Engineering technology open access Journal
ISSN: 2641-8185

Juniper Publishers
key to the Researchers

Research Article
Volume 2 Issue 1 - July 2018
DOI: 10.19080/ETOAJ.2018.02.555579

Eng Technol Open Acc

Non-Dominated Sorting Moth Flame Optimizer: A Novel Multi-Objective Optimization Algorithm for Solving Engineering Design Problems

Pradeep Jangir[1]*and Indrajit N Trivedi[2]

[1]*Department of Electrical Engineering, Rajasthan Rajya Vidyut Prasaran Nigam Losal-Sikar, India*

[2]*Department of Electrical Engineering, Government Engineering College, India*

Submission: April07, 2018; **Published:** July 05, 2018

*Corresponding author: Pradeep Jangir,Junior Engineer, Rajasthan Rajya Vidyut Prasaran Nigam Losal-Sikar, Rajasthan, India, Email: pkjmtech@gmail.com

Abstract

This novel article presents the multi-objective version of the recently proposed Moth-flame optimizer (MFO) known as Non-Dominated Sorting Moth Flame Optimizer (NSMFO). This proposed NSMFO algorithm works in such a manner that it first collects all non-dominated Pareto optimal solutions achieved 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 as navigation strategy to guide moths towards the dominated regions of multi-objective search spaces. For validate the efficiency and effectiveness of proposed NSMFO algorithm is applied to a set of standard unconstrained, constrained and engineering design problems. The results are verified by comparing NSMFO 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 NSMFO 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; NSMFO algorithm; Multi-objective algorithm; Economic constrained emission dispatch

Abbreviations: NSMFO: Non-Dominated Sorting Moth Flame Optimizer; MOCBO: Multi Objective Colliding Bodies Optimizer; MOPSO: Multi Objective Particle Swarm Optimizer; MOSOS: Multi Objective Symbiotic Organism Search; ET: Execution Time; GD: Generalized Distance; DM: Diversity Metric

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 Grey wolf optimizer (GWO) [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, better constraints handling etc [14,15]. 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 Figure 1, we consider two objectives, first cost and second comfort level. So we go for sole objective of minimum cost possible then we 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.

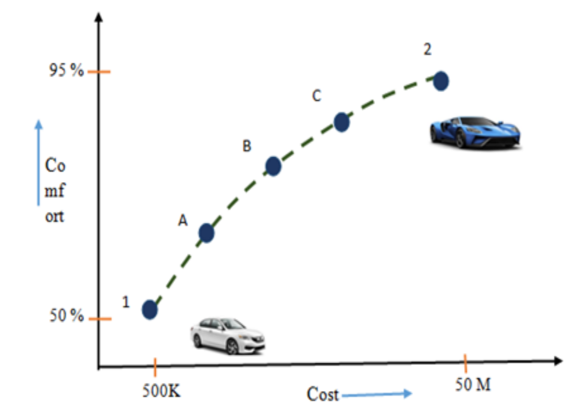

Figure 1: Car-buying decision-making problem (Hypothetical Optimal solutions).

Figure1 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 moth flame optimizer (MFO) [17]. In this paper non-sorted MFO (NSMFO) 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 NSMFO 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.

Literature Review

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

$$Minimize/maximize: \quad Fn(\vec{x})=\{fn_1(\vec{x}), fn_2(\vec{x}),\ldots, f_{no}(\vec{x})\} \tag{1}$$

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

$$t_i(\vec{x})=0, \qquad i=1,2,\ldots,r \tag{3}$$

$$L_i^{lb}\leq x_i \leq U_i^{ub} \quad , i=1,2,\ldots,k \tag{4}$$

Where q is the number of inequality constraints, r is the number of equality constraints is the number of variables, p_i is the ith inequality constraints, no is the number of objective functions, t_i indicates the ith equality constraints, and $[L_i^l b, U_i^u b]$ are the boundaries of ith 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:

$$\forall n\in\{1,2,\ldots,k\}: f_n(\vec{x})\leq f_n(\vec{y}) \;\wedge\; \exists n\in\{1,2,\ldots,k\}: f_n(\vec{x})< f_n(\vec{y}) \tag{5}$$

$$Where\,\vec{x}=(x_1,x_2,\ldots,x_k) and \vec{y}=(y_1,y_2,\ldots,y_k) \tag{6}$$

$$\forall n\in\{1,2,\ldots,k\}: f_n(\vec{x})\geq f_n(\vec{y}) \;\wedge\; \exists n\in\{1,2,\ldots,k\}: f_n(\vec{x})> f_n(\vec{y}) \tag{7}$$

$$\text{Where}\,\vec{x}=(x_1,x_2,\ldots,x_k)\,\text{and}\,\vec{y}=(y_1,y_2,\ldots,y_k) \tag{8}$$

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

Population based multi-objective algorithm's solution consists of multiple solutions. 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].

How to cite this article: Pradeep J, Indrajit N T. Non-Dominated Sorting Moth Flame Optimizer: A Novel Multi-Objective Optimization Algorithm for Solving Engineering Design Problems. Eng Technol Open Acc. 2018; 2(1): 555579. DOI: 10.19080/ETOAJ.2018.02.555579

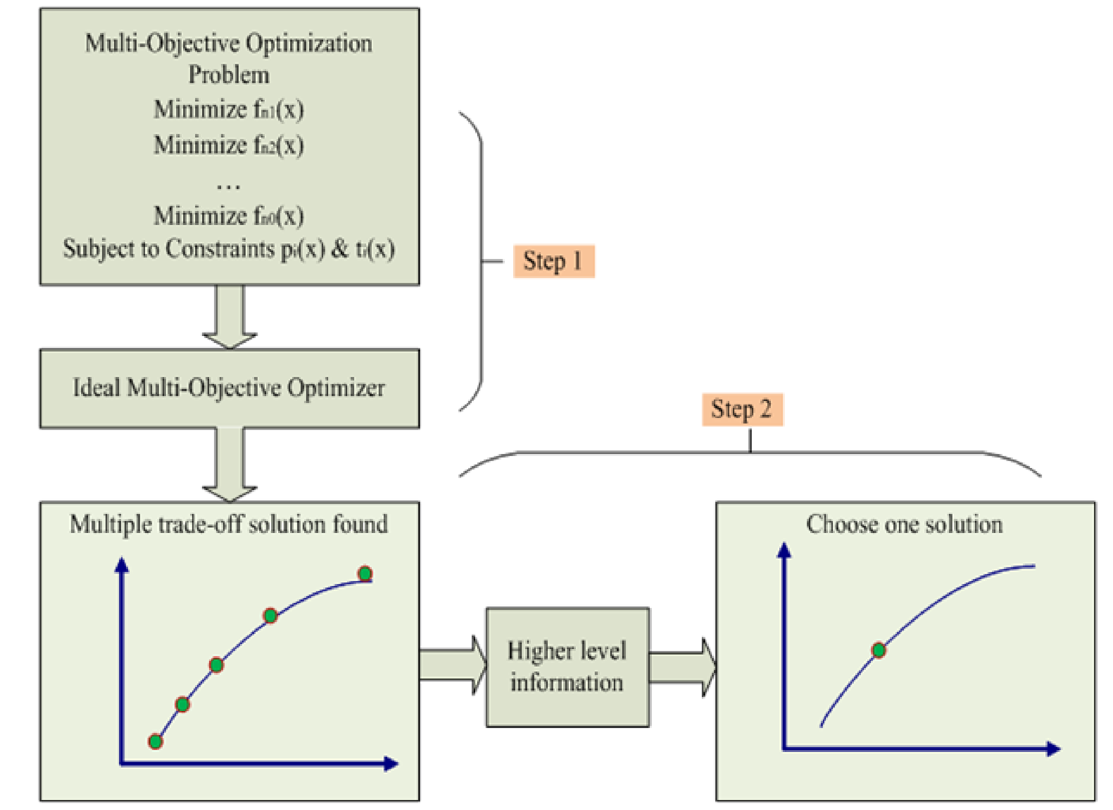
Multi-Objective Optimization Problem
Minimize fn1(x)
Minimize fn2(x)
...
Minimize fn0(x)
Subject to Constraints pi(x) & ti(x)
Step 1
Ideal Multi-Objective Optimizer
Step 2
Multiple trade-off solution found
Higher level information
Choose one solution

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

The principle of working for an ideal multi-objective optimization algorithm is as shown in Figure2.

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 fulfils the objectives.

Now a days, 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 NSMFO 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 NSMFO 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 $O(mn^3)$.

Non-Dominated Sorting Moth Flame Optimizer (NSMFO)

Moth-flame optimizer (MFO) algorithm 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 of navigation for travelling long distances in straight direction having a constant angle w.r.t moon (flame) is utilized by insects (Moths) in night. However, this strategy efficiently works only for the targeted flame is far distance away from the moths but fails if flame is nearer. Generally, if flame is far from moth then it is easy for the moths to keep the constant angle from it, but in opposite case when flame is nearer then the angle continuously decreasing and finally moths tends to dip into flame (targeted optimal solution).

Basic working of NSMFO algorithm is as follows

Stage 1:

a. First of all, initialize the population of moths

b. Randomly generated sets of moths & flames are represented in matrix for convenience to understand

c. Then fitness of moths & flames is calculated on an according as objective function

Stage 2:

a. Position of moths in a spiral path is a function of the position of moths and flames

b. The value of absolute distance is achieved which is basically a distance between the current best solution to the final optimal solution

c. Then exact position of moths & flames is achieved which is a mathematical spiral function of with a product of actual distance between them and an addition of the position of flame.

$$MFO = (J, Q, G)\,,\ I : \varnothing \rightarrow \{T, OT\} \tag{9}$$

$$Q : T \rightarrow T\,,\ G : T \rightarrow \{true,\ false\} \tag{10}$$

$$T_i = S(T_i, F_j)\ , \tag{11}$$

$$S(T_i, F_j) = D_i \cdot e^{bt} \cdot cos(2\pi t) + F_j \tag{12}$$

Where, J is a function that generates a random population of moths and their fitness values. Q is a function, moves the moths around unknown search space and G comes true when termination criterion fulfilled else false comes.

Step 3

a. 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)

b. Size of the flame matrix is continuously reduced over the course of iteration due to directed search to find global best solution (Quick sorting is used to provide directional search)

c. Continuously position of the moths and flames is updated via a spiral function equation for each iteration

Step 4

a. Likewise, multi-objective optimization the NSMFO algorithm is made to capable to store the pareto optimal solutions in a collection set and make it as flexible to change solution over the course of iteration

b. 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 over predefined size then some solutions that are dominated 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 NSMFO is very similar to that is MFO, in which solutions are improved using moths. Due to the existence of multiple best solutions, however, the best moth 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 NSMFO 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 MFO algorithm called NSMFO. 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 MFO algorithm. In the NSMFO algorithm, solutions are updated with the same equations presented in equation (12) In every iteration, however, the solutions to have optimal position of moths and flames are chosen using the following equation:

$$P_i = {}^{c}\!/_{Rank_i} \tag{13}$$

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 NSMFO algorithm.

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 types 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 NSMFO 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 fulfils 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 moths and flames) from archive is selected to get desired solution (figure 3).

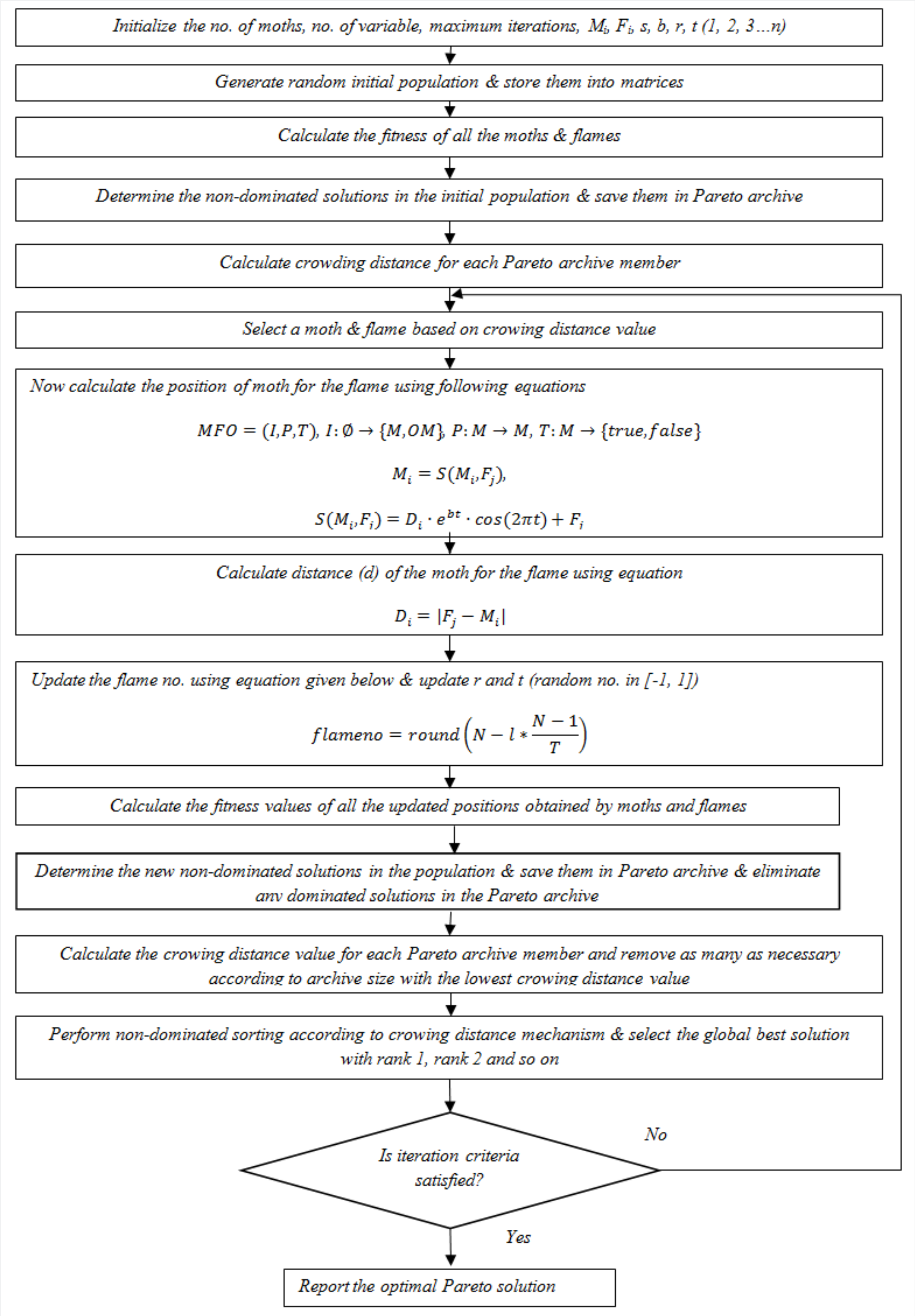
Initialize the no. of moths, no. of variable, maximum iterations, M_i, F_i, s, b, r, t (1, 2, 3...n)
Generate random initial population & store them into matrices
Calculate the fitness of all the moths & flames
Determine the non-dominated solutions in the initial population & save them in Pareto archive
Calculate crowding distance for each Pareto archive member
Select a moth & flame based on crowing distance value
Now calculate the position of moth for the flame using following equations
$MFO = (I,P,T)$, $I: \emptyset \to \{M, OM\}$, $P: M \to M$, $T: M \to \{true, false\}$
$M_i = S(M_i, F_j)$,
$S(M_i, F_j) = D_i \cdot e^{bt} \cdot cos(2\pi t) + F_j$
Calculate distance (d) of the moth for the flame using equation
$D_i = |F_j - M_i|$
Update the flame no. using equation given below & update r and t (random no. in [-1, 1])
$flameno = round\left(N - l * \frac{N-1}{T}\right)$
Calculate the fitness values of all the updated positions obtained by moths and flames
Determine the new non-dominated solutions in the population & save them in Pareto archive & eliminate anv dominated solutions in the Pareto archive
Calculate the crowing distance value for each Pareto archive member and remove as many as necessary according to archive size with the lowest crowing distance value
Perform non-dominated sorting according to crowing distance mechanism & select the global best solution with rank 1, rank 2 and so on
Is iteration criteria satisfied?
No
Yes
Report the optimal Pareto solution

Figure 3: Flow chart of NSMFO.

How to cite this article: Pradeep J, Indrajit N T. Non-Dominated Sorting Moth Flame Optimizer: A Novel Multi-Objective Optimization Algorithm for Solving Engineering Design Problems. Eng Technol Open Acc. 2018; 2(1): 555579. DOI: 10.19080/ETOAJ.2018.02.555579

Results Analysis on Test Functions

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

a. A set of unconstraint and constraint standard multi-objective test functions

b. Tested on well-known engineering design problems

c. 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

NSMFO 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:

a. Standard multi-objective unconstrained test functions (KUR, FON, ZDT1, ZDT2, ZDT3, ZDT4, SCHN1, and SCHN2)

b. Standard multi-objective constrained test functions (TNK, OSY, BNH, SRN, and CONST)

c. Real world engineering multi-objective design problem (Four bar truss design, welded beam design, speed reducer and disk brake design problem)

d. Modelling 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 NSMFO algorithm.

To deal with real world engineering design problem is really a typical task with unknown search space, in this article we include 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 pares to front with respect to their original or true pareto fronts.

For numeric as well as qualitative performance of purposed NSMFO 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 this performance indicating metric are as follows:

$$GD = \frac{\sqrt{\int_{i=1}^{n_{PFs}} (d_i)^2}}{n} \tag{14}$$

Where d_i shows the Euclidean distance (calculated in the objective space) between the ith 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.

$$\text{Ä} = \frac{\left| d_l + d_m + \int_{i=1}^{n_{PFs}} |d_i - d| \right|}{d_l + d_m + (n-1)d} \tag{15}$$

Where, d_l , d_m are Euclidean distances between extreme solutions in true pared to front and obtained pared to 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.

$$S = \sqrt{\frac{1}{n_{PFs} - 1} \int_{i=1}^{n_{PFs}} (d_i - d)^2} \tag{16}$$

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.

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 Figure4.

All the statistical results are shown Table 1 suggests that the NSMFO 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 MFO (NSMFO) can be seen in the Table 1, represents a greater robustness and accuracy of NSMFO algorithm in terms of mean and standard deviation with the help of GD, diversity matrix along with computational time. However, proposed NSMFO 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 NSMFO algorithm shows almost complete coverage with respect to true pareto front (Figure4).

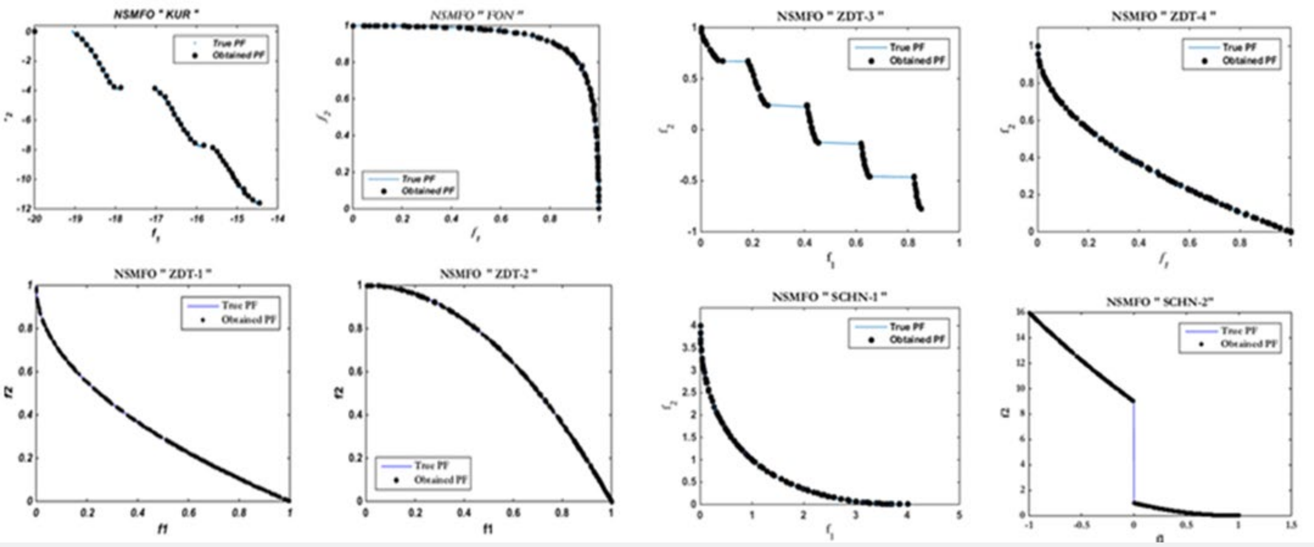

Figure 4: Best Pareto optimal front of KUR, FON, ZDT1, ZDT2, ZDT3, ZDT4, SCHN1 and SCHN2 obtained by the NSMFO algorithm.

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

Algorithm→Function↓	PFs	NSMFO MEAN ± SD	MOSOS MEAN ± SD	MOCBO MEAN ± SD	MOPSO MEAN ± SD	NSGA-II MEAN ± SD
	GD	0.00639 ± 0.00214	0.0075 ± 0.0042	0.0083 ± 0.0062	0.015 ± 0.0075	0.0301 ± 0.0043
KUR	Δ	0.02874 ± 0.01124	0.0295 ± 0.0122	0.0357 ± 0.0236	0.0991 ± 0.031	0.0362 ± 0.0240
	CT	6.75954 ± 0.54278	10.7413 ± 0.822	7.9531 ± 0.5823	8.0532 ± 0.621	20.4368 ± 3.102
	GD	0.00164 ± 0.00022	0.0019 ± 0.0002	0.0022 ± 0.0003	0.0042 ± 0.000	0.0026 ± 0.0003
FON	Δ	0.29102 ± 0.03367	0.3875 ± 0.0062	0.3955 ± 0.0068	0.4158 ± 0.008	0.3987 ± 0.0082
	CT	10.6782 ± 0.56894	11.4013 ± 1.140	8.6606 ± 0.8862	8.732 ± 0.9134	22.0323 ± 4.522
	GD	0.24853 ± 0.07316	0.3325 ± 0.0256	0.3337 ± 0.0319	0.3348 ± 0.035	0.3352 ± 0.038
ZDT-1	Δ	0.38639 ± 0.00952	0.3803 ± 0.0122	0.3825 ± 0.0125	0.3876 ± 0.024	0.3905 ± 0.0220
	CT	2.49985 ± 0.00425	8.2351 ± 0.0204	3.1435 ± 0.0193	3.7533 ± 0.006	11.2681 ± 0.364
	GD	0.07205 ± 0.00077	0.0731 ± 0.0010	0.0729 ± 0.0005	0.0733 ± 0.001	0.0725 ± 0.0004
ZDT-2	Δ	0.03903 ± 0.07408	0.4307 ± 0.0007	0.4316 ± 0.0007	0.4321 ± 0.001	0.431 ± 0.00075
	CT	5.88972 ± 0.02147	8.2345 ± 0.0457	3.1502 ± 0.0130	3.6113 ± 0.014	11.2811 ± 0.024
	GD	0.07264 ± 0.03926	0.1022 ± 0.5187	0.0982 ± 0.5007	0.1235 ± 0.009	0.1147 ± 0.0039
ZDT-3	Δ	0.75463 ± 0.20494	0.6537 ± 0.0052	0.65325 ± 0.002	0.8234 ± 0.108	0.7386 ± 0.0474
	CT	5.87754 ± 0.24445	13.4567 ± 0.129	6.2846 ± 0.1059	8.3764 ± 0.231	14.3406 ± 0.144
	GD	0.47752 ± 0.00023	0.5015 ± 0.0006	0.5078 ± 0.0013	0.5146 ± 0.001	0.5204 ± 0.0019
ZDT-4	Δ	0.34353 ± 0.01277	0.4585 ± 0.0073	0.4795 ± 0.0079	0.6543 ± 0.024	0.7003 ± 0.0089
	CT	5.66881 ± 0.12235	13.9022 ± 0.121	6.6922 ± 0.1440	8.8203 ± 0.218	14.8102 ± 0.170
	GD	0.00915 ± 0.00090	0.0028 ± 0.0024	0.0031 ± 0.0032	0.0032 ± 0.003	0.0034 ± 0.0042
SCHN-1	Δ	0.57809 ± 0.01973	0.5295 ± 0.1312	0.5302 ± 0.1356	0.8582 ± 0.164	0.5502 ± 0.1360
	CT	4.77895 ± 1.22584	8.2135 ± 1.121	5.4845 ± 1.1320	5.5721 ± 1.133	17.9121 ± 2.162
	GD	0.04569 ± 0.00200	0.0705 ± 0.0215	0.0932 ± 0.0228	0.1497 ± 0.022	0.3096 ± 0.0217
SCHN-2	Δ	0.67586 ± 0.04448	0.7821 ± 0.0512	0.801 ± 0.08326	0.8652 ± 0.060	0.9562 ± 0.0921
	CT	5.69984 ± 0.14458	8.7015 ± 0.4532	5.9751 ± 0.2821	6.0272 ± 0.582	18.421 ± 2.1802

How to cite this article: Pradeep J, Indrajit N T. Non-Dominated Sorting Moth Flame Optimizer: A Novel Multi-Objective Optimization Algorithm for Solving Engineering Design Problems. Eng Technol Open Acc. 2018; 2(1): 555579. DOI: 10.19080/ETOAJ.2018.02.555579

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 NSMFO 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).

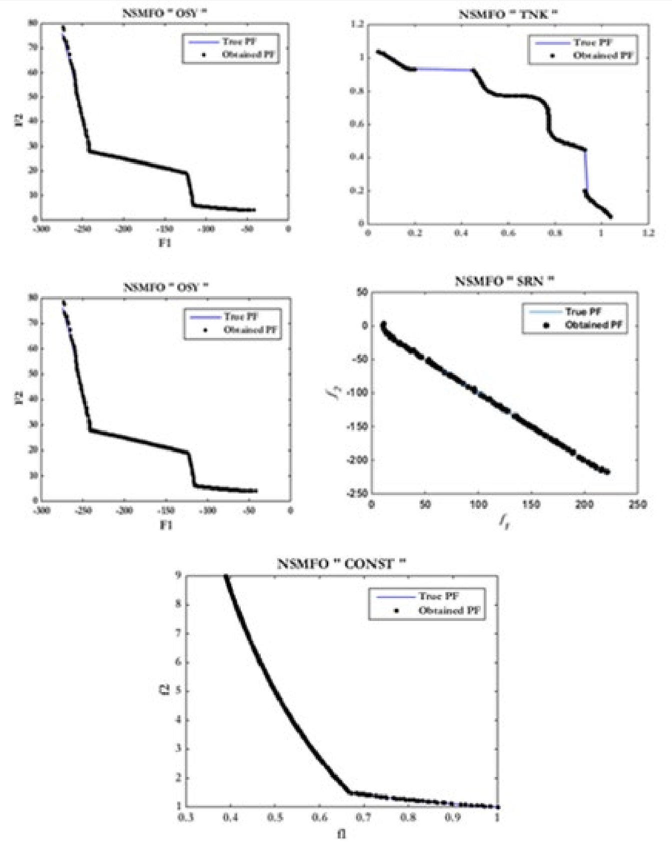

Figure 5: Best Pareto optimal front TNK, OSY, BNH, SRN and CONST obtained by NSMFO.

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

Algorithm↓ Function→	PFs	NSMFO MEAN ± SD	MOSOS MEAN ± SD	MOCBO MEAN ± SD	MOPSO MEAN ± SD	NSGA-II MEAN ± SD
	GD	0.34486 ± 0.00240	0.1508 ± 0.0040	0.1528 ± 0.0051	0.1576 ± 0.0062	0.1542 ± 0.0072
TNK	Δ	0.64252 ± 0.05689	0.1206 ± 0.0423	0.1242 ± 0.0512	0.1286 ± 0.0522	0.126 ± 0.06242
	CT	10.2356 ± 0.04511	15.1286 ± 0.063	11.0104 ± 0.052	12.0212 ± 0.054	17.4204 ± 0.055
	GD	0.10156 ± 0.00024	0.1196 ± 0.0031	0.1210 ± 0.0041	0.1282 ± 0.0042	0.1242 ± 0.0043
OSY	Δ	0.56186 ± 0.05689	0.5354 ± 0.0616	0.5422 ± 0.0712	0.5931 ± 0.0721	0.5682 ± 0.0751
	CT	11.5644 ± 0.02022	20.2124 ± 0.032	12.2104 ± 0.030	14.6420 ± 0.042	24.2204 ± 0.039
	GD	0.14356 ± 0.00428	0.1436 ± 0.0062	0.1498 ± 0.0076	0.1644 ± 0.0078	0.1566 ± 0.0042
BNH	Δ	0.43479 ± 0.02946	0.4288 ± 0.0625	0.4798 ± 0.0721	0.4975 ± 0.0632	0.4892 ± 0.0832
	CT	08.5422 ± 0.04552	16.2664 ± 0.054	9.1544 ± 0.0420	9.7452 ± 0.0464	19.652 ± 0.0511
	GD	0.05251 ± 0.01581	0.0988 ± 0.0014	0.1018 ± 0.0015	0.1125 ± 0.0026	0.1024 ± 0.0032
SRN	Δ	0.20415 ± 0.00098	0.2295 ± 0.0017	0.2352 ± 0.0019	0.2730 ± 0.0023	0.2468 ± 0.0018
	CT	5.44456 ± 0.00112	12.3254 ± 0.012	7.3251 ± 0.0082	9.2134 ± 0.0083	17.0231 ± 0.023
	GD	0.34183 ± 0.05007	0.5162 ± 0.0021	0.5202 ± 0.0034	0.5854 ± 0.0036	0.5532 ± 0.0041
CONST	Δ	0.75355 ± 0.00090	0.7122 ± 0.0072	0.7235 ± 0.0083	0.7344 ± 0.0084	0.8126 ± 0.0087
	CT	6.88856 ± 0.00039	10.0112 ± 0.003	5.2252 ± 0.0028	6.4766 ± 0.0035	14.0892 ± 0.003

Table 2 suggests that the NSMFO algorithm comparatively performs better than other four algorithms for most of the standard constrained test functions employed. The best Pareto optimal fronts in Figure 5 also helps in proving since all the Pareto

How to cite this article: Pradeep J, Indrajit N T. Non-Dominated Sorting Moth Flame Optimizer: A Novel Multi-Objective Optimization Algorithm for Solving Engineering Design Problems. Eng Technol Open Acc. 2018; 2(1): 555579. DOI: 10.19080/ETOAJ.2018.02.555579

optimal solutions exactly follow the true Pareto fronts obtained from by NSMFO 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 suggest that NSMFO algorithm has a capability to solve various type of constraint problem. All the constraint test functions are mathematically given in Appendix B.

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 NSMFO algorithm with respect to other algorithms to solve them, numeric results are given in Table 3 and Figure 6 shows the best optimal front obtained by NSMFO algorithm.

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

PFs→ Methods ↓	GD	S
	MEAN ± SD	MEAN ± SD
NSMFO	0.12643 ± 0.03274	1.1815 ± 0.1445
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

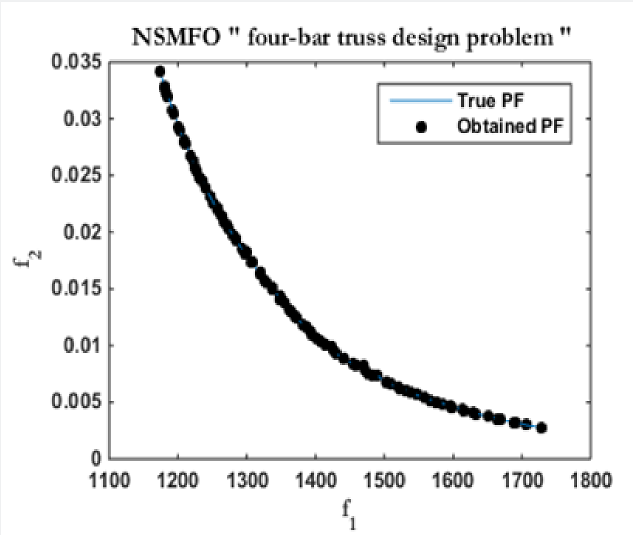

Figure 6: Pareto optimal front obtained by the NSMFO Algorithm for "Four –bus truss design problem".

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 Figure 6 It consists of two minimization objectives displacement and volume with four design control variable mathematically given in Appendix C (Figure 7).

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 Figure 7 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 NSMFO algorithms on four-bar truss design problem in terms mean and standard deviation.

PFs→ Methods ↓	GD	S
	MEAN ± SD	MEAN ± SD
NSMFO	0.95526 ± 0.33184	163.5783 ± 06.9575
MOWCA	0.98831 ± 0.17894217	16. 68520 ± 2.69694436
NSGA-II	9.843702 ± 7.08103039	2.765449155± 3.53493787
μ-GA	3.117536 ± 1.67810867	47.80098 ± 32.80151572
PAES	77.99834 ± 4.21026087	16.20129 ± 4.26842769

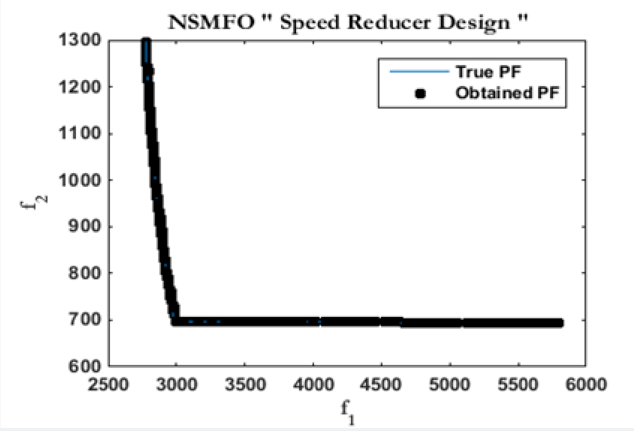

Figure 7: Pareto optimal front obtained by the NSMFO Algorithm for "Speed Reducer design problem".

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 Figure 8. 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 NSMFO algorithms on welded-beam design problem in terms mean and standard deviation.

PFs→ Methods ↓	GD	Δ
	MEAN ± SD	MEAN ± SD
NSMFO	0.13406 ± 0.018937	0.54221 ± 0.047701
MOWCA	0.04909 ± 0.02821	0.22478 ± 0.0928
NSGA-II	0.16875 ± 0.0803	0.88987 ± 0.11976
paε-ODEMO	0.09169 ± 0.00733	0.58607 ± 0.04366

How to cite this article: Pradeep J, Indrajit N T. Non-Dominated Sorting Moth Flame Optimizer: A Novel Multi-Objective Optimization Algorithm for Solving Engineering Design Problems. Eng Technol Open Acc. 2018; 2(1): 555579. DOI: 10.19080/ETOAJ.2018.02.555579

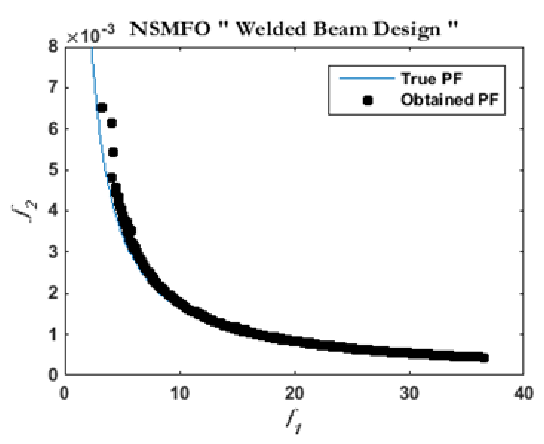

Figure 8: Paretooptimal front obtained by the NSMFO Algorithm for "Welded Beam Design problem".

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 Figure 9. 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 NSMFO algorithms on the Disk brake design problem in terms mean and standard deviation.

PFs→ Methods ↓	GD	Δ
	MEAN ± SD	MEAN ± SD
NSMFO	0.0233 ± 0.26810	0.83651 ± 0.08439
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

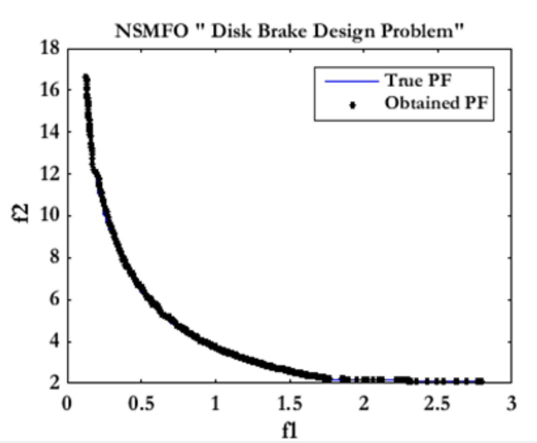

Figure 9: Pare to optimal front obtained by the NSMFO Algorithm for "Disk brake design problem".

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 NSMFO algorithm is covers almost whole solutions that are the actual/true solutions of an engineering design problem. From all above testedfunction we can conclude that problem either it consists of constraints or unconstraint problem NSMFO 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.

Formulation of economic constrained emission dispatch (ECED) with integration of wind power (WP)

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:

$$s(v) = (k/c)(v/c)^{k-1} * \exp\left[-(v/c)^{k}\right],\ v \geq 0 \tag{17}$$

$$S(v) = 1 - \exp\left[-(v/c)^{k}\right], v \geq 0 \tag{18}$$

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:

$$P_{wind} = \begin{cases} 0, & v < v_{in}\, or\, v \geq v_{out} \\ P^{rated}\dfrac{v - v_{in}}{v_{rated} - v_{in}} & v_{in} \leq v < v_{rated} \\ P_{rated} & v_{in} \leq v < v_{out} \end{cases} \tag{19}$$

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:

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

(20)

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

Modelling of ECEDWP problem: As wind power is formulated as system constraint, sothe objective function of economic emission dispatch problem (EEDP) stays on unchanged as classical EEDP:Fuel cost objective is given by:

$$Minimization \quad S(P_i) = \sum_{i}^{N}(a_i + b_i P_i + c_i P_i^2) \tag{21}$$

Where, the thermal power generators cost coefficients are a_i, b_i, c_i for ith generator, Sum of the total fuel cost of the system and N is the total number of generators.

Total Emission is calculated by:

$$Minimization\ E(P_i)=\sum_{i}^{N}[\{(\alpha_i+\beta_iP_i+\gamma_iP_i^2)*10^{-2}\}+\delta_i*\exp(\varphi_i*P_i)] \quad (22)$$

Where, $\alpha_i,\beta_i,\gamma_i,\delta_i$ and φ_i are emission coefficients with valve point effect taking into consideration for ith thermal generator.

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

$$P_r\sum_{i=1}^{N}(P_i+P_{Wind}\geq P_D+P_{Loss})\geq\eta_{pbc} \quad (23)$$

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:

$$P_{Loss}=\sum_{i=1}^{N}\sum_{j=1}^{N}P_iB_{ij}P_j+\sum_{i=1}^{N}P_iB_{i0}+B_{00} \quad (24)$$

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

$$P_r\left\{P_{Wind}<P_D+P_{Loss}-\sum_{i=1}^{N}P_i\right\}=F\left(P_D+P_{Loss}-\sum_{i=1}^{N}P_i\right)\leq 1-\eta_{pbc} \quad (25)$$

Assume that the wind turbine have same speed and same direction and combination of Eqs. (25), the power balance constraint is represented as:

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

(26)

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:

$$P_r\left\{\sum_{i=1}^{N}(P_i^{max}-P_i)\geq P_{sr}+t_u*P_{Wind}\right\}\geq\eta_{urc} \quad (27)$$

$$P_r\left\{\sum_{i=1}^{N}(P_i-P_i^{min})\geq t_d*(P_{rated}-P_{Wind})\right\}\geq\eta_{drc} \quad (28)$$

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 ith 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.

Generational capacity constraint: The real output power is bounded by each Generators upper and lower bounds given as:

$$P_i^{Minimum}\leq P_i\leq P_i^{Maximum} \quad (29)$$

40-Operational Thermal Generating Unit

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 NSMFO algorithm is added to Table 7 and best Pareto front obtained by NSMFO algorithm is represented in Figure 10.

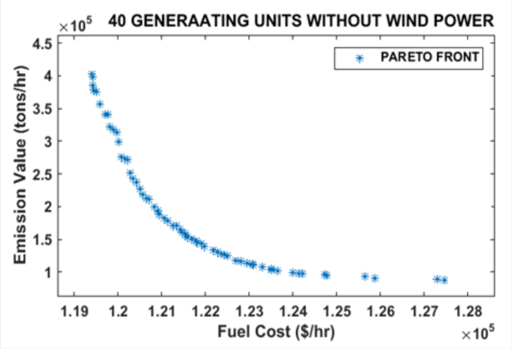

Figure 10: Pareto optimal front obtained by the NSMFO Algorithm for "40 thermal-generator lossless system without wind power".

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

Case Study I	SMODE [45]			NSGAII [45]			MBFA [46]			NSMFO		
	Beste mission	Best cost	Best com promise	Best emission	Best cost	Best com promise	Beste emission	Best cost	Best com promise	Best emission	Best cost	Best com promise
Cost ($/h)	1,56,700	1,19,650	1,24,230	1,28,490	1,24,380	1,26,180	1,29,995	1,21,415	1,23,638	1,27,468	119410	1,24,130
Emission (tons/h)	66,799	3,77,560	96,578	93,002	1,53,560	99,671	1,76,682	3,56,424	1,88,963	88,324	4,03,022	94,452

How to cite this article: Pradeep J, Indrajit N T. Non-Dominated Sorting Moth Flame Optimizer: A Novel Multi-Objective Optimization Algorithm for Solving Engineering Design Problems. Eng Technol Open Acc. 2018; 2(1): 555579. DOI: 10.19080/ETOAJ.2018.02.555579

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 NSMFO algorithm is reported in Table 8 [51] and best optimal front is represented in Figure 11.

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

Case Study-II	SMODE[45]			NSGAII [45]			MOEA/D[51]			NSMFO		
	Best emission	Best cost	Best Com promise point	Best emission	Best cost	Best Com promise point	Best emission	Best cost	Best Com promise	Best emission	Best cost	Best Com prom ise point
10, 245.76	10,1 77.55	10, 225.71	10, 241.72		10, 242.09	10, 241.63	10 ,244.43	10, 242.71	10, 242.80	10, 242.70	10, 224.18	10, 236.58
254.24	322.45	274.29	258.28	257.91	258.37	255.568	257.294	257.156	257.321	275.82	263.42	
Cost	1,53,830	1,16,430	1,23,590	1, 32,410	1,22,610	1,26,240	154,0 0 0	1,15,770	1,20,950	1,39,780	1,19,830	1, 23,450
Emission	54,055	3,85,770	68,855	73,894	1,21,850	78,860	55,754	4,40,240	79,485	60,023	1,01,990	68,734

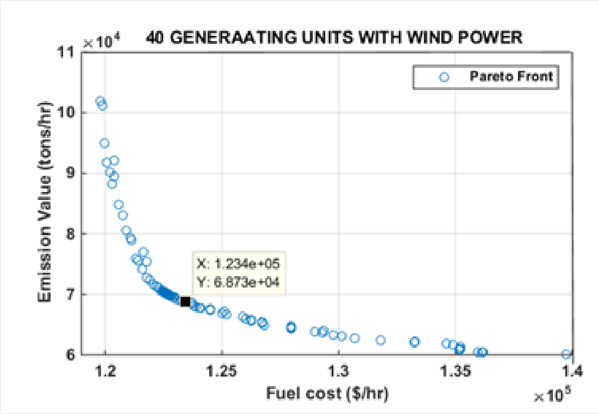

Figure 11: Pareto optimal front obtained by the NSMFO Algorithm for "40 thermal-generator lossless system with wind power".

Result & Discussion

In almost all the cases that we consider in this article where NSMFO algorithm proves its effectiveness in both respects quantitative and qualitative. From plots also evident that NSMFO 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 [48-51]. So, for this application Wilcoxon test (statistical test) is performed. In Table 9 the signed rank test is presented in forth row of each results whereas the calculation time is represented in third 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 NSMFO algorithm performs superior to other algorithms that are considered for comparative purpose. NSMFO algorithm shows good performance in both coverage and convergence as main mechanism that guarantee convergence in MFO and NSMFO are continuously shrink its virtual limitation in the movement of moths and flames for their random walk. Both mechanisms emphasize 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 express that NSGWO algorithm has a little effect of slow convergence at all.

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

		NSMFO	NSGAII [45]			NSMFO	NSGAII [45]
Case Study I Cost	Best	1,19,410	1,24,380	Case Study II Cost	Best	1,19,830	1,22,610
	Worst	1,27,468	1,47,760		Worst	1,39,780	1,73,060
	MeanWilcoxon	122320	1,31,710		Mean	1,26,010	1,34,880
	test (H/P)	1/ 5.42e −10			Wilcoxon	1/ 5.87e −10	
					test (H/P)		
Case Study I Cost Emission	Best	88,324	93,002	Case Study II Emission	Best	60,023	73,894
	Worst	403.022	1,94,830		Worst	1,01,990	1,58,250
	Mean Wilcoxon	1,74,284	1,41,800		Mean	78,356	1,02,120
	test (H/P)	1/ 5.56e −10			Wilcoxon	1/ 5.73e −10	

How to cite this article: Pradeep J, Indrajit N T. Non-Dominated Sorting Moth Flame Optimizer: A Novel Multi-Objective Optimization Algorithm for Solving Engineering Design Problems. Eng Technol Open Acc. 2018; 2(1): 555579. DOI: 10.19080/ETOAJ.2018.02.555579

					test (H/P)		
	Simulation speed (s)	24.57	154.78		Simulation speed (s)	33.04	127.57

NSMFO algorithm has an advantage of high coverage, which is the result of the selection of position of moth and flame and archive selection procedure. All the position is 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.

Conclusion

In this paper the non-dominated MFO multi-objective version of recently proposed moth flame optimizer (MFO) algorithm is proposed known as NSMFO algorithm. This paper also utilizes the transverse orientation strategy for exploration purpose used in its parent MFO version. NSMFO algorithm is developed with equipping MFO with crowding distance criterion, an archive and moth-flame position (accordance to ranking) selection method based on Pareto optimal dominance nature. The NSMFO algorithm is first applied on 17 standard test functions (including eight unconstraint, five constraints 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 NSMFO 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 NSMFO algorithm is able to find Pareto optimal front of any kind of shape. Finally, the result of complex real world ECEDWP problem validates that NSMFO 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 MFO 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 NSMFO algorithm on other real world complex problems. Also, it is worth to investigate and find the best constrained handling technique for this algorithm (Appendix) [48-50].

References

1. Kelley CT (1999)Detection and Remediation of Stagnation in the Nelder--Mead Algorithm Using a Sufficient Decrease Condition, SIAM Journal on Optimization 10(1): 43-55.
2. Vogl TP, Mangis J, Rigler A, Zink W, Alkon D(1988) Accelerating the convergence of the back-propagation method, Biological cybernetics59(4-5): 257-263.
3. Mirjalili S, Mohammad MS, Lewis A(2014) Grey Wolf Optimizer, Advances in Engineering Software, 69: 41-61.
4. Xin-She Yang (2010) The bat algorithm (BA), A Bio-inspired algorithm.
5. Kennedy J, Eberhart R(1995) Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia.
6. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE ComputIntell Mag 1(4):28-39
7. John H Holland(1992) adaptation in natural and artificial systems. MIT Press, Cambridge, UK.
8. Gandomi AH, Yang XS, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. EngComput 29(1):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(5):2592-2612.
10. GandomiAH, AlaviAH, Herd K (2012)A new bio-inspired optimization algorithm, Common Nonlinear Sci. Numer. Simul. 17(12): 4831-4845.
11. Gandomi AH (2014) Interior Search Algorithm (ISA): A Novel Approach for Global Optimization. ISA Trans 53(4): 1168-1183.
12. Beyer HG,SendhoffB (2007) Robust optimization-a comprehensive survey, Computer methods in applied mechanics and engineering 196: 3190-3218.
13. KnowlesJD, WatsonRA, CorneDW (2001) Reducing local optima in single-objective problems by multi-objectivization, in Evolutionary multi-criterion optimization, pp. 269-283.
14. Deb K, Goldberg DE (1993) Analyzing deception in trap functions, Foundations of genetic algorithms2: 93-108.
15. Coello CAC (2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Computer methods in applied mechanics and engineering191(11-12): 1245-1287.
16. WolpertDH, Macready WG (1997) No free lunch theorems for optimization, Evolutionary Computation, IEEE Transactions on1(1): 67-82.
17. Mirjalili S (2015) Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowledge-Based System89: 228-249.
18. Panda,PaniS (2015)Multiobjective colliding bodies optimization, in:Proceedings of 5th Int. Conf. on Soft Computing for Problem Solving, SocProS,IIT Roorkee, India.
19. Coello CA, PulidoGT, LechugaMS (2004) Handling multiple objectives withparticle swarm optimization, IEEE Trans. Evol. Comput 8(3):pp.256-279.

20. CoelloCoelloCA, LechugaMS (2002) MOPSO: a proposal for multiple objectiveparticle swarm optimization, in: Proceedings of the IEEE Congress onEvolutionary Computation, USA, Pp. 1051-1056.

21. DebK, AgrawalS, PratapA, MeyarivanT (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, in Parallel problem solving from nature PPSN VI pp. 849-858.

22. DebK, GoelT (2001) Controlled elitist non-dominated sorting genetic algorithms for better convergence, in Evolutionary multi-criterion optimization pp. 67-81.

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

24. Deb K, Pratap A, Agarwal S, Meyarivan TAMT (2002) A fast and elitistmultiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput 6(2): 182-197.

25. Panda A, Pani S (2016) A Symbiotic Organisms Search algorithm with adaptive penaltyfunction to solve multi-objective constrained optimization problems, Applied Soft Computing 46: 344-360.

26. Ngatchou P, Zarei A, El-SharkawiM (2005) Pareto multi objective optimization, in Intelligent Systems Application to Power Systems, Proceedings of the 13th International Conference, USA.

27. ParetoV (1964)Coursd'economie politique: LibrairieDroz.

28. EdgeworthFY (1881) Mathematical Physics: P. Keagan, London, England.

29. Mirjalili S, Saremi S, Mirjalili SM, CoelhoLdS (2016) Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization, Expert Systems with Applications 47: 106-119.

30. Yang XS (2011) Bat algorithm for multi-objective optimisation, International Journal of Bio-Inspired Computation 3(5):267-274.

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

32. Knowles JD, Corne DW(2000) Approximating the nondominated front using the Pareto archived evolution strategy. EvolComput 8(2): 149-172.

33. AbbassHA, SarkerR, NewtonC (2001) PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems, in Evolutionary Computation, Proceedings of the Congress, South Korea.

34. Zhang Q, Li H (2007) MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition, IEEE transactions on evolutionary computation 11(6).

35. Zitzler E (1999) Evolutionary algorithms for multiobjective optimization: Methods and applications 63: Citeseer.

36. Zitzler E, Thiele L(1999) Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach, Evolutionary Computation, IEEE Transactions 3: 257-271.

37. Sadollah A, Eskandar H, Bahreininejad A, Kim JH (2014) Water cycle algorithm for solving multi-objective optimization problems, Soft Comput 19(9): 2587-2603.

38. Sadollaha A, Eskandarb H, Kim JH (2015) Water cycle algorithm for solving constrained multi-objectiveoptimization problems, Applied Soft Computing 27(c): 279-298.

39. Veldhuizen DAV, Lamont GB (1998) Multiobjective evolutionary algorithm research: A history and analysis, Citeseer.

40. Schott JR (1995) Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization, DTIC Document.

41. Coello CAC (2000) Use of a self-adaptive penalty approach for engineering optimization problems, Computers in Industry, 41(2): 113-127.

42. Coello CC, Pulido GT (2005) Multiobjective structural optimization using a microgenetic algorithm, Structural and Multidisciplinary Optimization30(5): 388-403.

43. Kurpati, Azarm S, Wu J (2002) Constraint handling improvements for multiobjective genetic algorithms, Structural and Multidisciplinary Optimization23(3): 204-213.

44. Ray T, Liew KM (2002) A swarm metaphor for multiobjective design optimization, Engineering optimization 34(2): 141-153.

45. Qu BY, Liang JJ, Zhu YS, Wang ZY, Suganthan PN (2016) Economic emission dispatch problems with stochastic wind power using summation based multi-objective evolutionary algorithm, Information Sciences 351: 48-66.

46. Hota PK, Barisal AK, Chakrabarti R (2010) Economic emission load dispatch through fuzzy based bacterial foraging algorithm, Electr. Power Energy Syst 32(7): 794-803.

47. AbidoMA (2003) Environmental/economic power dispatch using multiobjective evolutionary algorithms: a comparative study, IEEE Trans Power Syst 18(4): 1529-1537.

48. BinhTT, KornU (1997) MOBES: A multiobjective evolution strategy for constrained optimization problems, in The Third International Conference on Genetic Algorithms (Mendel 97).

49. Osyczka, Kundu S (1995) A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm, Structural optimization10(2): 94-99.

50. Srinivasan, Deb K (1994) Multi-objective function optimisation using non-dominated sorting genetic algorithm, Evolutionary Comp 2(3): 221-248.

51. Zhu YS, Wang J (2014) Multi-objective economic emission dispatch considering wind power using evolutionary algorithm based on decomposition, Electr. Power Energy Syst 63: 434-445.

How to cite this article: Pradeep J, Indrajit N T. Non-Dominated Sorting Moth Flame Optimizer: A Novel Multi-Objective Optimization Algorithm for Solving Engineering Design Problems. Eng Technol Open Acc. 2018; 2(1): 555579. DOI: 10.19080/ETOAJ.2018.02.555579

DOI: 10.19080/ETOAJ.2018.02.555579