8(x) = (5 * x(2)/x(1)) - 1$

$g_9(x) = (x(1)/12 * x(2)) - 1$

$g_{10}(x) = ((1.5 * x(6) + 1.9)/x(4)) - 1$

$g_{11}(x) = ((1.1 * x(7) + 1.9)/x(5)) - 1$

$2.6 \le x_1 \le 3.6, 0.7 \le x_2 \le 0.8, 17 \le x_3 \le 28, 7.3 \le x_4 \le 8.3, 7.3 \le x_5 \le 8.3, 2.9 \le x_6 \le 3.9$

$5 \le x_7 \le 5.5$

Welded beam design problem:

The welded beam design problem has four constraints first proposed by Ray and Liew [44]. The fabrication cost (*f1*) and deflection of the beam (*f2*) of a welded beam should be minimized in this problem. There are four design variables: the thickness of the weld (*x1*), the length of the clamped bar (*x2*), the height of the bar (*x3*) and the thickness of the bar (*x4*).

Minimise: $f_1(x) = 1.10471 * x(1)\hat{}2 * x(2) + 0.04811 * x(3) * x(4) * (14.0 + x(2))$

Minimise: $f_2(x) = 65856000/(30 * 10\hat{}6 * x(4) * x(3)\hat{}3)$

Where: $g_1(x) = tau - 13600$

$g_2(x) = sigma - 30000$

$g_3(x) = x(1) - x(4)$

$$g_4(x) = 6000 - P$$

$$0.125 \leq x_1 \leq 5, 0.1 \leq x_2 \leq 10, 0.1 \leq x_3 \leq 10, 0.125 \leq x_4 \leq 5$$

Where

$$Q = 6000 * \left(14 + \frac{x(2)}{2}\right); D = sqrt\left(\frac{x(2)^2}{4} + \frac{\left(x(1) + x(3)\right)^2}{4}\right)$$

$$J = 2 * \left(x(1) * x(2) * sqrt(2) * \left(\frac{x(2)^2}{12} + \frac{\left(x(1) + x(3)\right)^2}{4}\right)\right)$$

$$alpha = \frac{6000}{sqrt(2) * x(1) * x(2)}$$

$$beta = Q * \frac{D}{J}$$

$$tau = sqrt\left(alpha^2 + 2 * alpha * beta * \frac{x(2)}{2 * D} + beta^2\right)$$

$$sigma = \frac{504000}{x(4) * x(3)^2}$$

$$tmpf = 4.013 * \frac{30 * 10^6}{196}$$

$$P = tmpf * sqrt\left(x(3)^2 * \frac{x(4)^6}{36}\right) * \left(1 - x(3) * \frac{sqrt\left(\frac{30}{48}\right)}{28}\right)$$

Disk Brake Design Problem:

The disk brake design problem has mixed constraints and was proposed by Ray and Liew [44]. The objectives to be minimized are: stopping time (*f1*) and mass of a brake (*f2*) of a disk brake. As can be seen in following equations, there are four design variables: the inner radius of the disk (*x1*), the outer radius of the disk (*x2*), the engaging force (*x3*), and the number of friction surfaces (x4) as well as five constraints.

Minimise: $f_1(x) = 4.9 * (10\text{^}(-5)) * (x(2)\text{^}2 - x(1)\text{^}2) * (x(4) - 1)$

Minimise: $f_2(x) = (9.82 * (10\text{^}(6)) * (x(2)\text{^}2 - x(1)\text{^}2))/((x(2)^3 - x(1)\text{^}3) * \ldots x(4) * x(3))$

Where: $g_1(x) = 20 + x(1) - x(2)$

$$g_2(x) = 2.5 * (x(4) + 1) - 30$$

$$g_3(x) = (x(3))/(3.14 * (x(2)\text{^}2 - x(1)\text{^}2)\text{^}2) - 0.4$$

$$g_4(x) = (2.22 * 10\text{^}(-3) * x(3) * (x(2)\text{^}3 - x(1)\text{^}3))/((x(2)\text{^}2 - x(1)\text{^}2)\text{^}2) - 1$$

$$g_5(x) = 900 - (2.66 * 10\text{^}(-2) * x(3) * x(4) * (x(2)\text{^}3 - x(1)\text{^}3))/((x(2)\text{^}2 - x(1)\text{^}2))$$

$$55 \leq x_1 \leq 80, 75 \leq x_2 \leq 110, 1000 \leq x_3 \leq 3000, 2 \leq x_4 \leq 20$$