

## Research Article

# A New Hybrid Metaheuristic Algorithm for Multiobjective Optimization Problems

M.A. Farag<sup>1,\*</sup>, M.A. El-Shorbagy<sup>1,2</sup>, A.A. Mousa<sup>1,3</sup>, I.M. El-Desoky<sup>1</sup>

<sup>1</sup>Department of Basic Engineering Science, Faculty of Engineering, Menoufia University, Shebin El-Kom, Egypt

<sup>2</sup>Department of Mathematics, College of Science and Humanities in Al-Kharj, Prince Sattam bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia

<sup>3</sup>Mathematics and Statistics Department, Faculty of Science, Taif University, Taif, Saudi Arabia

## ARTICLE INFO

### Article History

Received 16 Jan 2020

Accepted 10 Jun 2020

### Keywords

Sine-cosine algorithm

Nondominated sorting genetic algorithm

Multiobjective optimization problems

The economic emission dispatch problem

TOPSIS

## ABSTRACT

The elitist nondominated sorting genetic algorithm (NSGA-II) is hybridized with the sine-cosine algorithm (SCA) in this paper to solve multiobjective optimization problems. The proposed hybrid algorithm is named nondominated sorting sine-cosine genetic algorithm (NS-SCGA). The main idea of this algorithm is the following: NS-SCGA integrates the merits of exploitation capability of NSGA-II and exploration capability of SCA for a better search ability and speeds up the searching process. The performance of NS-SCGA is tested on the set of benchmark functions provided for CEC09. The NS-SCGA results are compared with other recently developed multiobjective algorithms in terms of convergence, spacing, and spread of the obtained nondominated solutions to the true Pareto front. The statistical analysis of the results obtained is performed by nonparametric Friedman and Wilcoxon signed-rank tests. The results prove that NS-SCGA is superior to or competitive with other multiobjective optimization algorithms considered in the comparison. Furthermore, the economic emission dispatch problem (EEDP) is solved by NS-SCGA. The operating cost (fuel cost) and pollutant emission of the standard IEEE 30-bus network with six generating units are minimized simultaneously by the NS-SCGA considering the losses. The results show the superiority of NS-SCGA and confirm its ability in solving EEDP. Finally, TOPSIS technique is applied to choose the best compromise solution from the obtained Pareto-optimal solutions of EEDP according to the decision-maker's preference.

© 2020 The Authors. Published by Atlantis Press SARL.

This is an open access article distributed under the CC BY-NC 4.0 license (<http://creativecommons.org/licenses/by-nc/4.0/>).

## 1. INTRODUCTION

In the real world, many engineering, science, finance, economics, and logistics optimization problems usually have more than one conflict objective. Problems of this kind are named as multiobjective optimization problems (MOPs).

MOPs are different from single-objective optimization, as there is no single solution that optimizes all objective functions simultaneously. It makes solving MOPs a challenging job, because of the conflicting nature of objective functions (improving one objective leads commonly to the deterioration of another, a.k.a improving one worsens others). In addition, the obtained set of solutions from solving MOPs (Pareto-optimal) has main features such as the following: each solution produces trade-offs between the different objectives, and no solution is subaltern to the other solutions [1].

Different conventional algorithms have been developed to solve MOPs such as the following: the goal programming, the hierarchical optimization algorithms, the  $\epsilon$ -constraint technique, and the objective weighted method [2]. These techniques converted the vector of objectives of MOP into a single objective by generating one particular nondominated solution at a time. The conventional algorithms suffer from some drawbacks, such as the differentiation

requirements of the objective function and constraints, the need for a good initial point, the expensive computation for Jacobian and/or Hessian [1,3]. Also, these methods are usually inefficient in case of some objectives involving concavity, discontinuity, uncertainty, and noise in their search space [4]. Therefore, they cannot be used universally in optimizing the complex MOPs.

Therefore, over the past two decades, much emphasis has been laid on developing metaheuristic methods to yield a computationally efficient and convergent procedure. Among these metaheuristic methods are evolutionary algorithms (EAs) and intelligent swarm algorithms [5]. Most of the metaheuristic algorithms can obtain the Pareto-optimal solutions of MOPs, which are difficult to obtain by the conventional algorithms within a reasonable computation time, as proven in [5].

The most popular metaheuristic algorithms for solving MOPs are the following: multiobjective genetic algorithm (MOGA) [6,7], nondominated sorting genetic algorithm (NSGA-II) [8], Niche Pareto genetic algorithm (NPGA) [9], a multiobjective differential-evolution algorithm (MO-DEA) [10], strength Pareto evolutionary algorithm (SPEA) [11], SPEA2 [12], Pareto archived evolutionary strategy (PAES) [13], multiobjective evolutionary algorithm (MOEA) based on decomposition (MOEA/D) [14], Liu Li algorithm [15], indicator-based EA (IBEA) [16], archived multiobjective simulated annealing (AMOS) [17], multiobjective particle

\*Corresponding author. Email: [mai.farag2015@gmail.com](mailto:mai.farag2015@gmail.com)

swarm optimization (MOPSO) algorithm [18], clustering multi-objective EA (clustering MOEA) [19], multiple trajectory search (MTS) [20], a multiobjective teaching learning based optimization (MO-TLBO) algorithm [21], multiobjective firefly algorithm for continuous optimization [22], multiobjective cuckoo search [23], a multiobjective ant colony algorithm [24], multiobjective optimization (MO) algorithm based on artificial algae (MO-AAA) [1], multiobjective ant lion optimizer [25], water cycle algorithm [26], multiobjective grey wolf optimizer (MOGWO) [27], dragonfly algorithm [28], multiobjective sine-cosine algorithm (SCA) [29,30], and others; it is difficult to mention all.

In addition, the hybrid algorithms have also been developed and applied to a variety of MOPs and their application such as [31–36]. Hybrid algorithms can be more efficient than the basic versions of the basic approaches, because of the fact that the hybrid algorithm benefits from all the advantages of the basic algorithms. Many researchers proposed hybrid metaheuristic algorithms for this aim. The literature proves that hybridizing metaheuristic techniques is one of the ways of using the advantages of many techniques. It also aids in developing an efficient technique to solve optimization problems, such as sine-cosine crow algorithm [37], NSGA-II/EDA hybrid EA [38], nondominated sorting moth-flame optimization (NS-MFO) [39], K-means-clustering-based EA [40], and GA-based clustering technique [41].

Genetic algorithms (GAs) are the most popular metaheuristic algorithms that introduced at 1975. GAs are well suited to solve MOPs by searching the Pareto front step by step. The primary reason for using GA for MO is its ability to deal with a set of potential solutions (population) and finding Pareto-optimal solutions at each simulation run. The ability of GA in the convergence and searching for the global optimum makes it possible to obtain a diverse solutions' set for the complex problems in multimodal, nonconvex, and discontinuous solution spaces [42]. Therefore, for MOPs, a variety of GA methods can be suggested: random weighted GA (RWGA), NPGA, nondominated sorting GA (NSGA), MOGA, SPEA, and NSGA-II. Of all the aforementioned methods, NSGA-II is a recently and efficiently well-known MOEA which has captured more attention due to its efficiency in producing better Pareto front for any number of objectives [43].

NSGA-II, introduced in 2000 [8], is an improved version of NSGA. NSGA-II uses the elitist technique, which consists of sorting the population at different "fronts" using the nondominated ranking technique with a particular bookkeeping strategy. The crowding distance (CRD) sorting is a secondary ranking that measures the population density around each solution and sorts the individuals, which enter the same front. The best solutions are then chosen according to nondominance and diversity. NSGA-II is computationally fast, incorporates elitism, and converges better in the obtained Pareto front compared to other MOEAs. The NSGA-II has been proved as a very efficient multiobjective GA which had been used successfully in solving various MOPs [44–47]. The algorithm's calculation performance depends on the optimizing strategy inside of the algorithm itself. Besides this, the calculation performance of one algorithm varies when dealing with various optimization problems. So, various researchers presented some improvement strategies into the NSGA-II to deal with the shortcomings when solving some complex optimization problems.

SCA is a recent metaheuristic algorithm presented by Mirjalili in 2016 [48]. SCA proved its efficiency in solving single-objective optimization [48]. SCA updates its solutions according to the trigonometric sine and cosine functions, where it oscillates around the best solution. This algorithm has been studied and investigated by many researchers and applied to solve many problems in different fields [29,49–52] due to its higher exploration capability, simple implementation, and less parameter setting. But SCA suffers from slow convergence and speed, low optimization precision, and weak exploitation capability. The algorithm is often trapped in local minima [53]. According to this, in the last two years, many researches proposed various improvements in SCA from different perspectives to handle these drawbacks [54–56].

In this article, a novel optimization algorithm called nondominated sorting sine-cosine genetic algorithm (NS-SCGA) is presented for solving MOPs. The proposed algorithm combined NSGA-II and SCA to integrate the merits of both techniques. The main idea of the proposed algorithm is the following: NS-SCGA integrates the merits of exploitation capability of NSGA-II and exploration capability of SCA for a better search ability as well as speeding up the searching process. The nondominated sorting and CRD techniques of NSGA-II divide the population into several ranks, which ensure Pareto dominance and solution diversity in NS-SCGA. The population evolves using the SCA mechanism along with the genetic operators, which integrate the merits of exploitation capability of genetic operators and the exploration capability of SCA. The search accuracy and performance of the NS-SCGA are verified on the set of benchmark functions provided for the 2009 IEEE Congress on Evolutionary Computation (CEC09). They are also compared with other recently developed multiobjective algorithms in terms of convergence, spacing, and spread of obtained nondominated solutions along the true Pareto front. The Lexicographic ordering (LO) is used to assess the overall performance of the proposed NS-SCGA among the compared techniques. The statistical analysis of the experimental results is also carried out by nonparametric Friedman and Wilcoxon signed-rank tests. The computational results prove that the NS-SCGA is superior to or competitive with other MO algorithms considered in the comparison.

The economic emission dispatch problem (EEDP) is a crucial task in the planning and operation of the power systems [57]. EEDP in power systems is a task of the determination of an optimum schedule of a generation of the generating units to minimize the total generation cost and pollutants emission within the constraints of the power system. EEDP is becoming more and more worthwhile for not only resulting in great economic benefit but minimizing the emission of pollutants also. EEDP is considered a highly constrained conflicting MOP, and it is very complex to solve due to its nonlinearity nature.

Different algorithms have been presented in the literature to solve EEDP including conventional optimization methods and metaheuristic algorithms. Among the conventional optimization algorithms which had been used in solving EEDP are the following: a linear programming method [58], weighted sum method [59], lambda iteration method [60],  $\epsilon$ -constraint method, Lagrangian multiplier method [61], and quadratic programming [62]. But these conventional techniques need the curves of incremental cost to be piece-wise or increasing monotonically linear. But, virtually, the

input characteristics of the generators are highly nonsmooth and nonlinear [63]. In addition, due to the nonlinearity and nonconvexity that arise from the effect of the valve point, using these conventional techniques is not recommended and is not suitable for a complex EEDP [64].

On the other hand, some metaheuristic algorithms, especially EAs and swarm intelligence algorithms, were used to solve EEDP [64–67]. Here, the NS-SCGA which is a hybrid algorithm of two metaheuristic search techniques, is implemented to solve the standard IEEE 30-bus with the six-generator network. The proposed NS-SCGA is compared to other MO algorithms, and the effectiveness of NS-SCGA in solving the EEDP is demonstrated.

Finally, Technique for Order Performance by Similarity to Ideal Solution (TOPSIS) is used to help the decision-maker (DM) to identify the best compromise solution from the nondominated solutions obtained by the proposed NS-SCGA [68]. TOPSIS takes into account the relative priorities of the two objective functions (minimum fuel cost and the minimum emission) in the Pareto-optimal solutions on the basis of the DM perceptions. TOPSIS technique is based upon distance maximization from a nadir point (NP) and distance minimization from the negative ideal solution simultaneously. A NP is constructed by the worst objective values of the solutions of the entire Pareto-optimal set.

The main contributions of the present article are as follows:

- i. A novel optimization algorithm, called NS-SCGA, is presented and evaluated.
- ii. A comparative study of the optimization results using NS-SCGA is presented to 20 benchmark multiobjective problems.
- iii. The results of 20 benchmark functions obtained by NS-SCGA are compared with other newly developed algorithms, and the LO, Friedman test and Wilcoxon signed-rank test are used for evaluating the overall performance of the comparing algorithms.
- iv. NS-SCGA is applied to EEDP and the results are compared with other optimization algorithms.
- v. The best compromise solution of EEDP has been extracted by the TOPSIS technique from nondominated solutions according to the DM's preference.

The rest of this paper is organized as follows: The preliminaries about the MO are briefly studied in Section 2. Section 3 describes the NSGA-II briefly. Section 4 includes the description of the SCA. The explanation about NS-SCGA is given in Section 5. Section 6 presents and discusses the results. Finally, Section 7 includes the conclusion of this paper.

## 2. MULTIOBJECTIVE OPTIMIZATION PROBLEMS

The MO deals with more than one objective function. In order to solve the MOPs, all objectives functions must be optimized simultaneously. In general, there are two categories of MOPs: one is unconstrained MOPs (UCMOPs) and the other is constrained MOPs

(CMOPs). The general formulation of the MOP is given by [69]

$$\begin{aligned} & \text{Minimize } F(\mathbf{y}) = \{f_1(\mathbf{y}), f_2(\mathbf{y}), \dots, f_k(\mathbf{y})\} \\ & \text{subject to } Q_t(\mathbf{y}) \geq 0 \quad t = 1, 2, \dots, n \\ & \quad E_i(\mathbf{y}) = 0 \quad i = 1, 2, \dots, q \\ & \quad \text{Lower}_j \leq y_j \leq \text{Upper}_j \quad j = 1, 2, \dots, \text{dim} \end{aligned} \quad (1)$$

where  $\mathbf{y}$  is a decision vector bounded in decision space  $\Psi$ ,  $R^K$  refers to the objective space,  $\text{dim}$  is the dimension of the problem,  $q$  is the number of equality constraints, and  $n$  is the number of inequality constraints. In addition  $\text{Upper}_j$  and  $\text{Lower}_j$  represent the upper and lower limits of the  $j$ th decision variable and  $F : \Psi \rightarrow R^K$  is the mapping function that consists of  $K$  real-valued objective functions.

In MO, the objectives are often conflicting and need special considerations. Therefore, comparing the MOPs solutions by the relational operators is no longer effective, because of the presence of more than one criterion for a comparison [70]. Consequently, other operators are needed to find out and measure how much a solution is better than another. The Pareto dominance is the most widely used operator for comparing the MOPs solutions which can be defined as follows [71]:

**Pareto Dominance.** It is the case if  $q$  and  $p$  are two feasible solutions belonging to  $\Psi$ . A solution  $p$  dominates a solution  $q$  denoted as ( $p < q$ ), iff

$$\begin{aligned} \forall r \in \{1, 2, \dots, k\}, f_r(p) \leq f_r(q) \\ \text{and at least } \exists t \{1, 2, \dots, k\}, f_t(p) < f_t(q) \end{aligned} \quad (2)$$

The Pareto-optimal solution can be defined as follows:

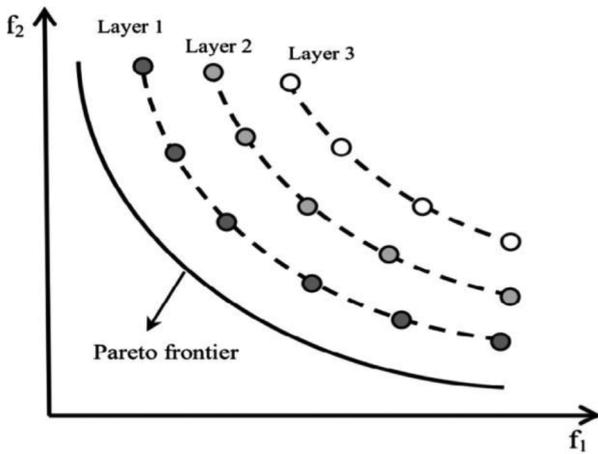
**Pareto-Optimal Solution.** The feasible solution,  $p^* \in \Psi$ , is called a Pareto-optimal solution, iff  $\nexists p \in \Psi$  such that  $p < p^*$ .

The set of all Pareto-optimal solutions is called the Pareto-optimal set (POS), while the Pareto-optimal front (POF):  $PF = \{F(\mathbf{y}) \in R^K | \mathbf{y} \in \text{POS}\}$  is the set of all Pareto-optimal objective vectors [25]

## 3. THE ELITIST NSGA-II

NSGA-II is a population-based algorithm, which is widely used in solving MOPs [8]. In this algorithm, the individuals are divided into nondominated layers on the basis of the concept of Pareto and sorting nondominated solutions as graphically explained in Figure 1.

In NSGA-II, the offspring population  $O_t$  (of size  $NS$ ) is first generated from the parent population ( $P_t$ ) by the genetic operators.  $O_t$  is added to  $P_t$  to generate an intermediate population ( $IN_t$ ) of size  $2NS$ . All the individuals in  $IN_t$  are distributed and sorted into different dominated layers. The ranking process inside a layer is according to the crowded-comparison operator. The procedure for sorting the solutions into the nondominated layers and the procedure for calculating CRD will be discussed in the following subsection. The new population is obtained by selecting solutions from nondominated layers. The individuals are chosen based on the priority assigned to the layers, from the best layer firstly, then from next best, and so on until  $NS$  solutions are created. However, if some layer should be incompletely taken, then the crowding comparison needs to be taken into account and the selection of  $NS$  individuals depends on the CRD [72].



**Figure 1** | The schematic view of grouping the solutions in different layers in nondominated sorting genetic algorithm (NSGA-II) [73].

### 3.1. Sorting the Individuals into Nondominated Layers (The Fast Nondominated Sorting Technique)

For each solution  $j$  in the intermediate population ( $IN_t$ ), two variables are defined: domination count ( $Dom_j$ ) and dominated set ( $Set_j$ ).  $Set_j$  is a set of solutions that the individual  $j$  dominates and  $Dom_j$  is the number of solutions that dominate the individual  $j$ . The pseudocode of the fast nondominated sorting technique is given in Figure 2. The process of sorting the population into nondominated layers and calculating “layer” for each solution is given in the following steps [74]:

- Step 1.** Evaluate all the objective functions for each solution.
- Step 2.** Assign a domination count zero for all the solutions and put them in the first nondominated layer.
- Step 3.** For each individual  $j$  with  $Dom_j = 0$ , each  $g$  individual of its set  $Set_j$  is visited and its domination count  $Dom_g$  is reduced by one. While reducing  $Dom_g$  count, in case it becomes  $Dom_g = 0$  the corresponding individual  $g$  is assigned in 2nd nondomination layer.
- Step 4.** The above step is repeated for each solution of 2nd non-domination layer to find the 3rd layer; thereafter, the procedure is repeated until all the population are assigned into different non-domination layers.

For each individual  $j$  in the 2nd or higher level of nondomination layer, the domination count  $Dom_j$  can be  $2NS-1$  at most. Therefore, each individual  $j$  will be visited  $2NS-1$  times at most before its  $Dom_j$  equals zero. At this point, the individual  $j$  is assigned in a nondomination layer and will never be visited again.

Therefore, if two individuals ( $i$  and  $j$ ) with different layers are compared, the individual with a better or lower layer is prioritized. But if both individuals belong to the same layer ( $i_{layer} = j_{layer}$ ), the ranking between them depends on CRD with higher rank given to solution which has higher CRD.

### 3.2. The CRD

The CRD is assigned to each individual  $i$  in the population with the goal of measuring the density of individuals surrounding it. Therefore, the  $CRD_i$  of individual  $i$  is the value of the average distance of its two neighboring individuals along each of the objectives. The boundary individuals which have the highest and lowest values of the objective function are given an infinite CRD so that they are always chosen. This process is done for each objective function. The final CRD of  $i$  individual is obtained by adding the entire individual CRD values in each objective function. Figure 3 shows the pseudocode of CRD computation with the procedure for calculating  $CRD_i$  for each individual  $i$  in the layer fare given in the following steps [75]:

- Step 1.** Determine the number of individuals in each layer as  $h = |Layer|$ .
- Step 2.** Assign  $CRD_i = 0$  for each individual  $i$  in the layer.
- Step 3.** Sort the individuals in the layer in the worst order of  $obj_k$  for each objective function  $k = 1, 2, \dots, K$ .
- Step 4.** Assign infinity CRD to the first and the last individuals in the sorted list, for each objective function ( $CRD_1 = CRD_h = \infty$ ).
- Step 5.** Assign CRD for the other individuals in the sorted list  $i = 2$  to  $(h-1)$  for each objective by

$$CRD_i = CRD_i + \frac{obj_k^{i+1} - obj_k^{i-1}}{obj_k^{\max} - obj_k^{\min}} \quad (3)$$

where  $obj_k^{\max}$  is The population-maximum value of the  $k$ th objective function and  $obj_k^{\min}$  is The minimum value of the  $k$ th objective function in the population.

The crowding-comparison operator ( $<_n$ ) is used for identifying the superior individual between two individuals under comparison, based on nondomination layer ( $Layer_i$ ) and  $CRD_i$  of each individual; that is,

$$i <_n j \text{ if } (i_{layer} < j_{layer}) \text{ or } ((i_{layer} = j_{layer}) \text{ and } (CRD_i > CRD_j)) \quad (4)$$

The crowding-comparison operator chooses the solutions from the least crowded region, which guarantees the diversity in the population.

### 4. SINE-COSINE ALGORITHM

SCA is a population-based algorithm which is based on sine and cosine mathematical functions for solving optimization problems. SCA initiates the search process with a random agent set, which is updated in each generation by using the mathematical expressions based on sine and cosine functions. The position of each individual is updated by the following equations [48]:

$$y_i^{q+1} = \begin{cases} y_i^q + d_1 \times \sin(d_2) \times |d_3 b_i^q - y_i^q| & d_4 < 0.5 \\ y_i^q + d_1 \times \cos(d_2) \times |d_3 b_i^q - y_i^q| & d_4 \geq 0.5 \end{cases} \quad (5)$$

where  $y_i^q$  is the position of current agent in  $i$ th dimension at  $q$ th iteration, while  $d_1$  is a deterministic parameter and  $d_2, d_3, d_4$  are the

```

Sorting the individuals into nondominated layers
For each  $j \in IN_i$  do
   $Dom_j = 0$ 
   $set_j = \{\}$ 
  For each  $q \in IN_i$  do
    If  $j \prec q$  then
       $set_j = set_j \cup q$ 
    else if  $q \prec j$  then
       $Dom_j = Dom_j + 1$ 
  End if
End for
If  $Dom_j = 0$  then
   $j_{layer} = 1$ 
   $layer_1 = layer_1 \cup j$ 
End If
End for
 $i=1$  %Initialize the layer counter
while  $layer_i \neq \{\}$  do
   $ST=\{\}$  %Used to store the individuals of the next layer
  For each  $j \in layer_i$  do
    For each  $q \in set_j$  do
       $Dom_q = Dom_q - 1$ 
      If  $Dom_q = 0$  then % $q$  belongs to the next layer
         $q_{layer} = i + 1$ 
         $ST = ST \cup q$ 
      End If
    End For
  End For
   $i=i+1$ 
   $layer_i = ST$ 
End For
End while
  
```

Figure 2 | The elitist nondominated sorting technique.

```

Procedure Crowding Distance (layer)
 $h = |Layer|$ 
For  $i=1:h$  do
   $CRD_i = 0$ 
End For
For  $k=1:K$  do
  Sort(layer,k)
   $CRD_1 = CRD_h = \infty$ 
  For  $i=2:(h-1)$  do
     $CRD_i = CRD_i + \frac{obj_k^{i+1} - obj_k^{i-1}}{obj_k^{max} - obj_k^{min}}$ 
  End For
End For
  
```

Figure 3 | The pseudocode of crowding distance computation.

evenly distributed random numbers which control the exploitation and exploration of the search space and  $b_i^q$  is the position of the best agent in the  $i$ th dimension after  $q$  iteration.

As shown in (5), the deterministic parameter  $d_1$  and stochastic parameters ( $d_2, d_3, d_4$ ) of SCA play main roles in the performance of SCA, where  $d_1, d_2, d_3, d_4$  are [76]:

- $d_1$  is the maximum magnitude of the individual step (individual movement) which could be either in the area between the solution and the best solution or outside it as shown in Figure 4 and assigned by

$$d_1 = c - c \left( \frac{q}{q_{max}} \right) \tag{6}$$

where  $c$  is constant,  $q$  is the current iteration, and  $q_{max}$  is the maximum number of iterations.

- $d_2$  is an evenly distributed random number where  $0 \leq d_2 \leq 2\pi$  which determines the margin of movement toward or outward the terminus point and also the direction of the movement.
- $d_3$  assigns weight randomly for the best solution for stochastically emphasizing ( $d_3 < 1$ ) or deemphasizing ( $d_3 > 1$ ) the effect of the best solution in defining the distance assigned by  $d_3 = a \times w_1$  where  $a$  is a constant greater than 1 and  $w_1$  is an evenly distributed random number, where  $0 \leq w_1 \leq 1$
- Finally,  $d_4$  is an evenly distributed random number where  $0 \leq d_4 \leq 1$  which equally switches between the cosine and sine functions.

$d_1, d_2, d_3, d_4$  are updated at each iteration for increasing the diversity of the solutions. By changing the sine and cosine functions ranges, the solutions can search outside the space between their corresponding destinations. Also, this leads to exploring the search space. Figure 4 describes how the above equation defines space between two individuals in the search space.

The general steps of SCA are implemented as follows:

```

Initialize random solutions set (search agents) inside the range of upper and lower limits of decision variables.
While ( $q < q_{max}$ )
Do
  Evaluate the objective functions of each solution.
  Identify the best individual according to the objective function ( $\mathbf{b} = \mathbf{y}^*$ ).
  Update the deterministic parameter  $d_1$  and stochastic parameters ( $d_2, d_3, d_4$ ) of SCA
  Update the individual position through. (5).
End
Save the best individual obtained so far as the optimal solution
  
```

## 5. THE NS-SCGA

This section introduces the NS-SCGA. NS-SCGA aims to integrate the merits of exploration capability of SCA and exploitation capability of NSGA-II to quicken the process of searching, avoid getting

stuck in local minima, and speed up the convergence to best solutions in a reasonable time. In NS-SCGA, the nondominated sorting and CRD techniques of NSGA-II divide the population into several ranks, which ensure solution diversity and Pareto dominance in NS-SCGA. Meanwhile, the population is evolved by the SCA mechanism along with the genetic operators, which integrates the merits of exploitation capability of genetic operators and the exploration capability of SCA. In NS-SCGA, the initial set of candidate solutions is generated and iteratively updated. Each new generation is produced by stochastically removing less desired solutions and introducing small random changes. The population of the solutions is subjected to selection based on CRD operator and crossover and mutation operators and updating formula of SCA. As a result, the population will gradually evolve toward an optimal solution. Therefore, the combination of the two well-known search strategies in the optimization field by this different technique aims to speed the process of searching, speed up the convergence to best solutions in a reasonable time, and avoid getting stuck in local minima, which can make a contribution in the field of multi-objective optimization. The basic steps of NS-SCGA are shown in Figure 5 and can be illustrated as follows:

**Step 1 (Initialization):**

Firstly, initialize the set of parameters which are the following:

- $N_{pop}$ : the size of population
- Crossover-probability  $p_c$
- Mutation-probability  $p_m$

Secondly, initialize the parent population ( $par_{pop}$ ) randomly in the feasible region.

**Step 2 (Evaluation):**

Once the population is initialized, the objective functions of each individual are evaluated.

**Step 3 (Generating Offspring Population):**

Divide  $par_{pop}$  into two equal subpopulations ( $pop_{sca}$  and  $pop_{ga}$ ); then update the individuals in subpopulation  $pop_{sca}$  by SCA by using (5), where the individuals in subpopulation  $pop_{ga}$  are evolved by crossover and mutation operators according to their probabilities. After completing the updating process of the two subpopulations, the subpopulations are added together to build the offspring population  $off_{pop} = pop_{sca} \cup pop_{ga}$ .

**Step 4 (Evaluation):**

Once the offspring population is generated, the objective functions of each individual are evaluated.

**Step 5 (Merge):**

Merge  $par_{pop}$  and  $off_{pop}$  to create an intermediate population  $IN_t = par_{pop} \cup off_{pop}$ .

**Step 6 (Sorting the Population-Based on Crowding-Comparison Operator):**

Compute  $Layer$  and  $CRD$  for each individual of  $IN_t$  population and according to them sort  $IN_t$  based on the crowding-comparison operator.

**Step 7 (Generate the New Population):**

Choose the new population ( $N_{pop}$ ) solutions from the sorted  $IN_t$  population by selecting solutions from non-dominated layers, from the best layer firstly, then from next best, and so on until  $N_{pop}$  solutions are chosen.

**Step 8 (Termination Condition):**

When the termination condition (maximum number of function evaluations [ $NFEs$ ] or maximum number of generations) has been satisfied, the algorithm is stopped.

If the termination condition is not satisfied, the selected  $N_{pop}$  solutions are assigned as  $par_{pop}$  and the previous steps are repeated from Step 3.

**Step 9 (The Optimal Pareto Front)**

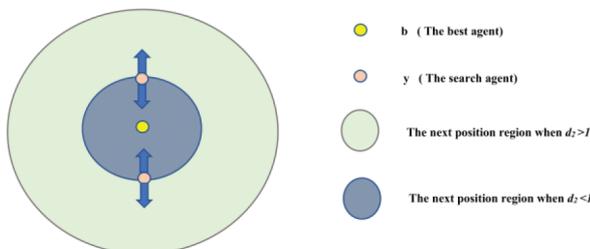
When the termination condition is satisfied the nondominated solutions in the first  $layer$  are selected and assigned as the optimal Pareto solution.

## 6. EXPERIMENTAL RESULTS

The performance of NS-SCGA is benchmarked by three main groups of test function which are as follows:

- Unconstrained multiobjective test functions: 10 unconstrained benchmark functions from the CEC09 test suite (UF1-UF10).
- Constrained multiobjective test functions: 10 constrained benchmark functions from the CEC09 test suite (CF1-CF10).
- Engineering application: The EEDP.

The NS-SCGA is coded in MATLAB 2018. The parameters of NS-SCGA used are given in Table 1. The NS-SCGA is implemented 30 times for each problem and the average results of the NS-SCGA are compared with the results of the other techniques available in the literature in terms of three commonly used performance metrics: inverted generational distance ( $IGD$ ), maximum spread ( $MS$ ), and spacing metric ( $SP$ ).



**Figure 4** | Effect of sine and cosine components in (5).

**Table 1** | The description of the parameters set.

$P_c$	<b>0.95</b>
Crossover operator	Single_point
$P_m$	0.09
Mutation operator	Real_value
NS-SCGA iterations	100–500

NS-SCGA, nondominated sorting sine-cosine genetic algorithm.

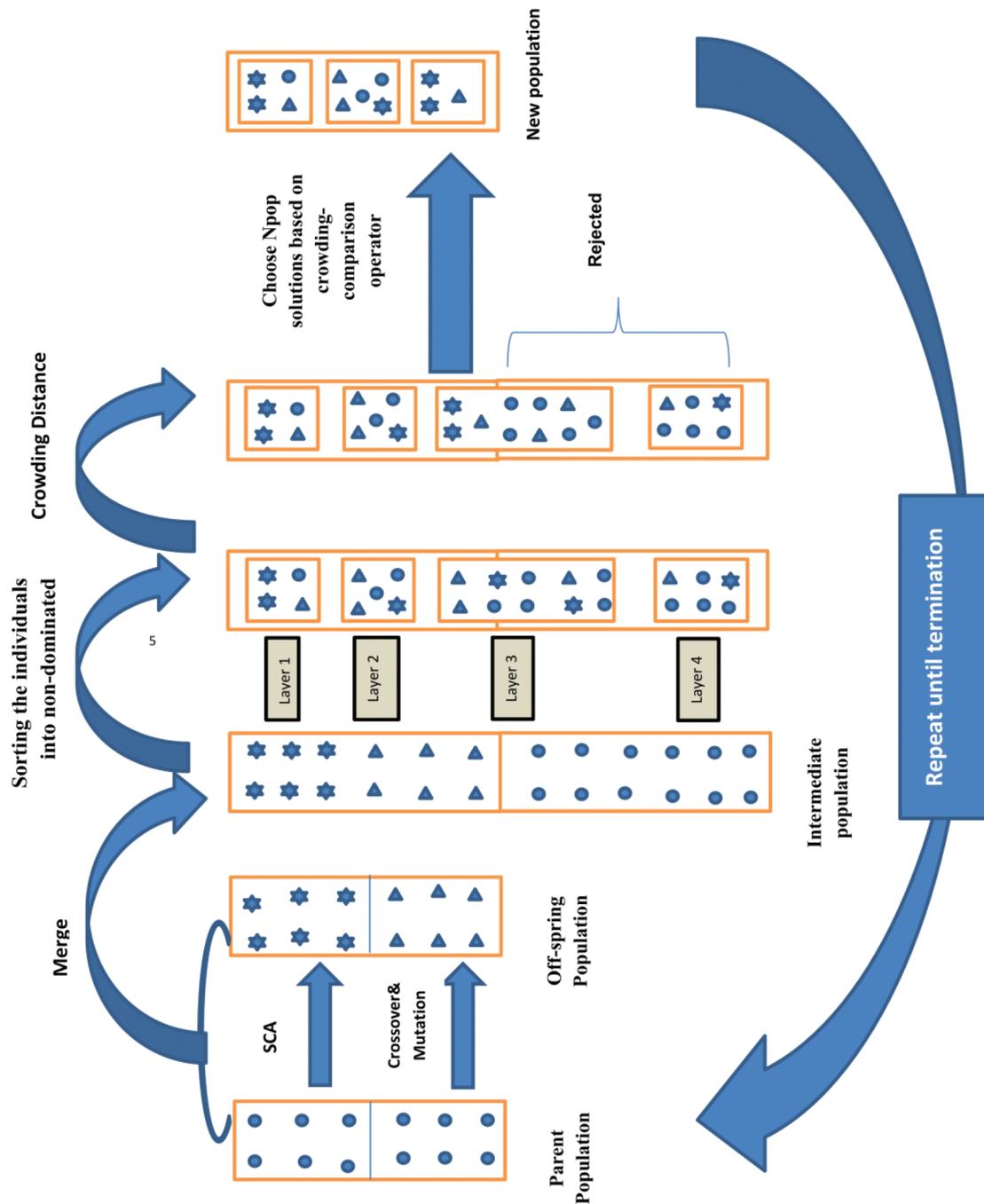


Figure 5 | The framework of nondominated sorting sine-cosine genetic algorithm (NS-SCGA).

### 6.1. Performance Metrics

In order to validate the quality of the Pareto-optimal solutions obtained by the NS-SCGA, both quantitative and qualitative metrics are employed, including *IGD*, *MS*, and *SP*. For quantifying the convergence, *IGD* is employed [1,77], where *SP* [78] and *MS* [79] have been employed to quantify the distribution of the obtained Pareto-optimal front (coverage).

- **Inverted Generational Distance (IGD):** IGD is used to quantify the convergence by calculating the distance of the obtained Pareto front (*OPF*) by the proposed algorithm from the true Pareto front (*TPF*). A small value of *IGD* denotes that

the *OPF* closely approaches the *TPF*. The mathematical equations of this performance metric are given by

$$IGD(OPF, TPF) = \frac{\sum_{z \in TPF} d(z, OPF)}{|TPF|}, \tag{7}$$

where *TPF* is the set of uniformly distributed points along true Pareto front and *OPF* is the set of the obtained Pareto-optimal solutions by the proposed algorithm. Furthermore  $d(z, OPF)$  refers to the minimum Euclidean distance between  $z$  and the point in *OPF*. Finally  $|TPF|$  is the number of the nondominated solutions in *TPF*

- **SP:** *SP* is used to measure the extent of the spread of the nondominated solutions throughout the Pareto front (i.e., it describes the distribution of nondominated solutions vectors). Through comparison with the true Pareto, coverage degree or distribution of nondominated solution vectors searched by algorithm could be

$$SP = \sqrt{\frac{1}{|OPF| - 1} \sum_{r=1}^{|OPF|} (\overline{dif} - dif_r)^2}$$

where

$$dif_r = \min_j \left( \sum_{v=1}^k |f_v^r - f_v^j| \right), \quad r, j = 1, 2, \dots, |OPF|, r \neq j$$

$$\overline{dif} = \frac{\sum_{r=1}^{|OPF|} dif_r}{|OPF|}$$
(8)

where  $|OPF|$  is the number of the obtained nondominated solutions, where *obj* refers to the number of objective functions.

The lower value of *SP* indicates a better uniform distribution in *OPF*. As the smaller value of *SP* indicates the better distribution of the current solution set of the algorithm as the zero value of *SP* indicates that all the Pareto points on the front are equidistant to each other and the Pareto points are evenly distributed on the front

- **MS:** *MS* is used to measure the maximum extension covered by the nondominated solution in the Pareto front. The mathematical equation of this performance metric is

$$MS = \sqrt{\sum_{j=1}^K \max_{z_j^{fmax}, z_j^{fmin} \in OPF} \left( d \left( z_j^{fmax}, z_j^{fmin} \right) \right)}$$
(9)

where  $z_j^{fmin}$  is the solution that has the minimum value in the *j*th objective and  $z_j^{fmax}$  is the solution that has the maximum value in the *j*th objective, while  $d \left( z_j^{fmax}, z_j^{fmin} \right)$  denotes the Euclidean distance.

A higher value of *MS* indicates that the nondominated solutions covered a large area of Pareto front.

## 6.2. Results of Unconstrained (CEC09) Benchmark Functions

To investigate the NS-SCGA performance, unconstrained test suites CEC09 have been employed. Among all the ten unconstrained benchmark problems, UF1 to UF7 are biobjective and UF8 to UF10 are triobjective optimization problems. The mathematical equations of these test functions are given in [80].

The maximum *NFEs* is set as 300000. The proposed NS-SCGA is compared with another 11 newly developed algorithms for MOPs in terms of the average value of *IGD* obtained in 30 independent runs. These algorithms are MO-TLBO [21], MOEA/D [14], MTS [20], MOEA/D with guided mutation and priority update

(MOEADGM) [81], dynamical MOEA with decomposition technique (DMOEADD) [82], multiobjective artificial bee colony algorithm (MOABC) [83], Liu Li algorithm [15], hybrid NSGA-II with adaptive operators selection (HNSGA) [84], a new decomposition-based MOEA with the  $\epsilon$ -constraint framework (DMOEAE- $\epsilon$ C) [85], adaptive cross-generation differential evolution (ACGDE-NSGA-II) [86], and MO-AAA [1]. The computational results of UF1 to UF10 benchmark functions are given in Table 2. Moreover, NS-SCGA is compared with another three recent algorithms, which solved unconstrained CEC09 in terms of the average value of *SP* and *MS* obtained in 30 independent runs in Table 3. These algorithms include enhanced multiobjective particle swarm optimization (MOPSO+EPD) [87], MOGWO [27], and multiobjective grasshopper optimization algorithm (MOGOA) [88]. The best Pareto-optimal fronts obtained by the NS-SCGA are given in Figure 6.

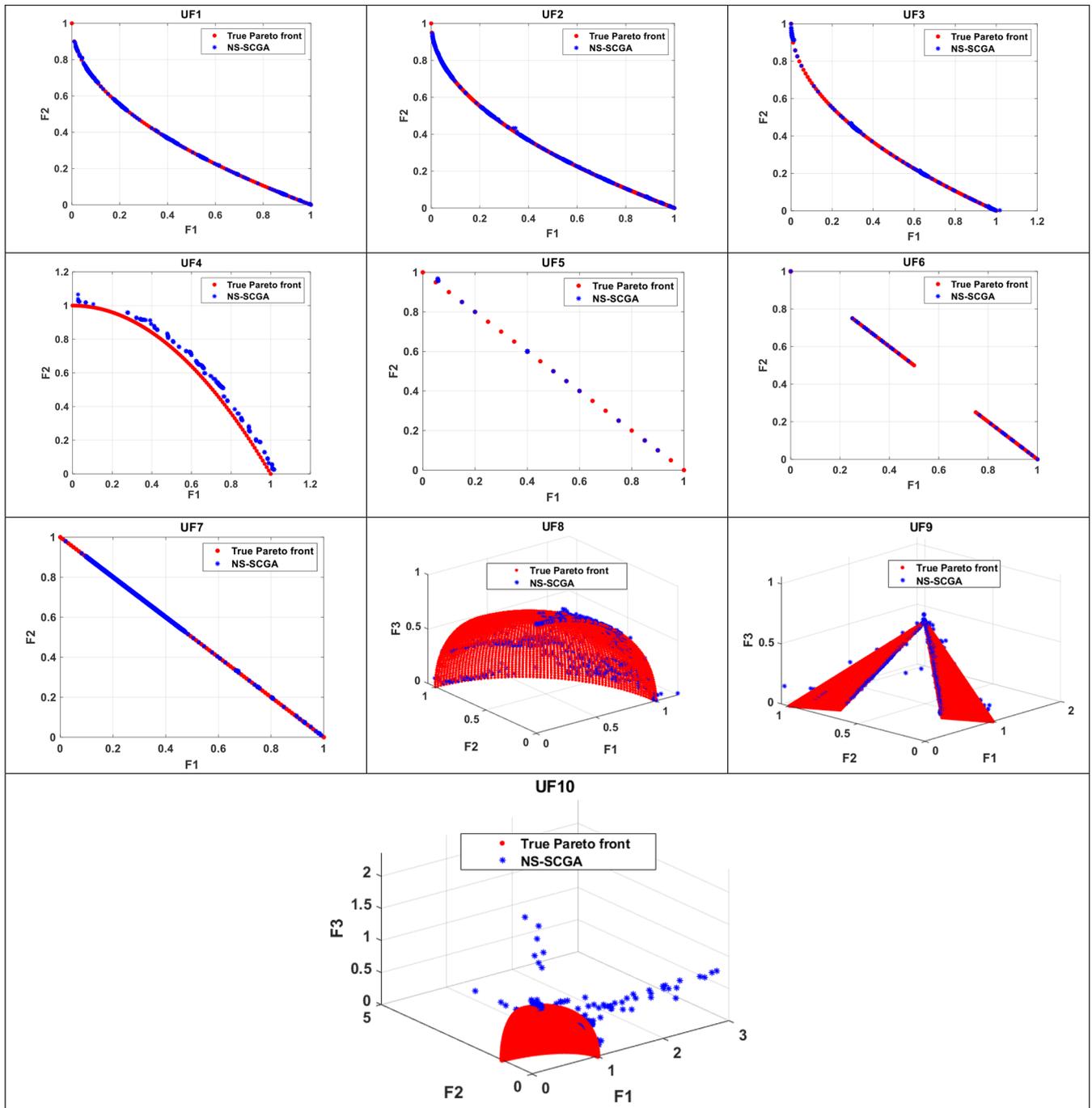
It can be noticed that three other methods are used in the comparison between *SP* and *MS* values in Table 3, as *IGD* calculation has two formulas in the literature (the one which was used in formula 7 and the other one can be found in [27,87]).

The 11 techniques used in the comparison in Table 2 use the same formula which is used in calculating the *IGD* metric in this paper, while those 11 techniques did not calculate *SP* and *MS* metric. That is why we used the only three methods that were found in the literature which calculate *SP* and *MS* metrics for these benchmark functions in the comparison of the results of *SP* and *MS* metric. (Meanwhile, they did not use the same formula of *IGD* metric which is used in our work.)

Based on Table 2, the NS-SCGA showed improved average results in comparison to all other techniques across all test suites except for problems UF3, UF8, and UF10.

As shown in Table 2 in UF2, UF5, UF6, UF7, and UF9 function, the average value of *IGD* produced by NS-SCGA is better than the other techniques. Therefore, our algorithm outperforms the other techniques in these functions. As shown in Figure 6, the obtained Pareto fronts of these functions by the NS-SCGA show a good convergence and suitable distribution over the Pareto front.

Meanwhile, the NS-SCGA obtains a competitive result for UF1 and UF4 functions. Only MO-AAA obtains results comparable to the NS-SCGA, but the latter still produces slightly better solutions. The obtained Pareto front for UF1 and UF4 test functions represents good distribution and appropriate proximity in the objective space. But, for the UF3 function, the NS-SCGA is placed at the third rank in solving it. Only DMOEA- $\epsilon$ c and MOEA/D algorithms give a good result compared to NS-SCGA. As shown in Figure 6, the obtained nondominated solutions of UF3 are uniformly distributed along the Pareto front. The UF8 is the first triobjectives benchmark function solved in this work. The NS-SCGA is ranked the eighth in solving UF8 among the competing algorithms. The MO-AAA, MOEA/D, DMOEADD, MOABC, Liu Li, HNSGA, and DMOEA-Ec algorithms obtained the best result for this benchmark function. However, the obtained Pareto front of UF8 shows that NS-SCGA obtained a set of solutions distributed appropriately in the three-dimensional objective space. Also, in the case of the UF10 benchmark function, MO-TLBO, MTS, MOABC, and DMOEA-c produced competitive performance. The proposed NS-SCGA obtained the



**Figure 6** | The obtained Pareto fronts by the nondominated sorting sine-cosine genetic algorithm (NS-SCGA) for CEC09 unconstrained benchmark functions.

fifth rank in solving this problem. Figure 6 shows that most of the portions of the Pareto front of UF10 are not covered successfully by NS-SCGA. Thus, the *IGD* value is a large value in the UF10 test problem.

This shows that, overall, the NS-SCGA has good convergence for unconstrained MOPs and rivals other techniques that we compared it against in these experiments. On the other hand, Figure 6 proves that the obtained Pareto-optimal solutions by the NS-SCGA tend to be closer to the true Pareto-optimal front. That is, the proposed NS-SCGA is capable of approximating the true Pareto front.

Note that *MS* with high values and *SP* with low values indicate that the obtained Pareto-optimal fronts have a more uniform distribution. It can be observed from Table 3 that our algorithm is superior in almost half of the benchmark problems. These results prove that the coverage of the obtained solutions by the NS-SCGA is of high distribution across all objectives of the benchmark functions. This can be observed as well in Figure 6 where the obtained solutions by the NS-SCGA are well-distributed on the majority of the benchmark functions.

**Table 2** IGD values for different CEC09 unconstrained test functions obtained by nondominated sorting sine-cosine genetic algorithm (NS-SCGA) and other techniques in 30 independent runs.

	MO-AAA	MO-TLBO	MOEA/D	MTS	DMOEADD	MOABC	Liu Li Algorithm	MOEADGM	HNSGA	DMOEAE $\epsilon$ C	ACGDE-NSGA-II	NS-SCGA
UF1	0.003910	0.008790	0.004350	0.006460	0.01040	0.006180	0.007850	0.006200	0.01203	0.004407	0.05280	0.003840
UF2	0.006720	0.007840	0.006790	0.006150	0.006790	0.004840	0.01230	0.006400	0.003897	0.006530	0.02050	0.003700
UF3	0.07620	0.06930	0.007420	0.05310	0.03340	0.05120	0.01500	0.04290	0.02875	0.007209	0.09470	0.01290
UF4	0.03560	0.07010	0.06390	0.02630	0.04270	0.05800	0.04350	0.04760	0.04121	0.06017	0.04100	0.02540
UF5	0.04130	0.09790	0.1810	0.04490	0.3150	0.07780	0.1620	1.790	0.3792	0.3845	0.2870	0.04122
UF6	0.04820	0.05530	0.005870	0.05920	0.06670	0.06540	0.1760	0.5560	0.1508	0.2662	0.1576	0.004801
UF7	0.006420	0.06800	0.004440	0.04080	0.01030	0.05570	0.007300	0.007600	0.009700	0.004079	0.02620	0.004022
UF8	0.05980	0.09930	0.05840	0.1130	0.06840	0.06730	0.08240	0.2450	0.08169	0.05284	0.1383	0.09634
UF9	0.2000	0.1640	0.07900	0.1140	0.04900	0.06150	0.09390	0.1880	0.1120	0.04290	0.1776	0.03990
UF10	0.5120	0.1790	0.4740	0.1530	0.3220	0.1950	0.4470	0.5650	0.3165	0.02240	0.6440	0.2873

**Table 3** SP and MS values for different CEC09 unconstrained test functions obtained by nondominated sorting sine-cosine genetic algorithm (NS-SCGA) and other techniques in 30 independent runs.

	MOGOA		MOGWO		MOPSO+EPD		NS-SCGA	
	MS	SP	MS	SP	MS	SP	MS	SP
UF1	0.7270	0.001200	0.9268	0.01237	1.263	0.008000	1.401	0.001000
UF2	0.8845	0.0007000	0.9097	0.01108	0.8640	0.008000	1.403	0.0009531
UF3	0.6051	0.001900	0.9498	0.04590	1.406	0.007000	1.412	0.0005095
UF4	0.9050	0.0001000	0.9424	0.009690	1.025	0.007000	1.232	0.0007899
UF5	0.2379	0.0007000	0.3950	0.1523	1.275	0.005000	1.525	0.001300
UF6	0.2525	0.0003000	0.6736	0.01446	0.6060	0.007000	1.031	0.007500
UF7	0.8460	0.0001000	0.8013	0.008240	0.7860	0.007000	1.361	0.00009000
UF8	0.4417	0.01750	0.4457	0.006870	4.065	0.02600	1.731	0.01250
UF9	0.1938	0.01390	0.8399	0.01730	3.838	0.02300	1.724	0.01290
UF10	0.3233	0.006700	0.2972	0.02523	2.913	0.01900	1.725	0.02990

As per the results of Tables 2 and 3 and Figure 6, it is proved that NS-SCGA is able to outperform other techniques in the majority of unconstrained benchmark problems and the obtained Pareto-optimal fronts are well spread across all objectives.

### 6.3. Results of Constrained (CEC09) Benchmark Functions

The performance of NS-SCGA in solving the CMOPs is evaluated by investigating constrained CEC09 test instances. Among the constrained CEC09 benchmark problems, CF1–CF7 are biobjective where CF8, CF9, and CF10 are triobjective problems. The mathematical equations of these test instances are available in [80]. The *NFEs* is set as 300000. The NS-SCGA is compared with eight other MO algorithms for CMOPs in terms of the average IGD value estimated in 30 independent runs. These algorithms include MTS [20], MOEADGM [81], Liu Li algorithm [15], differential evolution with self-adaptation and local search for constraint multiobjective optimization algorithm (DECMOSA-SQP) [89], NSGA-II with local search (NSGAIIS) [47], generalized differential evolution (GDE3), and MO-AAA [1]. The computational results of CF1 to CF10 benchmark functions are shown in Table 4. All the results except for the NS-SCGA were taken from [1] for the comparison purpose. The best obtained Pareto-optimal fronts by the NS-SCGA are given in Figure 7.

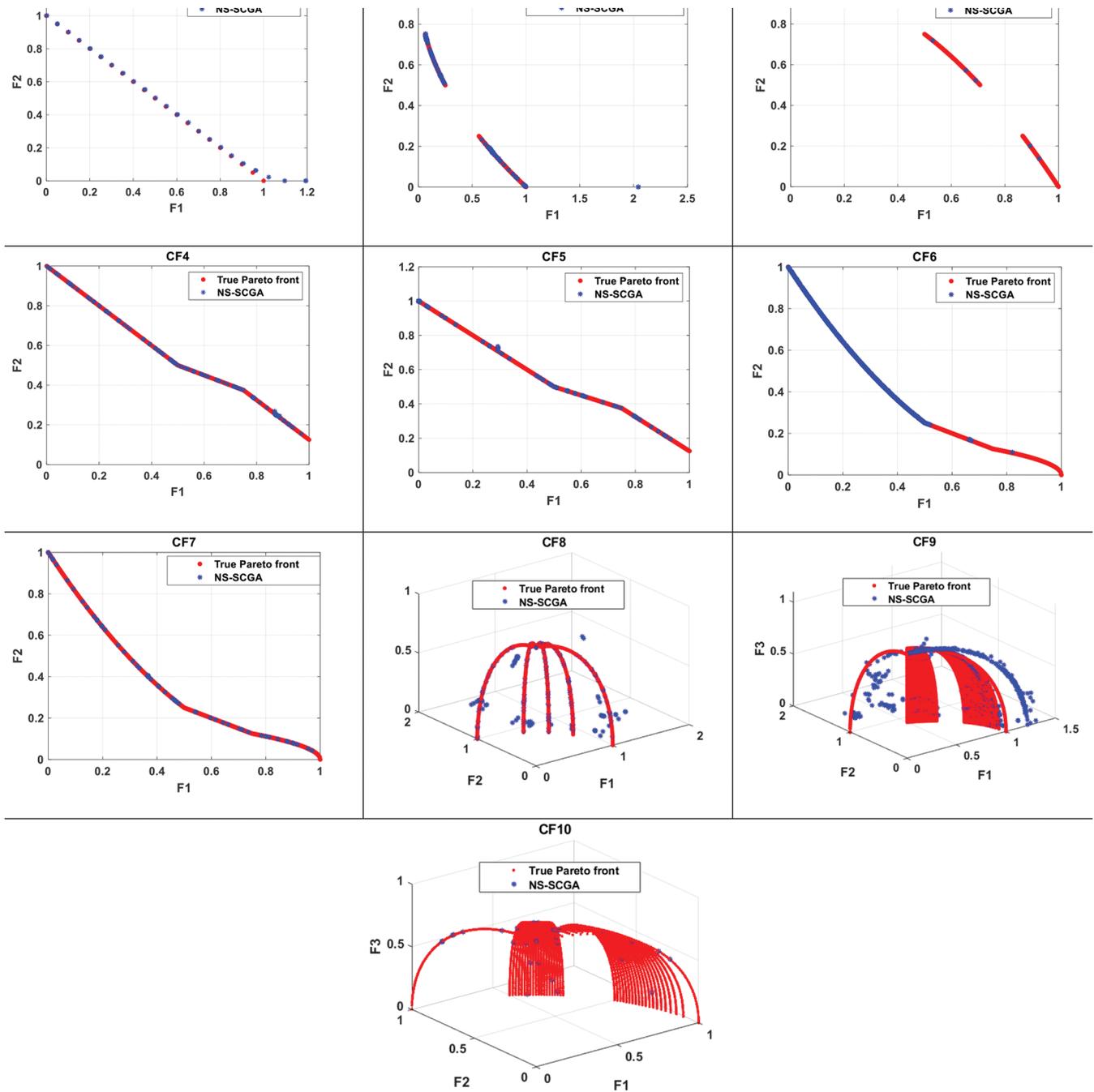
As shown in Table 4, NS-SCGA obtains the best IGD values for all the constrained test functions, except in CF1, CF4, CF6, and CF10, for which MO-AAA and Liu Li algorithm obtain the best results for CF4&CF10 and CF1&CF6, respectively. However, the other compared algorithms do not perform as well as NS-SCGA, MO-AAA, and Liu Li algorithm. It is worth stressing that NS-SCGA outperforms MO-TLBO, MTS, NSGAIIS, and MOEADGM, which

are mainly designed for MOPs; therefore, their performance is not evaluated on *CF* test functions.

The approximated fronts obtained by NS-SCGA for each *CF* test functions are shown in Figure 7 with the true fronts. The obtained Pareto fronts by the NS-SCGA algorithm in the CF1, CF4, CF6, and CF10 test functions are well-converged and well-distributed also. Figure 7 proves that NS-SCGA is capable of converging to a certain degree to the true Pareto front. The proposed NS-SCGA performs well in CF8 and CF9 which are triobjective test functions. The Pareto-optimal solutions obtained by NS-SCGA cover almost the whole true Pareto fronts for the triobjective benchmark functions. The above results indicate that NS-SCGA is effective in solving CMOPs.

### 6.4. The LO for Unconstrained and Constrained Benchmark Problem

The *LO* [75] is used to assess the overall performance of the NS-SCGA between the 9 compared techniques in solving the constrained test problems and the 12 compared techniques in solving the unconstrained test functions. *LO* defines the overall rank of the compared techniques. For any test instance, the algorithm which occupies the 1st rank is the one which obtains the best value of average IGD for that function as compared to the rest of the techniques, and the next better performing algorithm obtains the 2nd rank, and so on. In this section, the ranks are assigned to the compared techniques for each constrained and unconstrained test instance. Then, the mean ranking is calculated for each algorithm for the constrained test instances as well as unconstrained test instances. The algorithm which has minimum mean rank is defined as the best algorithm and occupies the 1st rank in *LO*. Similarly, the next better



**Figure 7** | The Pareto front obtained by the nondominated sorting sine-cosine genetic algorithm (NS-SCGA) for CEC09 constrained multiobjective optimization problems (CMOPs).

**Table 4** | IGD values for different CEC09 constrained multiobjective optimization problems (CMOPs) obtained by nondominated sorting sine-cosine genetic algorithm (NS-SCGA) and other techniques in 30 independent runs.

	MO-AAA	Liu Li Algorithm	DECMOSA-SQP	MTS	NSGAILS	GDE3	MO-TLBO	MOEADGM	NS-SCGA
Cf1	0.06820	0.0008500	0.1080	0.01920	0.006920	0.02940	0.02490	0.01080	0.006000
Cf2	0.003920	0.004200	0.09460	0.02680	0.01180	0.01600	0.01200	0.008000	0.003851
CF3	0.04250	0.1830	1.000	0.1040	0.2400	0.1280	0.09800	0.5130	0.04170
CF4	0.002810	0.01420	0.1530	0.01110	0.01580	0.007990	0.008680	0.07070	0.007100
CF5	0.06230	0.1100	0.4130	0.02080	0.1840	0.06800	0.07640	0.5450	0.01579
CF6	0.04010	0.01390	0.1480	0.01600	0.02010	0.06200	0.0640	0.2070	0.03003
CF7	0.02880	0.1040	0.2600	0.02470	0.2330	0.04170	0.03330	0.5360	0.01120
CF8	0.03110	0.06070	0.1760	1.090	0.1110	0.1390	0.2980	0.4060	0.02290
CF9	0.07620	0.05050	0.1270	0.08510	0.1060	0.1150	0.08100	0.1520	0.05041
CF10	0.09990	0.1970	0.5070	0.1380	0.3590	0.4920	0.3010	0.3140	0.1034

**Table 5** | The lexicographic ordering (LO) for the constrained multiobjective optimization problems (NS-SCGA) and the other compared algorithms which solved unconstrained and constrained test instances for the CEC09.

Constrained Function Algorithm	Mean Ranking	Rank According to LO	Unconstrained Function Algorithm	Mean Ranking	Rank According to LO
NS-SCGA	1.600	1.000	NS-SCGA	2.300	1.000
MO-AAA	3.000	2.000	DMOEA	4.900	2.000
Liu Li algorithm	3.800	3.000	MOEA/D	5.200	3.000
MTS	4.500	4.000	MO-AAA	5.770	4.000
MO-TLBO	5.100	5.000	MOABC	5.800	5.000
NSGAIILS	5.600	6.000	MTS	6.500	6.000
GDE3	5.720	7.000	Liu Li algorithm	6.800	7.000
MOEADGM	7.500	8.000	DMOEADD	7.000	8.000
DECMOSA-SQP	8.200	9.000	MO-TLBO	8.200	9.000
			HNSGA	8.500	10.00
			MOEADGM	9.000	11.00
			ACGDE-NSGA-II	9.900	12.00

mean rank is determined and the algorithm which obtains that rank is assigned in the 2nd order in the LO and so on. The mean ranking and ranking based on LO of the considered techniques which solved the constrained and unconstrained test functions are given in Table 5. It can be observed from Table 5 that the NS-SCGA obtains the minimum mean ranking and occupies the 1st rank among the 9 algorithms and the 12 algorithms in solving the constrained and unconstrained test instances, respectively.

## 6.5. Statistical Analysis

In this section, Friedman test [90] is utilized to analyze the results of IGD metric acquired from using various algorithms in case of unconstrained and constrained benchmark functions. Furthermore, Wilcoxon signed-rank test [91] is performed to illustrate the significant differences between the NS-SCGA and the other algorithms in each case.

### • Friedman test

The nonparametric Friedman test is used with a significance level of 0.05 to statistically assess the IGD's experimental results. The Friedman statistical test is performed to demonstrate whether the differences in performance between the NS-SCGA and the compared algorithms are significant or not in solving unconstrained and constrained benchmark functions.

Friedman test ranks the algorithms for each benchmark function separately, and the best performing algorithm obtains rank 1, the second-best rank 2, and so forth. In case of ties, it assigns average rank. Then, the Friedman test compares the average rank for the algorithms and obtains the Friedman statistics. In the Friedman test, the smaller the ranking, the better the performance of the algorithm. One more important term in the Friedman test is the  $p$ -value. A  $p$ -value gives an indicator whether there is a significant difference between algorithms or not, where the smaller the  $p$ -value, the stronger the evidence of a significant difference [90].

Figures 8 and 9 show Friedman's average rankings of the compared algorithms on the unconstrained and constrained benchmark problems.

In Friedman's test statistics in Figures 8 and 9, there are three values: Chi-Square, df, and Asymp. Sig., where the Chi-Square basically summarizes how differently the algorithms were rated in a single number, while df is the degrees of freedom associated with our test statistic, and finally Asymp. Sig. is the approximate  $p$ -value.

As shown in Figures 8 and 9, we can see that the  $p$ -value (Asymp. Sig.) is less than the level of significance ( $\text{Alpha} = 5\%$ ) in the two cases (unconstrained and constrained problems). This denotes that the algorithms used in the comparison differ in performance. Furthermore, the performance of NS-SCGA is better than the competitors as it has the lowest ranked mean.

According to the Friedman test, it can be concluded that NS-SCGA is an efficient and competitive MO algorithm.

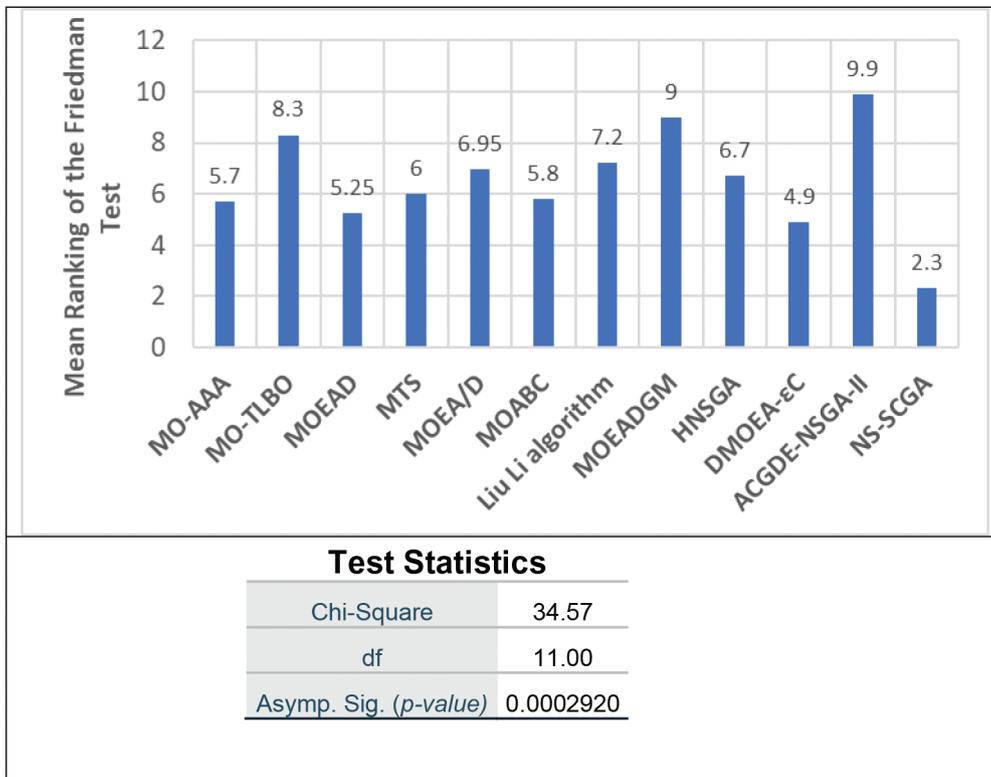
### • The Wilcoxon signed-rank test

The Wilcoxon signed-rank test was also performed to detect significant differences between the behaviors of the algorithms' pair. Wilcoxon test provides the  $p$ -value, that is, the probability that the difference in the performance achieved by the two algorithms is obtained by chance and is not statistically significant [91].

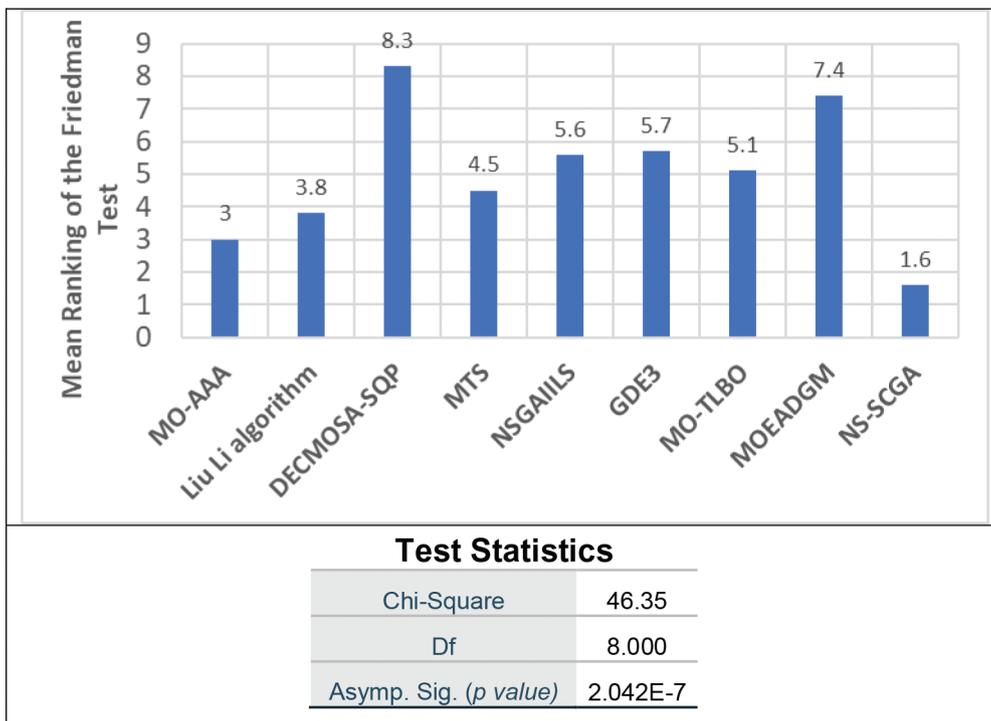
In Tables 6 and 7, NS-SSGA is compared with the other algorithms in terms of IGD in constrained and unconstrained benchmark function, respectively, by applying the Wilcoxon signed-rank test to determine whether NS-SCGA is statistically better than other algorithms or not.

As shown in Table 6, the Wilcoxon test results indicate that the NS-SCGA had a better statistically significant performance compared to MO-AAA, DMOEADD, Liu li algorithm, MOEADGM, and ACGDE-NSGAI as  $p$ -value is smaller than the level of significance ( $\text{Alpha} = 5\%$ ). But there is no statistically significant difference between the NS-SCGA on one hand and MO-TLBO, MOEA/D, MTS, MOABC, and DMOEA- $\epsilon$ C on the other hand as  $p$ -value is greater than the level of significance (5%).

On the other hand, all  $p$ -values of the Wilcoxon test results in Table 7 are less than 5%. Therefore, there are statistically significant differences between NS-SCGA and the other algorithms. NS-SCGA was significantly better than the other algorithms in solving constrained benchmark functions.



**Figure 8** | Friedman test on nondominated sorting sine-cosine genetic algorithm (NS-SCGA) and its 11 competitors on all the unconstrained CEC09 benchmark problems in terms of inverted generational distance (IGD).



**Figure 9** | Friedman test on nondominated sorting sine-cosine genetic algorithm (NS-SCGA) and its 8 competitors on all the constrained CEC09 benchmark problems in terms of inverted generational distance (IGD).

**Table 6** | Nondominated sorting sine-cosine genetic algorithm (NS-SCGA) pairwise comparisons: Wilcoxon test summary for the CEC09 unconstrained benchmark functions.

		N	Mean Rank	Sum of Ranks	p-value
NS-SCGA vs MO-AAA	Negative ranks	9.000	5.440	49.00	0.02800
	Positive ranks	1.000	6.000	6.000	
NS-SCGA vs MO-TLBO	Negative ranks	9.000	5.110	46.00	0.05900
	Positive ranks	1.000	9.000	9.000	
NS-SCGA vs MOEA/D	Negative ranks	8.000	5.500	44.00	0.09300
	Positive ranks	2.000	5.500	11.00	
NS-SCGA vs MTS	Negative ranks	9.000	5.000	45.00	0.07400
	Positive ranks	1.000	10.00	10.00	
NS-SCGA vs DMOEADD	Negative ranks	9.000	5.330	48.00	0.03700
	Positive ranks	1.000	7.000	7.000	
NS-SCGA vs MOABC	Negative ranks	8.000	5.130	41.00	0.1640
	Positive ranks	2.000	7.000	14.00	
NS-SCGA vs Liu Li algorithm	Negative ranks	9.000	5.560	50.00	0.02200
	Positive ranks	1.000	5.000	5.000	
NS-SCGA vs MOEADGM	Negative ranks	10.00	5.500	55.00	0.005000
	Positive ranks	0.000	.000	.00	
NS-SCGA vs HNSGA	Negative ranks	9.000	5.670	51.00	0.01700
	Positive ranks	1.000	4.000	4.000	
NS-SCGA vs DMOEA-εC	Negative ranks	7.000	4.860	34.00	0.5080
	Positive ranks	3.000	7.000	21.00	
NS-SCGA vs ACGDE-NSGAI	Negative ranks	10.00	5.500	55.00	0.005000
	Positive ranks	0.00	.00	.00	

**Table 7** | Nondominated sorting sine-cosine genetic algorithm (NS-SCGA) pairwise comparisons: Wilcoxon test summary for the CEC09 constrained benchmark functions.

		N	Mean Rank	Sum of Ranks	p-value
NS-SCGA vs MOAAA	Negative ranks	8.000	6.000	48.00	0.03700
	Positive ranks	2.000	3.500	7.000	
NS-SCGA vs Liu Li algorithm	Negative ranks	8.000	5.880	47.00	0.04700
	Positive ranks	2.000	4.000	8.000	
NS-SCGA vs DECMOSA-SQP	Negative ranks	10.00	5.500	55.00	0.005000
	Positive ranks	0.00	0.000	0.000	
NS-SCGA vs MTS	Negative ranks	9.000	5.560	50.00	0.02200
	Positive ranks	1.000	5.000	5.000	
NS-SCGA vs NSGAI	Negative ranks	9.000	5.670	51.00	0.01700
	Positive ranks	1.000	4.000	4.000	
NS-SCGA vs GDE3	Negative ranks	10.00	5.500	55.00	.005000
	Positive ranks	0.000	0.000	0.000	
NS-SCGA vs MO-TLBO	Negative ranks	10.00	5.500	55.00	0.005000
	Positive ranks	0.00	.0000	0.00	
NS-SCGA vs MOEADGM	Negative ranks	10.00	5.500	55.00	0.005000
	Positive ranks	0	0.000	0.00	

In view of the results of Wilcoxon signed-rank test, it can be concluded that the NS-SCGA is superior to most of the compared algorithms in solving MOPs.

## 6.6. The EEDP: Case Study

The main objective of EEDP is the minimization of fuel cost and emission of atmospheric pollutants which are two conflicting objective functions while satisfying inequality and equality constraints foist by the power system requirements. The objectives and constraints of EEDP are given below [38]:

- **Objective**

The total fuel cost is formulated as a quadratic function in terms of real power output as follows:

$$f_c = \sum_{g=1}^{ng} \alpha_g + \beta_g p_{g_g} + \delta_g p_{g_g}^2 \quad (10)$$

and the total emission of atmospheric pollutants from fossil-fueled thermal stations in ton/h can be represented by

$$te_t = \sum_{g=1}^{ng} 10^{-2} \left( \varepsilon_g + \eta_g p_{g_g} + \lambda_g p_{g_g}^2 \right) + \mu_g \exp \left( \sigma_g p_{g_g} \right) \quad (11)$$

where  $\alpha_g, \beta_g, \delta_g$  are the coefficients of the  $g$ th generator fuel cost and  $\varepsilon_g, \eta_g, \lambda_g, \mu_g, \sigma_g$  are the emission coefficients of the  $g$ th generating unit, while  $p_{g_g}$  is the real output power of the  $g$ th generating unit, and  $ng$  refers to the total number of generating units.

- **Constraints**

I. **Power balance constraint** The total generated power must cover the power demand  $P_{Demand}$  and the losses of the transmission line  $P_{losses}$ ; that is,

$$\sum_{g=1}^{ng} p_{g_g} = P_{Demand} + P_{losses}, \quad (12)$$

where (13)

$$P_{losses} = \sum_{i=1}^{ng} \sum_{j=1}^{ng} pg_i B_{ij} pg_j + \sum_{i=1}^{ng} pg_i B_{0i} + B_{00}.$$

- II. **Generation capacity constraints** Each generator is restricted to generate output real power within the range of the maximum and minimum output power; that is,

$$PG_g^{\min} \leq pg_g \leq PG_g^{\max} \quad (14)$$

where  $P_{Demand}$  refers to the forecasted power demand and  $p_{losses}$  refers to the losses of the network transmission lines.  $B_{ij}$ ,  $B_{0i}$ ,  $B_{00}$  are the coefficient of the transmission network power loss and  $PG_g^{\max}$  and  $PG_g^{\min}$  represent the maximum and minimum output power of  $g^{\text{th}}$  generating unit, respectively.

NS-SCGA has been tested on IEEE 30-bus 6-generator test system for power demand  $P_{Demand} = 2.834$  per unit (p.u.) while taking transmission line losses into consideration, in order to validate the NS-SCGA performance in solving the EEDP. A per-unit system (p.u) in the power systems analysis, is the expression of system quantities as fractions of a defined base unit quantity. Calculations are simplified because quantities expressed as per-unit do not change when they are referred from one side of a transformer to the other. For more clarification, in per unit notation, the physical quantity is expressed as a fraction of the reference value, that is, per unit value equal to the actual value divided by the base value in the same unit.

The IEEE 30-bus 6-generator test system has 21 load buses and 41 transmission lines [92].

The parameters of that test system including fuel cost coefficients, emission coefficients, and the B-coefficients are taken from [93]. The parameter settings which are used for optimizing the EEDP are the following: the size of population  $N_{pop} = 200$  individuals, the maximum number of generations  $q_{\max} = 100$  and  $NFEs = 20,000$ , the crossover rate  $Pc = 0.97$ , and mutation rate  $P_m = 0.09$ , respectively. The NS-SCGA was run 30 independent runs and it is compared with NPGA [9], a hybrid MO algorithm based on particle swarm optimization and differential evolution (MO-DE/PSO) [31], MOPSO with time variant inertia and acceleration coefficients (TV-MOPSO) [94], SPEA [11], multiobjective harmony search (MOHS) algorithm [95], NSGA-II, hybrid evolutionary algorithm (NSGA-II/EDA) [38], and Multiobjective Collective Decision Optimization Algorithm (MOCDOA) [96] in terms of their best solutions for both minimum emissions and minimum fuel cost (the extreme points on the Pareto front). Moreover, the comparison is also done in terms of the SP metrics.

Figure 10 gives the obtained Pareto-optimal front by NS-SCGA during the best optimization run implementation. This figure shows the tradeoff between fuel cost and emissions. This figure proves that NS-SCGA is capable of obtaining the Pareto-optimal front with good diversity and convergence characteristics.

The schedules of the generation that give the minimum fuel cost (along with the corresponding emission) obtained by NS-SCGA and the other techniques are presented in Table 8. Similarly, Table 9

gives the schedules of the generation that yield to the minimum emissions. Since the power demand is measured in “p.u.,” the unit of output powers obtained by the proposed algorithm is measured by p.u. Moreover, Table 10 shows the results of the comparison between the three techniques in terms of SP metric.

It is quite evident from Table 8 that the NS-SCGA obtains the minimum fuel cost which is equal to 605.4 \$/h (United States dollar/hour) using 2.000E+4 NFEs. Furthermore, the performance of NS-SCGA is better compared to NPGA, MO-DE/PSO, TV-MOPSO, SPEA, MOHS, NSGA-II, NSGA-II/EDA, and MOCDOA in terms of fuel cost. Also, NS-SCGA outperforms NPGA, TV-MOPSO, SPEA, MOHS, NSGA-II, and NSGA-II/EDA in terms of NFEs. MO-DE/PSO returned the lowest value of NFEs which is 1.000E+4 but it obtained fuel cost equal to 606.0 which is more than that obtained by MOCDOA and by the NS-SCGA.

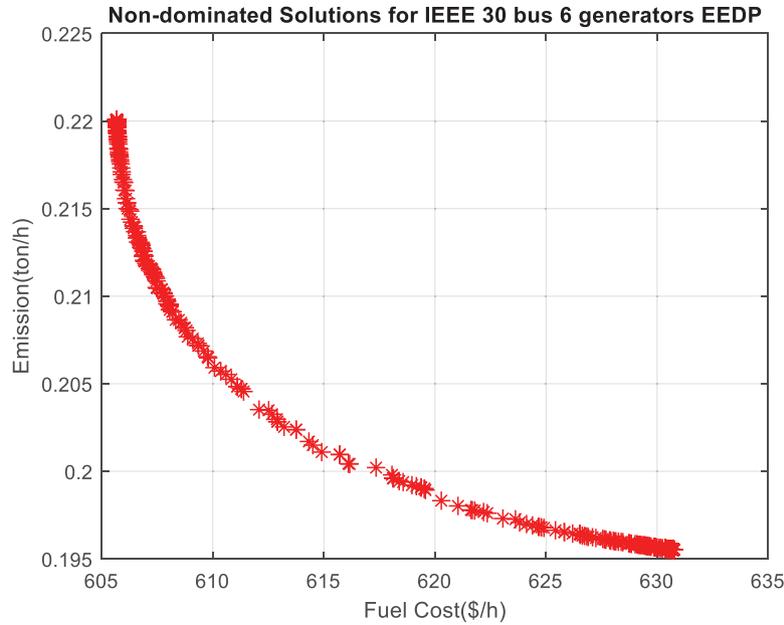
In Table 9, the minimum emission obtained by NS-SCGA is equal to 0.1941 which is almost the same as that obtained by MO-DE/PSO, NSGA-II, and MOCDOA. But the fuel cost corresponding to the minimum emission obtained by NS-SCGA is lower than other algorithms.

In addition, from Table 10 it is quite evident that the performance of NS-NSGA is far better than the other compared algorithms, where the distribution of TV-MOPSO and MO-DE/PSO is not as good as that of NS-SCGA. Moreover, the obtained Pareto front by NS-SCGA shown in Figure 10 further demonstrates the excellent solutions distribution of the obtained Pareto front. As seen from the above results, NS-SCGA is able to obtain high-quality results when compared with the other techniques in solving EEDP of IEEE 30-bus 6-generator system.

### 6.6.1. Identifying the best compromise solutions from the obtained Pareto front by the TOPSIS technique

In this subsection, the obtained Pareto-optimal solutions, from NS-SCGA, undergo the TOPSIS techniques as a multiattribute decision analyzing technique, to rank the Pareto-optimal solutions according to the preference of DM and identifying the best compromise solutions. The principle of TOPSIS is based on obtaining the best compromise solution which is the farthest from the negative ideal solution (nis) and closest to the positive ideal solution (pis). The computational steps of TOPSIS technique which are applied to Pareto-optimal solutions of EEDP are as follows [68]:

- I. Input the objective matrix  $OM = (OM_{nm})$ , whose elements are the Pareto solutions. Each row of the objective matrix is allocated to one alternative and each column to one objective. Therefore, the element  $OM_{nm}$  of the objective matrix is the  $m$ th objective value of the  $n$ th optimal solution, where  $n = 1, 2, \dots, nd$  and  $m = 1, 2$ , as  $nd$  is the number of non-dominated solutions.
- II. Input as well a weight vector  $wn = [w_1, w_2]$  (weight of the two objectives) associated with the evaluation criteria that reflect the DM preference where  $w_1 + w_2 = 1$ .
- III. Normalize the OM to be normalized objective matrix NOR =  $(NOR_{nm})$  using the equation below to be the normalized value  $NOR_{nm}$



**Figure 10** | The obtained Pareto front of IEEE 30-bus 6-generator system by nondominated sorting sine-cosine genetic algorithm (NS-SCGA).

**Table 8** | Comparison of the best solutions obtained by different algorithms for minimizing the fuel cost.

	NPGA	MO-DE/PSO	MOHS	SPEA	NSGA-II	NSGA-II/EDA	TV-MOPSO	MOCDOA	NS-SCGA
<i>PG</i> <sub>1</sub>	0.1245	0.1220	0.06790	0.1086	0.1119	0.1399	0.1482	0.1204	0.1148
<i>PG</i> <sub>2</sub>	0.2792	0.2843	0.3515	0.3056	0.3265	0.3219	0.3062	0.2871	0.2901
<i>PG</i> <sub>3</sub>	0.6284	0.5857	0.5174	0.5818	0.5857	0.5597	0.5798	0.5803	0.5742
<i>PG</i> <sub>4</sub>	1.026	0.9962	0.8839	0.9846	0.9329	0.9487	1.000	0.9922	0.9922
<i>PG</i> <sub>5</sub>	0.4693	0.5149	0.5991	0.5288	0.5623	0.5254	0.4529	0.5259	0.5241
<i>PG</i> <sub>6</sub>	0.3993	0.3566	0.4317	0.3584	0.3404	0.3651	0.3725	0.3537	0.3619
<b>Fuel Cost</b>	608.1	606.0	606.3	607.8	606.6	606.5	606.4	606.0	605.4
<b>Emission</b>	0.2236	0.2209	0.2148	0.2202	0.2170	0.2164	0.2198	0.2206	0.2206
<b>NFEs</b>	1.000E5	1.000E4	N/a	1.000E5	2.250E4	2.250E4	2.500E4	2.000E4	2.000E4

**Table 9** | Comparison of the best solutions obtained by different algorithms for minimizing emissions.

	SPEA	NPGA	NSGA-II/EDA	NSGA-II	MOHS	MO-DE/PSO	TV-MOPSO	MOCDOA	NS-SCGA
<i>PG</i> <sub>1</sub>	0.4043	0.3923	0.4135	0.4135	0.4937	0.4118	0.3926	0.4103	0.3985
<i>PG</i> <sub>2</sub>	0.4525	0.4700	0.4662	0.4663	0.3908	0.4616	0.4724	0.4668	0.4626
<i>PG</i> <sub>3</sub>	0.5525	0.5565	0.5483	0.5482	0.5506	0.5435	0.5484	0.5437	0.5514
<i>PG</i> <sub>4</sub>	0.4079	0.3695	0.3948	0.3949	0.3774	0.3922	0.4133	0.3901	0.4066
<i>PG</i> <sub>5</sub>	0.5468	0.5599	0.5484	0.5483	0.5420	0.5454	0.5503	0.5426	0.5460
<i>PG</i> <sub>6</sub>	0.5005	0.5163	0.5187	0.5186	0.5021	0.5148	0.4909	0.5157	0.5025
<b>Emission</b>	0.1942	0.1942	0.1942	0.1942	0.1951	0.1942	0.1943	0.1942	0.1941
<b>Fuel Cost</b>	642.6	646.0	650.8	650.8	674.4	646.0	642.8	646.3	643.4

**Table 10** | SP metric values of the proposed NS-SCGA, TV-MOPSO, and MO-DE/PSO for EEDP.

Algorithm	Spacing		
	Best	Mean	Standard Deviation
NS-SCGA	0.005500	0.005987	0.0009411
TV-MOPSO	0.01080	0.01080	0.01740
MO-DE/PSO	0.005100	0.007400	0.002800

As follows:

$$NOR_{nm} = \frac{OM_{nm}}{\sqrt{\sum_{m=1}^2 (OM_{nm})}}, \quad n = 1, 2, \dots, nd \text{ and } m = 1, 2. \tag{15}$$

where  $NOR_{nm}$  is the normalized rating;  $OM_{nm}$  is the  $m$ th objective value of the  $n$ th optimal solution.

- IV. Determine the weighted normalized objective matrix  $WN = (WN_{nm})$ , and the weighted normalized value  $WN_{nm}$  is calculated by

$$WN_{nm} = wn_m \times NOR_{nm} \tag{16}$$

- V. Calculate  $nis$  and  $pis$  individually as follows:

$$pis = \{(max\ WN_{nm} | m \in c), (min\ WN_{nm} | m \in e)\}, \quad n = 1, 2, \dots, nd \tag{17}$$

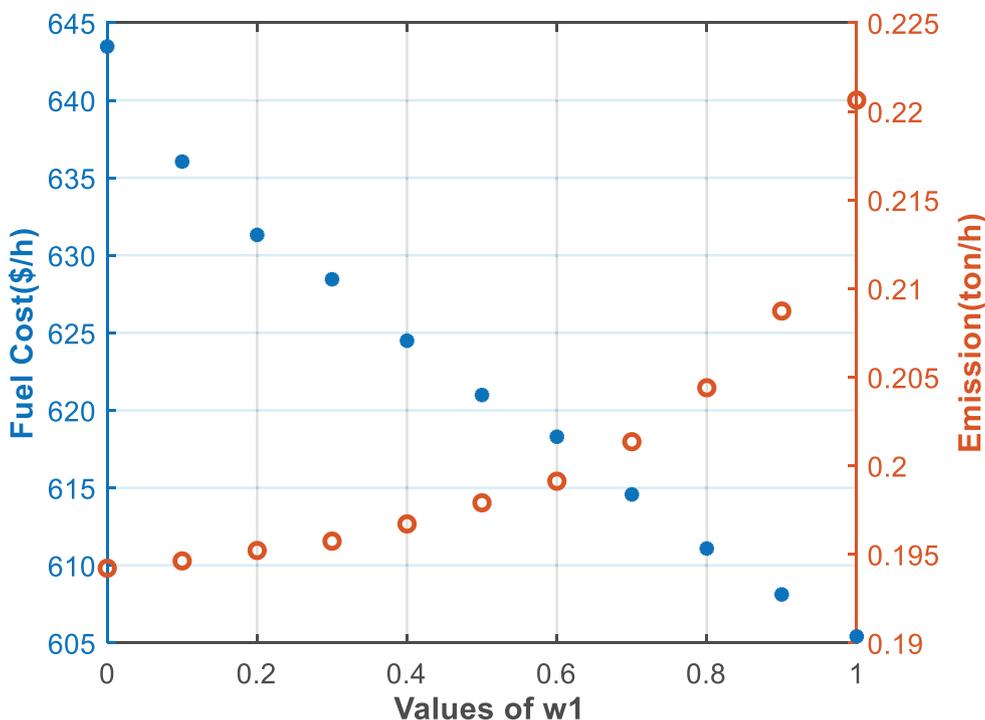


Figure 11 | The best compromise solutions in the objective space according to  $w_1$ .

$$nis = \{(min WN_{nm} | m \in c), (max WN_{nm} | m \in e)\}, \quad (18)$$

$$n = 1, 2, \dots, nd$$

where  $c = \{1\}$  and  $e = \{2\}$ .

- VI. Calculate the separation measure  $sep$  and  $sn$  from the  $pis$  and  $nis$ , respectively:

$$sep_n = \left( \sum_m (WN_{nm} - pis_m)^2 \right)^{1/2}, \quad n = 1, 2, \dots, nd; \quad (19)$$

$$m = 1, 2.$$

$$sn_n = \left( \sum_m (WN_{nm} - nis_m)^2 \right)^{1/2}, \quad n = 1, 2, \dots, nd; \quad (20)$$

$$m = 1, 2.$$

- VII. Calculate the relative closeness  $\bar{c}_n^*$  for each nondominated solution by

$$\bar{c}_n^* = \frac{sn_n}{sep_n + sn_n}, \quad n = 1, \dots, nd \quad (21)$$

- VIII. Identify the best compromise solution which has relative closeness  $\bar{c}_n^*$  that is as close to 1 as possible.

The above steps are applied to the obtained Pareto-optimal solutions by NS-SCGA.  $w_1$  and  $w_2$  represent the DM's preference of each objective, where  $w_1$  is the weight of fuel cost (i.e., the relative importance of the fuel cost) and  $w_2$  is the weight of emission (i.e., the relative importance of the emission).

In order to study the effect of changing the weights on the fuel cost and emission, the weight of fuel cost is assumed  $w_1 = 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1$ , and the weight of emission is obtained

Table 11 | The best compromise solutions for fuel cost and emission according to  $w_1$ .

$w_1$	Fuel cost	Emission	$PG_1$	$PG_2$	$PG_3$	$PG_4$	$PG_5$	$PG_6$
0	643.5	0.1942	0.3985	0.4626	0.5514	0.4066	0.5460	0.5025
0.1	636.0	0.1946	0.3690	0.4441	0.5544	0.4664	0.5445	0.4867
0.2	631.3	0.1952	0.3490	0.4303	0.5505	0.5108	0.5425	0.4806
0.3	628.5	0.1957	0.3377	0.4211	0.5486	0.5373	0.5509	0.4672
0.4	624.5	0.1967	0.3149	0.4086	0.5618	0.5761	0.5407	0.4594
0.5	621.0	0.1979	0.2960	0.3967	0.5602	0.6174	0.5405	0.4497
0.6	618.3	0.1991	0.2821	0.3857	0.5561	0.6490	0.5587	0.4279
0.7	614.6	0.2013	0.2471	0.3839	0.5569	0.7007	0.5585	0.4112
0.8	611.1	0.2044	0.2233	0.3558	0.56290	0.7685	0.5322	0.4149
0.9	608.1	0.2087	0.1958	0.3329	0.5662	0.8424	0.5290	0.3908
1.0	605.4	0.2206	0.1148	0.2901	0.5742	0.9922	0.5241	0.3619

by  $w_2 = 1 - w_1$ . According to different weights, TOPSIS determines the best compromise solution for EEDP. The different values of weight  $w_1$  versus the best compromise solution of fuel cost and emission are shown in Figure 11. Table 11 gives the values of the weights and the best compromise solution (objective functions) chosen from the Pareto-optimal solutions and corresponding optimum generation schedule of available generators versus weights for the 11 cases. It is obvious that, for each weight, different best compromise solutions have been found in proportional to the criterion importance for each objective function.

## 7. CONCLUSIONS

A new MO algorithm called NS-SCGA is presented in this paper. The NS-SCGA is tested on 20 challenging benchmark problems from CEC09 for UCMOPs and CMOPs and its performance is calculated by IGD, spacing, and spreading metrics. NS-SCGA has been compared with another 11 newly developed algorithms and another 8 recent algorithms for unconstrained problems and constrained

problems, respectively. The LO is used to assess the overall performance of the proposed NS-SCGA among the compared techniques. The statistical analysis of the experimental results is also carried out by nonparametric Friedman and Wilcoxon signed-rank tests. In addition, NS-SCGA has been applied to handle EEDP. The EEDP is formulated as a MOP with two conflicting objectives, that is, cost and emission. NS-SCGA has been validated on the IEEE 30-bus 6-generator system. The obtained results by the NS-SCGA have been compared with other metaheuristic algorithms. It is evident from the comparison that the NS-SCGA gives a competitive performance in terms of solution. Finally, the TOPSIS method is used to identify the best compromise solution from the obtained nondominated solutions of EEDP according to the DM's preference.

The results prove that NS-SCGA is better or on a par with the other modern methods for solving MOPs. It is robust, efficient, and simple, and it has the possibility of being applied to solve some other MO power system problems.

## CONFLICT OF INTEREST

The authors declare no conflict of interest.

## AUTHORS' CONTRIBUTIONS

All authors are equally contributed in this article.

## Funding Statement

The authors received no specific funding for this work.

## ACKNOWLEDGMENTS

The authors would like to thank the referees for valuable remarks and suggestions that helped to increase the clarity of arguments and to improve the structure of the paper. In addition, thanks are due to Mr. Muhammad Anwar (muhammad.anwar.hashim@gmail.com), MA student in linguistics and English instructor at German University in Cairo, for English-language editing.

## REFERENCES

- [1] M.A. Tawhid, V. Savsani, A novel multi-objective optimization algorithm based on artificial algae for multi-objective engineering design problems, *Appl. Intell.* 48 (2018), 3762–3781.
- [2] K. Deb, Multi-objective optimization, in: E. Burke, G. Kendall (Eds.), *Search Methodologies*, Springer, Boston, MA, USA, 2014, pp. 403–449.
- [3] L.M. Rios, N.V. Sahinidis, Derivative-free optimization: a review of algorithms and comparison of software implementations, *J. Global Optim.* 56 (2013), 1247–1293.
- [4] B. Akay, Synchronous and asynchronous Pareto-based multi-objective artificial bee colony algorithms. *J. Global Optim.* 57 (2013), 415–445.
- [5] E. Zitzler, *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*, vol. 63, Shaker, Ithaca, NY, USA, 1999.
- [6] T. Murata, H. Ishibuchi, MOGA: multi-objective genetic algorithms, in *IEEE International Conference on Evolutionary Computation*, Perth, Australia, 1995, vol. 1, pp. 289–294.
- [7] W.H. Chen, P.H. Wu, Y.L. Lin, Performance optimization of thermoelectric generators designed by multi-objective genetic algorithm, *Appl. Energy*. 209 (2018), 211–223.
- [8] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, in: M. Schoenauer, et al. (Eds.), *International Conference on Parallel Problem Solving From Nature*, Springer, Berlin, Heidelberg, Germany, 2000, pp. 849–858.
- [9] M.A. Abido, A Niche Pareto genetic algorithm for multiobjective environmental/economic dispatch, *Int. J. Electr. Power Energy*. 25 (2003), 97–105.
- [10] A.M. Shaheen, R.A. El-Sehiemy, S.M. Farrag, Solving multi-objective optimal power flow problem via forced initialised differential evolution algorithm, *IET Gener. Transm. Distrib.* 10 (2016), 1634–1647.
- [11] M.A. Abido, Environmental/economic power dispatch using multiobjective evolutionary algorithms, in 2003 IEEE Power Engineering Society General Meeting (IEEE Cat. No. 03CH37491), Toronto, Canada, 2003, vol. 2, pp. 920–925.
- [12] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: improving the strength Pareto evolutionary algorithm for multiobjective optimization, in *Proceedings of The EUROGEN 2001 – Evolutionary Methods for Design Optimisation and Control with Applications to Industrial Problems*, Barcelona, Spain, 2001. <https://pdfs.semanticscholar.org/6672/8d01f9ebd0446ab346a855a44d2b138fd82d.pdf>
- [13] J. Knowles, D. Corne, The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimization, in *Congress on Evolutionary Computation (CEC99)*, Washington, DC, USA, 1999, vol. 1, pp. 98–105.
- [14] Q. Zhang, W. Liu, H. Li, The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances, in 2009 IEEE Congress on Evolutionary Computation, Trondheim, Norway, 2009, pp. 203–208.
- [15] H.L. Liu, X. Li, The multiobjective evolutionary algorithm based on determined weight and sub-regional search, in 2009 IEEE Congress on Evolutionary Computation, Trondheim, Norway, 2009, pp. 1928–1934.
- [16] E. Zitzler, S. Künzli, Indicator-based selection in multiobjective search, in: X. Yao, et al. (Eds.), *International Conference on Parallel Problem Solving from Nature*, Springer, Berlin, Heidelberg, Germany, 2004, pp. 832–842.
- [17] H.H. Benderbal, M. Dahane, L. Benyoucef, Modularity assessment in Reconfigurable Manufacturing System (RMS) design: an archived multi-objective simulated annealing-based approach, *Int. J. Adv. Manuf. Technol.* 94 (2018), 729–749.
- [18] A.A. Mousa, M.A. El-Shorbagy, W.F. Abd-El-Wahed, Local search-based hybrid particle swarm optimization algorithm for multiobjective optimization, *Swarm Evol. Comput.* 3 (2012), 1–14.
- [19] Y. Wang, C. Dang, H. Li, L. Han, J. Wei, A clustering multi-objective evolutionary algorithm based on orthogonal and uniform design, in 2009 IEEE Congress on Evolutionary Computation, Trondheim, Norway, 2009, pp. 2927–2933.
- [20] L.Y. Tseng, C. Chen, Multiple trajectory search for unconstrained/constrained multi-objective optimization, in 2009 IEEE Congress on Evolutionary Computation, Trondheim, Norway, 2009, pp. 1951–1958.

- [21] V. Patel, V. Savsani, A multi-objective improved teaching– learning based optimization algorithm (MO-ITLBO), *Inf. Sci.* 357 (2016), 182–200.
- [22] X.S. Yang, Multiobjective firefly algorithm for continuous optimization, *Eng. Comput.* 29 (2013), 175–184.
- [23] X.S. Yang, S. Deb, Multiobjective Cuckoo search for design optimization, *Comput. Oper. Res.* 40 (2013), 1616–1624.
- [24] Y. Gao, H. Guan, Z. Qi, Y. Hou, L. Liu, A multi-objective ant colony system algorithm for virtual machine placement in cloud computing, *J. Comput. Syst. Sci.* 79 (2013), 1230–1242.
- [25] S. Mirjalili, P. Jangir, S. Saremi, Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems, *Appl. Intell.* 46 (2017), 79–95.
- [26] A. Sadollah, H. Eskandar, A. Bahreinejad, J.H. Kim, Water cycle algorithm for solving multi-objective optimization problems, *Soft Comput.* 19 (2015), 2587–2603.
- [27] S. Mirjalili, S. Saremi, S.M. Mirjalili, L.D.S. Coelho, Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization, *Expert Syst. Appl.* 47 (2016), 106–119.
- [28] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, *Neural Comput. Appl.* 27 (2016), 1053–1073.
- [29] J. Wang, W. Yang, P. Du, T. Niu, A novel hybrid forecasting system of wind speed based on a newly developed multi-objective sine cosine algorithm, *Energy Convers. Manag.* 163 (2018), 134–150.
- [30] M.A. Tawhid, V. Savsani, Multi-Objective Sine-Cosine Algorithm (MO-SCA) for multi-objective engineering design problems, *Neural Comput. Appl.* 31 (2019), 915–929.
- [31] D.W. Gong, Y. Zhang, C.L. Qi, Environmental/economic power dispatch using a hybrid multi-objective optimization algorithm, *Int. J. Electr. Power Energy. Syst.* 32 (2010), 607–614.
- [32] A.A. El-Sawy, Z.M. Hendawy, M.A. El-Shorbagy, Reference point based TR-PSO for multi-objective environmental/economic dispatch, *Appl. Math.* 4 (2013), 803.
- [33] G.G. Wang, A.H. Gandomi, X. Zhao, H.C. Chu, Hybridizing harmony search algorithm with cuckoo search for global numerical optimization, *Soft Comput.* 20 (2016), 273–285.
- [34] T. Cheng, M. Chen, P.J. Fleming, Z. Yang, S. Gan, A novel hybrid teaching learning based multi-objective particle swarm optimization, *Neurocomputing.* 222 (2017), 11–25.
- [35] A. Kishor, P.K. Singh, J. Prakash, NSABC: non-dominated sorting based multi-objective artificial bee colony algorithm and its application in data clustering, *Neurocomputing.* 216 (2016), 514–533.
- [36] M.A. Farag, M.A. El-Shorbagy, I.M. El-Desoky, A.A. El-Sawy, A.A. Mousa, Binary-real coded genetic algorithm based K-means clustering for unit commitment problem, *Appl. Math.* 6 (2015), 1873.
- [37] S.H.R. Pasandideh, S. Khalilpourazari, Sine cosine crow search algorithm: a powerful hybrid meta heuristic for global optimization, *Third International Conference on Artificial Intelligence and Soft Computing*, Dubai, UAE, Arxiv Preprint arXiv:1801.08485, 2018. <https://arxiv.org/ftp/arxiv/papers/1801/1801.08485.pdf>
- [38] K.O. Alawode, G.A. Adegboyega, J. Abimbola Muhideen, NSGA-II/EDA hybrid evolutionary algorithm for solving multi-objective economic/emission dispatch problem, *Electr. Power Compo. Syst.* 46 (2018), 1060–1172.
- [39] V. Savsani, M.A. Tawhid, Non-Dominated Sorting Moth Flame Optimization (NS-MFO) for multi-objective problems, *Eng. Appl. Artif. Intell.* 63 (2017), 20–32.
- [40] A.A. Mousa, M.A. El-Shorbagy, M.A. Farag, K-means-clustering based evolutionary algorithm for multi-objective resource allocation problems, *Appl. Math. Inf. Sci.* 11 (2017), 1681–1692.
- [41] M.A. El-Shorbagy, A.A. Mousa, M. Farag, Solving nonlinear single-unit commitment problem by genetic algorithm based clustering technique, *Rev. Comput. Eng. Res.* 4 (2017), 11–29.
- [42] H. Zhang, B. Lennox, P.R. Goulding, A.Y. Leung, A float-encoded genetic algorithm technique for integrated optimization of piezoelectric actuator and sensor placement and feedback gains, *Smart Mater. Struct.* 9 (2000), 552.
- [43] Y. Chen, Y. Li, *Computational Intelligence Assisted Design: in Industrial Revolution 4.0.*, CRC Press, Boca Raton, FL, USA, 2018.
- [44] S. Carlucci, G. Cattarin, F. Causone, L. Pagliano, Multi-objective optimization of a nearly zero-energy building based on thermal and visual discomfort minimization using a Non-Dominated Sorting Genetic Algorithm (NSGA-II), *Energy Build.* 104 (2015), 378–394.
- [45] R. Bolaños, M. Echeverry, J. Escobar, A multiobjective Non-Dominated Sorting Genetic Algorithm (NSGA-II) for the multiple traveling salesman problem, *Decis. Sci. Lett.* 4 (2015), 559–568.
- [46] T. Vo-Duy, D. Duong-Gia, V. Ho-Huu, H.C. Vu-Do, T. Nguyen-Thoi, Multi-objective optimization of laminated composite beam structures using NSGA-II algorithm, *Compos. Struct.* 168 (2017), 498–509.
- [47] K. Sindhya, A. Sinha, K. Deb, K. Miettinen, Local search based evolutionary multi-objective optimization algorithm for constrained and unconstrained problems, in *Proceeding of Congress on Evolutionary Computation (CEC'09)*, Trondheim, Norway, 2009, pp. 2919–2926.
- [48] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, *Knowl. Based Syst.* 96 (2016), 120–133.
- [49] S.M. Mirjalili, S.Z. Mirjalili, S. Saremi, S. Mirjalili, Sine cosine algorithm: theory, literature review, and application in designing bend photonic crystal waveguides, in: S. Mirjalili, J. Song Dong, A. Lewis (Eds.), *Nature-Inspired Optimizers*, Springer, Cham, Switzerland, 2020, pp. 201–217.
- [50] A. F Attia, R.A. El Sheimy, H.M. Hasanien, Optimal power flow solution in power systems using a novel sine-cosine algorithm, *Int. J. Electr. Power Energy. Syst.* 99 (2018), 331–343.
- [51] S. Li, H. Fang, X. Liu, Parameter optimization of support vector regression based on sine cosine algorithm, *Expert Syst. Appl.* 91 (2018), 63–77.
- [52] M.A. El-Shorbagy, M.A. Farag, A.A.Mousa, I.M. El-Desoky, A hybridization of sine cosine algorithm with steady state genetic algorithm for engineering design problems, in: A. Hasanien, A. Azar, T. Gaber, R. Bhatnagar, M. Tolba (Eds.), *International Conference on Advanced Machine Learning Technologies and Applications*, Springer, Cham, Switzerland, 2019, pp. 143–155.
- [53] C. Qu, Z. Zeng, J. Dai, Z. Yi, W. He, A modified sine-cosine algorithm based on neighborhood search and greedy levy mutation, *Comput. Intell. Neurosci.* 62 (2018), 1019–1043.
- [54] H. Nenavath, R.K. Jatoth, Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking, *Appl. Soft Comput.* 62 (2018), 1019–1043.
- [55] S.N. Chegini, A. Bagheri, F. Najafi, PSOSCALF: a new hybrid PSO based on sine cosine algorithm and levy flight for solving optimization problems, *Appl. Soft Comput.* 73 (2018), 697–726.

- [56] S. Bureerat, N. Pholdee, Adaptive sine cosine algorithm integrated with differential evolution for structural damage detection, in: O. Gervasi, et al. (Eds.), *International Conference on Computational Science and its Applications*, Springer, Cham, Switzerland, 2017, pp. 71–86.
- [57] A. Chakrabarti, S. Halder, *Power System analysis: Operation and Control*, PHI Learning Pvt. Ltd, Delhi, India, 2006, pp. 425–547. <https://books.google.com.et/books?id=Z9zqsx2KEfEC&printsec=frontcover#v=onepage&q&f=false>
- [58] A. Farang, S. Al-Baiyat, T.C Cheng, Economic load dispatch multi-objective optimization procedures using linear programming techniques, *IEEE Trans. PWRs.* 10 (1995), 731–738.
- [59] J.S. Dhillon, S.C. Parti, D.P. Kothari, Stochastic economic emission load dispatch, *Electr. Power Syst. Res.* 26 (1993), 179–176.
- [60] J.Y. Fan, L. Zhang, Real-time economic dispatch with line flow and emission constraints using quadratic programming, *IEEE Trans. Power Syst.* 13 (1998), 320–325.
- [61] C.L. Chen, S.C. Wang, Branch and bound scheduling for thermal generating units, *IEEE Trans. Energy Convers.* 8 (1993), 184–189.
- [62] J. Nanda, L. Hari, M.L. Kothari, Economic emission dispatch with line flow constraints using a classical technique, *IEEE Proc. Gener. Trans. Distrib.* 141 (1994), 1–10.
- [63] A.Y. Abdelaziz, E.S. Ali, S.A. Elazim, Implementation of flower pollination algorithm for solving economic load dispatch and combined economic emission dispatch problems in power systems, *Energy.* 101 (2016), 506–518.
- [64] T. Niknam, H.D. Mojarrad, H.Z. Meymand, A new particle swarm optimization for non-convex economic dispatch, *Eur. Trans. Electr. Power.* 21 (2011), 656–679.
- [65] 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, *Inf. Sci.* 351 (2016), 48–66.
- [66] M. Mohammadian, A. Lorestani, M.M. Ardehali, Optimization of single and multi-areas economic dispatch problems based on evolutionary particle swarm optimization algorithm, *Energy.* 161 (2018), 710–724.
- [67] M.J. Mokarram, T. Niknam, J. Aghaei, M. Shafie-Khah, J.P. Catalão, Hybrid optimization algorithm to solve the nonconvex multiarea economic dispatch problem, *IEEE Syst. J.* 13 (2019), 3400–3409.
- [68] C.-L. Hwang, K. Yoon, *Multiple Attribute Decision Making: Methods and Applications*, Springer-Verlag, New York, NY, USA, 1981, pp. 58–191.
- [69] K. Miettinen, *Nonlinear Multiobjective Optimization*, vol. 12, Springer Science & Business Media, New York, NY, USA, 2012.
- [70] K. Deb, N. Padhye, G. Neema, Multiobjective evolutionary optimization-interplanetary trajectory optimization with swing-bys using evolutionary multi-objective optimization, in: L. Kang, Y. Liu, S. Zeng (Eds.), *Advances in Computation and Intelligence*, Lecture Notes Computer Science, vol. 4683, Springer, Berlin, Heidelberg, Germany, 2007, pp. 26–35.
- [71] R. Wang, J. Xiong, H. Ishibuchi, G. Wu, T. Zhang, On the effect of reference point in MOEA/D for multi-objective optimization, *Appl. Soft Comput.* 58 (2017), 25–34.
- [72] S.A. Sadatsakkak, M.H. Ahmadi, R. Bayat, S.M. Pourkiaei, M. Feidt, Optimization density power and thermal efficiency of an endoreversible Braysson cycle by using non-dominated sorting genetic algorithm, *Energy Convers. Manag.* 93 (2015), 31–39.
- [73] M.H. Ahmadi, A.H. Mohammadi, S. Dehghani, A. Barranco-Jimenez-Marco, Multiobjective thermodynamic-based optimization of output power of solar dish–stirling engine by implementing an evolutionary algorithm, *Energy Convers. Manag.* 75 (2013), 438–45.
- [74] P. Mohapatra, A. Nayak, S.K. Kumar, M.K. Tiwari, Multi-objective process planning and scheduling using controlled elitist non-dominated sorting genetic algorithm, *Int. J. Prod. Res.* 53 (2015), 1712–1735.
- [75] R.V. Rao, D.P. Rai, J. Balic, Multi-objective optimization of machining and micro-machining processes using non-dominated sorting teaching–learning-based optimization algorithm, *J. Intell. Manuf.* 29 (2018), 1715–1737.
- [76] U. Raut, S. Mishra, Power distribution network reconfiguration using an improved sine–cosine algorithm-based meta-heuristic search, in: J. Bansal, K. Das, A. Nagar, K. Deep, A. Ojha (Eds.), *Soft Computing for Problem Solving*, Springer, Singapore, 2019, pp. 1–13.
- [77] M.R. Sierra, C.A.C. Coello, Improving PSO-based multi-objective optimization using crowding, mutation and  $\epsilon$ -dominance, in: C.A. Coello Coello, A. Hernández Aguirre, E. Zitzler (Eds.), *International Conference on Evolutionary Multi-Criterion Optimization*, Springer, Berlin, Heidelberg, Germany, 2005, pp. 505–519.
- [78] C.A.C. Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, *IEEE Trans. Evol. Comput.* 8 (2004), 256–279.
- [79] L. Miqing, X. Yao, Quality evaluation of solution sets in multiobjective optimisation: a survey, *ACM Comput. Surv.* 52 (2019), 26.
- [80] Q. Zhang, A. Zhou, S. Zhao, P.N. Suganthan, W. Liu, S. Tiwari, Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition, Special Session On Performance Assessment Of Multi-Objective Optimization Algorithms, Technical Report, University of Essex, Colchester, UK and Nanyang Technological University, Singapore, 2008. [https://www.researchgate.net/publication/265432807\\_Multiobjective\\_optimization\\_Test\\_Instances\\_for\\_the\\_CEC\\_2009\\_Special\\_Session\\_and\\_Competition](https://www.researchgate.net/publication/265432807_Multiobjective_optimization_Test_Instances_for_the_CEC_2009_Special_Session_and_Competition)
- [81] C.M. Chen, Y.P. Chen, Q. Zhang, Enhancing MOEA/D with guided mutation and priority update for multi-objective optimization, in 2009 IEEE Congress on Evolutionary Computation, Trondheim, Norway, 2009, pp. 209–216.
- [82] M. Liu, X. Zou, Y. Chen, Z. Wu, Performance assessment of DMOEA-DD with CEC 2009 MOEA competition test instances, in 2009 IEEE Congress on Evolutionary Computation, Trondheim, Norway, 2009, pp. 2913–2918.
- [83] R. Akbari, R. Hedayatzadeh, K. Ziarati, B. Hassanizadeh, A multi-objective artificial bee colony algorithm, *Swarm Evol. Comput.* 2 (2012), 39–52.
- [84] W.K. Mashwani, A. Salhi, O. Yeniay, H. Hussian, M.A. Jan, Hybrid non-dominated sorting genetic algorithm with adaptive operators selection, *Appl. Soft Comput.* 56 (2017), 1–18.
- [85] J. Chen, J. Li, B. Xin, DMOEA- $\epsilon$ C: decomposition-based multi-objective evolutionary algorithm with the  $\epsilon$ -constraint framework, *IEEE Trans. Evol. Comput.* 21 (2017), 714–730.
- [86] X. Qiu, J.X. Xu, K.C. Tan, H.A. Abbass, Adaptive cross-generation differential evolution operators for multiobjective optimization, *IEEE Trans. Evol. Comput.* 20 (2016), 232–244.

- [87] S. Saremi, S. Mirjalili, A. Lewis, A.W.C. Liew, J.S. Dong, Enhanced multi-objective particle swarm optimisation for estimating hand postures, *Knowl. Based Syst.* 158 (2018), 175–195.
- [88] S.Z. Mirjalili, S. Mirjalili, S. Saremi, H. Faris, I. Aljarah, Grasshopper optimization algorithm for multi-objective optimization problems, *Appl. Intell.* 48 (2018), 805–820.
- [89] A. Zamuda, J. Brest, B. Boskovic, V. Zumer, Differential evolution with self-adaptation and local search for constrained multiobjective optimization, in 2009 IEEE Congress on Evolutionary Computation, Trondheim, Norway, 2009, pp. 195–202.
- [90] J. Derrac, S. Garcia, D. Molina, F. Herrera, A Practical tutorial on the use of nonparametric statistical tests as methodology for comparing evolutionary intelligence algorithms, *Swarm Evol. Comput.* 1 (2011), 3–18.
- [91] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, *Inform Sci.* 180 (2010), 2044–2064.
- [92] M.A. Abido, Multiobjective evolutionary algorithms for electric power dispatch problem, *IEEE Trans. Evol. Comput.* 10 (2006), 315–329.
- [93] T.C. Bora, V.C. Mariani, L. Dos Santos Coelho, Multi-objective optimization of the environmental-economic dispatch with reinforcement learning based on non-dominated sorting genetic algorithm, *Appl. Therm. Eng.* 146 (2019), 688–700.
- [94] P.K. Tripathi, S. Bandyopadhyay, S.K. Pal, Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients, *Inform Sci.* 177 (2007), 5033–5049.
- [95] S. Sivasubramani, K.S. Swarup, Environmental/economic dispatch using multi-objective harmony search algorithm, *Electr. Power Syst. Res.* 81 (2011), 1778–1785.
- [96] X. Xu, Z. Hu, Q. Su, Z. Xiong, Multiobjective collective decision optimization algorithm for economic emission dispatch problem, *Complexity.* 2018 (2018), 1–20.