Multi-Method based Algorithm for Multi-objective Problems under Uncertainty

Forhad Zaman*, Saber M. Elsayed*, Ruhul Sarker*, Daryl Essam* and Carlos A. Coello Coello**

School of Engineering and Information Technology, University of New South Wales, Canberra, Australia

f.zaman@unsw.edu.au; s.elsayed@unsw.edu.au; r.sarker@unsw.edu.au; and d.essam@unsw.edu.au

***Depto. de Computación, CINVESTAV-IPN, Mexico, email: ccoello@cs.cinvestav.mx*

Abstract

Many real-world decision problems consider more than one objective. These objectives usually conflict with each other. In the problem environments, many data and parameters are usually uncertain at the time of planning. So multi-objective optimization combined with uncertainty is a challenging research topic. To deal with the basic search process in such problems, in this paper, a new approach has been proposed that combines multiple population based algorithms under a single algorithm structure. To deal with the uncertainty, a dynamic scenario-based approach is developed and integrated with the search process, in which the number of scenarios is dynamically set during the solution process. This is a new way of solving multi-objective optimization problems under uncertainty. To judge the performance of the proposed approach, we have solved a set of standard test problems without uncertainty and a number of practical problems with uncertainty. The practical problems are the well-known dynamic economic and emission dispatch problems, with different combinations of energy sources. The experimental studies demonstrated that the proposed approach performs better than other well-known algorithms compared in this paper.

Keywords: multi-objective optimization, evolutionary algorithm; dynamic economic and environmental dispatch; renewable energy; uncertainty.

1. Introduction

Over the last few decades, multi-objective optimization has become a very popular decision making tool in practice, and also a challenging research topic, where the problem objectives cannot be represented by a single objective function. The basic condition of multi-objective optimization is that these objectives must be conflicting to each other. If they are not conflicting, they can

be combined together as a single objective function with appropriate scaling. In multi-objective optimization problems (MOPs), there is no single optimal solution, instead a set of non-dominated solutions (also called Pareto front) is generated [3].

The optimization algorithms for MOPs are categorized into two types: a *priori* and *posteriori* [32]. In the former, a single-objective function is generated based on a weighted sum of all the considered objectives. With a given set of weights, the MOP can be solved using a single objective algorithm that generates a single solution. The major drawback of this approach is to determine the right weights for the objectives. However, depending on the mathematical properties of the functions, multiple solutions may be generated by changing the weights, where one solution is generated through a single run of the algorithm. That means it is necessary to repeatedly run the algorithm for multiple solutions. A *posteriori* approach optimizes all the objective functions simultaneously, so that it finds a set of non-dominated solutions in a single run. For this approach, the population-based algorithms, such as multi-objective evolutionary algorithms (MOEAs) have been receiving growing interest, in solving many real-world complex MOPs, due to their flexible, efficient search features and the generation of many non-dominated solutions in a single run [32, 47, 33]. Among the existing MOEAs, non-dominated sorting genetic algorithm-II (NSGA-II), multi-objective particle swarm optimization (MOPSO), multi-objective differential evolution (MODE), multi-objective bee Algorithm and multi-objective bat algorithm (MOBA) have shown admirable performances for many problems [32].

The existing MOEAs have been widely applied to MOPs where the data and parameters are known with certainty. There are only very limited attempts to deal with MOPs under uncertainty, which is inevitable in many real-world applications, such as power system, mining, and project scheduling [24]. Over the last few years, several scenario-based approaches have been developed for solving different optimization problems with uncertainty [5, 48]. In such approaches, a scenario represents an instance of the problem with a fixed set of parameters. Those parameters are generated using a stochastic representation, such as the probability distribution of the given uncertain parameters [48]. Once a specified number of scenarios is generated, all the problem instances are solved. To increase the solution stability of this approach, it is necessary to consider a large number of scenarios, which renders these algorithms computationally burden for a large system [34].

Considering the context of multi-objective optimization with uncertainty, the dynamic economic and emission dispatch (DEED) problem is an excellent example from real-world applications [21]. In this problem, it is intended to minimize both fuel cost and gas emission, simultaneously, by scheduling the given electrical generators, that include fossil-fuel-base as well as renewable generators under an uncertain environment. As the generation of renewable energy fluctuates greatly, scheduling the right mix of generation from a number of renewable and thermal units to satisfy a daily load demand, while dealing with the generation capacity and technical constraints, is a challenging optimization

problem [47, 36]. Although many efficient algorithms based on different evolutionary algorithms (EAs) have been developed [47], most of them disregard the uncertainties of renewable energy production, in spite of the fact that managing them is a critical issue when solving the DEED problem.

It is well-known that no single algorithm performs well for solving a wide range of MOPs, with and without considering their uncertainties. This might be due to the underlying characteristics of optimization problems are not consistent. Although multi-method based multi-operator algorithms have been successfully applied for solving several optimization problems without uncertainty [48, 18], to our best knowledge, adopting them to solve MOPs under uncertainty has not yet been explicitly explored.

In this paper, we develop a multi-method-based bi-objective algorithm (MMBA) for solving MOPs, with or without uncertainties. The MMBA considers three efficient algorithms, namely NSGA-II, MODE and MOPSO, based on their demonstrated performances in solving many real-world optimization problems. These three algorithms are evolved in three different sub-populations, where the algorithms perform sequentially one after another. In the initial generation, the same numbers of individuals are allocated to each sub-population. However, in subsequent generations, the sizes of the sub-populations are varied, based on their performance in previous generations, with the best algorithm involving more individuals than the others. After a certain number of generations (say, a cycle), only the best algorithm performs for the next cycle, and it considers all the individuals from the other two algorithms, while they are not run. Once that cycle is completed, all three algorithms are restarted with equal numbers of individuals, and the above steps are repeated. This process continues until a stopping criterion is reached. Furthermore, to increase the convergence rate of the MMBA algorithm, a heuristic repair technique is employed to reduce the sum of constraints violations (CV) for infeasible individuals.

The performance of the proposed algorithm is demonstrated by solving six standard MOPs, and three DEED problems, with and without uncertainty. The experimental results demonstrate the superiority of the proposed approach, when compared with other well-known approaches.

The key contributions of this study can be summarized as follow:

- Developing a new multi-method based bi-objective optimization algorithm for solving MOPs, with and without uncertainty.
- Introducing a new way of handling uncertainty in MOPs.
- Proposing a new heuristic to repair infeasible individuals into feasible ones.

The rest of this paper is organized as follows: Section 2 provides a literature review, Section 3 presents the problem formulation, Section 4 contains the proposed MMBA algorithm, Section 5 reports our experimental results and their corresponding analysis, and Section 6 provides our conclusions and some possible directions for future research.

2. Literature Review

Over the last few decades, a significant number of MOEAs have been developed for solving different MOPs. Of them, multi-objective GA, niched Pareto genetic algorithm, NSGA-II, strength Pareto evolutionary algorithm-2, MODE, and MOPSO are well-known for solving MOPs efficiently. The general idea of the above-mentioned algorithms is almost similar, in that the best non-dominated solutions are stored, and updated in each iteration. The primary difference between one to another algorithm is to set a mechanism to update the non-dominated solutions. However, most of the algorithms find a common challenge in maintaining a good convergence and coverage of the selected non-dominated solutions, as the ultimate goal of a MOEA is to find an approximate Pareto-front that covers maximum areas of the true optimal Pareto front. In the selection operator, convergence helps by pushing solutions toward optimal or sub-optimal areas, while coverage ensures that selected non-dominated solutions cover the entire area of the Pareto-front. Nevertheless, they are conflicting, if a MOEA emphasizes on accuracy (convergence), then diversity (coverage) will be poor, and vice-versa. Although most of the existing MOEAs periodically maintain both convergence and coverage of the non-dominated solutions during the search process, that is not guaranteed for many complex MOPs [32].

An alternative but popular approach is the decomposition based MOEAs (MOEAs/D), in which a MOP is decomposed into a number of scalar optimization problems with adjusted different weight-values for different objectives. Subsequently, the scalar problems are solved as single-objective optimization problems. However, this process is assumed to be overextended for some MOPs. In that case, sometimes grid-based approaches are used, in which the non-dominated solutions are selected by using a grid-based method towards the optimal direction while conserving identical distribution between the selected solutions [31].

However, no algorithm performs well for a wide range of optimization problems, with each sometimes showing a better performance in one problem, but a worse performance in another [47]. To overcome this, recently several researchers have designed ensemble algorithmic frameworks that consider multiple algorithms, search operators, and heuristics [19]. Initially, ensemble techniques were used to enhance the performance of a single EA by using multiple search operators; for example, Gong et. al. [22] enhanced some existing single objective EAs by employing a new ensemble technique, based on a surrogate model. Subsequently, ensemble techniques have been widely used in various optimization areas, such as single objective, constrained, multi-objective, and multi-modal optimization problems [43]. For MOPs, Haiping et. al. [29] developed an ensemble algorithm based on three population-based algorithms, in which they performed in parallel in three different populations with new offspring generated using their own operators. After a generation, the selected individuals from all algorithms were combined to update the parent population. This process led to a better set of solutions at the end.

Although ensemble approaches have shown great performance in solving

some problems, they have not been studied well for optimization with uncertainty, particularly in uncertain MOPs. Solving an uncertain MOP with the population-based algorithms, is challenging due to the problem characteristics [45]. In practice, many real-world problems have been formulated as bi-objective optimization problems with uncertainty. One of the complex optimization problems in power system, namely DEED problems, are formulated as uncertain bi-objective problems, with uncertainties in renewable generations and load demands [48]. During the last decade, various MOEAs have been applied to solve different DEED problems to obtain a set on non-dominated solutions in a single run [21]. For example, Purkayastha et al. [38] applied NSGA-II with an adaptive crowding distance mechanism to improve diversity when solving a bi-objective DEED problem. Although this method was tested on a 40-unit test problem, the results were not compared with any others, even those of the traditional NSGA-II. Also, when a modified crowding distance approach was applied, solutions with lower crowding distances were rejected as survivors for the following generations, and although diversity was improved, the convergence rate decreased. To tackle this drawback, a niched Pareto GA with a clustering technique, in which the tournament size for the crossover was adaptively set, was developed in [1, 2]. This algorithm started with a large population, and the clustering technique was used to determine the manageable non-dominated solutions from the population. This method was tested on a 6-unit static problem, with the results indicating that it obtained a wider set of Pareto solutions than other state-of-the-art algorithms. Later, the same authors developed a MOPSO to solve the same problem [4]. This approach used the above mentioned clustering technique to update the set of non-dominated solutions. A comparison of the niched Pareto GA and MOPSO showed that the latter performed better in terms of the quality of the non-dominated solutions obtained. However, both were tested on only a small-scale problem with a single-hour load demand. Basu [10] developed a MODE for solving a large DEED problem, in which a non-dominated sorting technique [14] was used as a selection operator in a DE algorithm. It was the best of several other MOEAs, but since its parameters were arbitrarily set, it might not work well for other DEED problems. Recently, Liang [27] developed a multiobjective hybrid bat algorithm for the large-scale DEED problems, in which a learning-based non-dominated sorting archive was used. In it, a learning algorithm was employed to determine the global and local best for the algorithm to use in the next stage. Although they obtained very competitive results comparing to other algorithms, they solved only a few deterministic problems. Later, the same authors developed another variant of the hybrid bat algorithm [28] for solving single-objective, uncertain, economic dispatch problems.

Multi-objective mixed DEED problems, such as wind-thermal, solar-thermal and hydro-thermal ones, have not been fully studied in the literature. Recently, a few attempts have been made to solve mixed bi-objective DEED problems using MOEAs, such as a NSGA-II and MODE for hydro-thermal [41, 9], and MOPSO for wind/solar-thermal [11, 17]. Although these methods have the capability to generate all the trade-off solutions in a single run, in order for

them to handle a large number of equality constraints, they require extensive computational time, and some authors have simplified the DEED formulations by ignoring ramp constraints. Furthermore, in most of the above methods, the uncertainties of the renewable generations and the forecasted load demand, are ignored. Although this simplification of a DEED problem makes it easier to solve, the applicability of the method is not guaranteed.

Recently, a few researchers have incorporated some uncertainties into a DEED model in which the problem is formulated as a scenario-based probabilistic DEED one [35, 12, 48]. The scenarios represent the stochastic behaviors of variable load demand and intermittent generation from renewable generators. Each scenario is solved using various optimization methods, such as MILP [42], branch and bound algorithm [15], 2m-point estimated method (PEM) [7], GA [48] and DE [51]. However in these methods, the quality of a solution depends on the given number of scenarios; the more scenarios, the better the quality of solutions and vice versa. As a result, they require a great deal of computational effort to produce good solutions [44].

Using an ensemble algorithm, the authors in [40, 16] developed a GA framework while considering multiple crossover operators, Elsayed et al. [18] developed an evolutionary framework using multi-operator of GA and DE, and Zaman et. al. [47] a GA-DE framework for solving a broad range of single objective economic dispatch problems. All these multi-operator approaches consider two or more algorithms, each of which uses different methods to share information among the considered algorithms. Nevertheless, we believe that solving an uncertain bi-objective DEED problem using such a multi-method-based approach has not yet been tested.

3. Background of DEED Problems

Generally, the DEED problem considers two objectives to simultaneously minimize the fuel costs and gas emissions of the given power plants while satisfying their equality and inequality constraints. However, with considering the uncertainties of forecasted load demand and renewable generations, the problem to be modeled as an uncertain optimization problem. The notation to specify uncertain parameters (say, X) is that of adding a tilde above the variable, i.e., $\tilde{X}$. Details of the mathematical formulations of all these hydro-, wind and solar-thermal DEED problems are described below.

3.1. Hydro-thermal

In this section, the formulation of a bi-objective uncertain hydro-thermal DEED problem is described. The objectives and constraints are shown below.

3.1.1. Objective Functions

In a hydro-thermal system, the objectives are to simultaneously minimize both the operating cost and gas emission, defined as follows [47]:

$$\text{Min: } \mathbb{E}\left\langle F_c \right\rangle = \sum_{s=1}^{n} \sum_{t=1}^{T} \sum_{i=1}^{N_T} \left(a_i + b_i P_{T_{i,t,s}} + c_i P_{T_{i,t,s}}^2 + \left| d_i \sin \left\{ e_i \left(P_{T_{i,t}}^{\min} - P_{T_{i,t,s}} \right) \right\} \right| \right) \tag{1}$$

$$\text{Min: } \mathbb{E}\left\langle F_E \right\rangle = \sum_{s=1}^{n} \sum_{t=1}^{T} \sum_{i=1}^{N_T} \left(10^{-2} \left(\alpha_i + \beta_i P_{T_{i,t,s}} + \gamma_i P_{T_{i,t,s}}^2 \right) + \eta_i e^{\lambda_i P_{T_{i,t,s}}} \right), \forall i, t, s \tag{2}$$

The first (Eqn. (1)) and second (Eqn. (2)) objectives represent the expected fuel costs and gas emissions, respectively. $P_{T_{i,t,s}}$ is the i^{th} thermal power plant at t^{th} time period of an operational cycle T in s^{th} scenario. Note that, the uncertainties of renewable generations and forecasted load demand to be represented using a number of possible scenarios, which are generated using a normal distribution in which its mean and standard deviation are taken from historical data (see subsection 4.3 for details). N_T is the number of thermal power plants, and $\mathbb{E}\left\langle F_c \right\rangle$ and $\mathbb{E}\left\langle F_E \right\rangle$ are their expected fuel costs and gas emissions, respectively. a_i, b_i, c_i, d_i, e_i are the cost coefficients, and α_i, β_i, γ_i the emission coefficients of the i^{th} thermal generator.

The cost function considers the valve point effect of the thermal power plants, which adds a sinusoidal term to the basic quadratic cost function. The importance of adding this to the cost function is discussed in previous research [51]. The emission function is for the atmospheric pollutants, as SO_2 and NO_2 caused for the fossil-fuel-based-thermal power plants which modeled in previous research [45].

3.1.2. Constraints

In the hydro-thermal DEED problem, a number of equality and inequality constraints are considered, as shown below:

$$\sum_{i=1}^{N_T} P_{T_{i,t,s}} + \sum_{j=1}^{N_H} P_{H_{j,t,s}} = \tilde{P}_{D_{t,s}} \; t \in T, \; s \in n \tag{3}$$

$$P_{H_{j,t,s}} = C_{1,j} V_{j,t,s}^2 + C_{2,j} X_{j,t,s}^2 + C_{3,j} V_{j,t,s} X_{j,t,s} + C_{4,j} V_{j,t,s} + C_{5,j} X_{j,t,s} + C_{6,j} \; j \in N_H, \; t \in T, \; s \in n \tag{4}$$

$$V_{j,t+1,s} = V_{j,t,s} - X_{j,t,s} + I_{j,t,s} - S_{j,t,s} + \sum_{r=1}^{N_{up}} \left(X_{r,(t-t_{drj}),s} + S_{r,(t-t_{dr,j}),s} \right), \; j \in N_H, \; s \in n \tag{5}$$

$$P_{H_j}^{\min} \leq P_{H_{j,t,s}} \leq P_{H_j}^{\max} \; j \in N_H, \; t \in T, \; s \in n \tag{6}$$

$$P_{T_i}^{\min} \leq P_{T_{i,t,s}} \leq P_{T_i}^{\max} \; i \in N_T, \; t \in T, \; s \in n \tag{7}$$

$$V_{H_j}^{\min} \leq V_{H_{j,t,s}} \leq V_{H_j}^{\max}\ j \in N_H,\ t \in T,\ s \in n \tag{8}$$

$$X_{H_j}^{\min} \leq X_{H_{j,t,s}} \leq X_{H_j}^{\max}\ j \in N_H,\ t \in T,\ s \in n \tag{9}$$

$$|V_{j,t,s}|^{t=0} = V_j^{ini},\ \text{and}\ |V_{j,t,s}|^{t=T} = V_j^{end}\ j \in N_H,\ s \in n \tag{10}$$

Eqn. (3) represents the power balance constraint, in which the forecasted demand is considered as uncertain and is represented in different random scenarios as $\tilde{P}_{D_{t,s}}$, which is the power demand at the t^{th} time period in the s^{th} scenario. The hydro power generation of the j^{th} hydro plant at the t^{th} time interval in the s^{th} scenario is $P_{H_{j,t,s}}$, water storage rate $X_{j,t,s}$, and volume $V_{j,t,s}$. N_H is the number of hydro power plants and $C_{k,j}(k = 1, 2, \ldots, 6)$ are its generation coefficients, $I_{j,t,s}$, N_{up} and $S_{j,t,s}$ are the water inflow rate, number of upstream plants, spillage water (set to zero, as in [9]). $t_{d_{r,j}}$is the transportation delay of water from r^{th} to j^{th} reservoir. The capacity limits of the hydro and thermal plants are represented in Eqns. (6) to (9), in which the minimum and maximum output power of the hydro generations, water-discharge rates and -storage volumes are $P_{H_{j,t,s}}^{min}$ and $P_{H_{j,t,s}}^{max}$, $X_{j,t,s}^{min}$ and $X_{j,t,s}^{max}$ and $V_{j,t,s}^{min}$ and $V_{j,t,s}^{max}$, respectively. Eqn. (10) is the initial and final reservoir storage volumes, where, V_j^{ini} and V_j^{end} are the initial and final water-volumes of the j^{th} reservoir respectively.

3.2. Solar-thermal

Based on the nature of the decision variables, the solar-thermal DEED problem is represented as a mixed-integer non-linear optimization problem, in which a thermal unit is represented as a continuous variable and a solar unit as a binary one [25]. Furthermore, the generation from solar sources and the forecasted load demand, are considered as uncertain and are represented using different random scenarios. The details of the formulation are discussed below.

3.2.1. Objective Functions

The objective functions of an uncertain bi-objective solar-thermal DEED problem are defined as:

$$\begin{gathered}\text{Min: } \mathbb{E}\left\langle F_C \right\rangle = \sum_{s=1}^{n} \sum_{t=1}^{T} \left(\sum_{i=1}^{N_T} \left(F_{c_{i,s}}(P_{T_{i,t,s}}) \right) + \right. \\ \left. \sum_{k=1}^{N_S} \left(F_{S_{k,s}}(U_{S_{k,t,s}}) \right) \right)\end{gathered} \tag{11}$$

$$\begin{gathered}\text{where, } F_{c_{i,s}}(P_{T_{i,t,s}}) = a_i + b_i P_{T_{i,t,s}} + c_i P_{T_{i,t,s}}^2 \\ + \left| d_i \sin \left\{ e_i \left(P_{T_{i,t}}^{\min} - P_{T_{i,t,s}} \right) \right\} \right|\end{gathered} \tag{12}$$

$$\begin{gathered}F_{S_{k,s}}(U_{s_{k,t,s}}) = P U_{\cos t_k} \tilde{P}_{S_{k,t,s}} U_{s_{k,t,s}},\ U_{S_{k,t,s}} \in \{0, 1\} \\ k \in N_S\ t \in T,\ s \in n\end{gathered} \tag{13}$$

$$\tilde{P}_{S_{k,t,s}} = P_{r_k} \left\{ 1 + \Omega \left(\tilde{T}_{amb_{k,t,s}} - T_{ref_k} \right) \right\} \frac{Si_{k,t,s}}{1000} \tag{14}$$

$$\text{Min: } \mathbb{E}\left\langle F_E \right\rangle = \sum_{s=1}^{n}\sum_{t=1}^{T}\sum_{i=1}^{N_T} h_i \left(F_{E_{i,s}}(P_{T_{i,t,s}})\right) \tag{15}$$
$$= \sum_{s=1}^{n}\sum_{t=1}^{T}\sum_{i=1}^{N_T} \left(\frac{F_{c_{i,s}}\left(P_i^{\max}\right)}{F_{E_{i,s}}\left(P_i^{\max}\right)}\right)\left(F_{E_{i,s}}(P_{T_{i,t,s}})\right)$$

$$F_{E_{i,s}}(P_{T_{i,t,s}}) = \alpha_i + \beta_i P_{T_{i,t,s}} + \gamma_i P^2_{T_{i,t,s}} + \eta_i e^{\lambda_i P_{T_{i,t,s}}} \; i \in N_T \, t \in T \, s \in n \tag{16}$$

Eqn. (11) shows the first objective function that associates with the operational costs of both thermal and solar generators, and Eqn. (15) is the second objective function, which corresponds to the gas emissions from the considered thermal plants. Note that, the gas emission function is normalized to make a same order as the cost function. $\tilde{P}_{S_{k,t,s}}$ is the solar power generation which is considered as uncertain and its operation cost is represented in Eqn. (13), in which $U_{S_{k,t,s}}$ is a binary variable indicates the k^{th} unit is turned on or off in the s^{th} scenario at the t^{th} time interval, and PU_{cost_k} is its per unit cost. The available solar generation $\left(P_{S_{k,t,s}}\right)$ of the k^{th} unit at the t^{th} time interval in the s^{th} scenario is determined using Eqn. (14), where P_{r_k} is its rated power, T_{ref_k} and $T_{amb_{k,t,s}}$ are the reference temperature and ambient temperature respectively, the temperature coefficient is Ω, and $Si_{k,t,s}$ is the incident solar radiation.

3.2.2. Constraints

The following equality and inequality constraints are considered for the solar-thermal DEED problem.

$$\sum_{i=1}^{N_T} P_{T_{i,t,s}} + \sum_{k=1}^{N_S} \tilde{P}_{S_{k,t,s}} U_{S_{k,t,s}} = \tilde{P}_{D_{t,s}}, \; t \in T, \; s \in n \tag{17}$$

$$P_{T_i}^{\min} \leq P_{T_{i,t,s}} \leq P_{T_i}^{\max} \; i \in N_T, \; t \in T, \; s \in n \tag{18}$$

$$-DR_i \leq P_{T_{i,t,s}} - P_{T_{i,t-1,s}} \leq UR_i \;\; i \in N_T \, t \in T, \; s \in n \tag{19}$$

Eqn. (17) represents the power balance equality constraints, in which $\tilde{P}_{D_{t,s}}$ and $\tilde{P}_{S_{k,t,s}}$ are considered as uncertain parameters and are represented in the form of random scenarios. The capacity limits of the thermal generators are shown in Eqns. (18), while their ramp limits are in (19), in which UR and DR are the upward and downward transition rates, respectively.

3.3. Wind-Thermal

In this section, the mathematical model of a wind-thermal DEED problem is discussed.

3.3.1. Objective Function

The objectives of a wind-thermal DEED problem are to simultaneously minimize the operating cost and gas emission by utilizing the thermal and wind generators over a planning horizon of T hours. The power generated by a wind generator, and its forecasted load demand, are considered as uncertain, and vary within a range. The objective functions are defined as [37]:

$$\begin{aligned}\text{Min: } \mathbb{E}\langle F_C\rangle = &\sum_{s=1}^{N_S}\sum_{t=1}^{T}\sum_{i=1}^{N_T} F_{c_i}\left(P_{T_{i,t,s}}\right) + \sum_{s=1}^{N_S}\sum_{t=1}^{T}\sum_{w=1}^{N_W}\Big(\\ &F_{W\,w}\left(\tilde{P}_{W_{w,t,s}}\right) + F_{U_w}\left(\tilde{P}_{W_{w,t,s}}\right) + F_{O_w}\left(\tilde{P}_{W_{w,t,s}}\right)\Big)\end{aligned} \tag{20}$$

$$\begin{aligned}\text{Min: } \mathbb{E}\left\langle F_E\right\rangle = &\textstyle\sum_{s=1}^{N_S}\sum_{t=1}^{T}\sum_{i=1}^{N_T}\Big(10^{-2}\left(\alpha_i + \beta_i P_{T_{i,t,s}}\right. \\ &\left.+\gamma_i P^2_{T_{i,t,s}}\right) + \eta_i e^{\lambda_i P_{T_{i,t,s}}}\Big)\end{aligned} \tag{21}$$

where N_W is the number of wind power plants, $P_{T_{i,t,s}}$ and $\tilde{P}_{W_{w,t,s}}$ are the output of the i^{th} and w^{th} thermal and wind generator, respectively, in the s^{th} scenario in which F_{c_i} is the fuel cost of the i^{th} thermal generator (as in Eqn. (12)), while F_{W_w} is the operating cost, F_{U_w} and F_{O_w} are the penalty costs for the under- and over-estimated wind energy, respectively. Eqn. (21) is the cost for gas emissions where α_i, β_i, γ_i, η_i and λ_i are the emission coefficients of the i^{th} thermal generator, respectively. A liner function is used to represent the operating cost of wind generators, as:

$$F_{W\,w}(\tilde{P}_{W_{w,t,s}}) = \delta_w \tilde{P}_{W_{w,t}},\ w \in N_W\, t \in T\, s \in N_S \tag{22}$$

where δ_w is the per unit cost of the w^{th} wind generator, with its output at the t^{th} time interval expressed as [23]:

$$\tilde{P}_{W_{w,t,s}} = \begin{cases} 0 & \text{if } v_{out_w} < \tilde{v}_{w,t,s} < v_{in_w} \\ P_{R_w}\frac{\tilde{v}_{w,t,s} - v_{in_w}}{v_{r_w} - v_{in_w}} & \text{if } v_{in_w} < \tilde{v}_{w,t,s} < v_{r_w} \\ P_{R_w} & \text{if } v_{r_w} < \tilde{v}_{w,t,s} < v_{out_w} \end{cases} \tag{23}$$

where v_{out_w}, v_{in_w}, v_{r_w} and $\tilde{v}_{w,t,s}$ are the cut-out, cut-in, rated and t^{th}-hour wind speed of the w^{th} wind farm in the s^{th} scenario, respectively, and P_{R_w} is the rated wind power from the w^{th} wind generator.

Furthermore, we include penalty costs for any forecasted wind farm being under- or over-estimated, which are expressed as [47]:

$$F_{U_w}(\tilde{P}_{W_{w,t,s}}) = k_{U_w}\int_{P_{W_{w,t,s}}}^{P_{R_w}}\left(w - \tilde{P}_{W_{w,t,s}}\right) f_{P_{W_{w,t,s}}}(w)dw \tag{24}$$

$$F_{O\,w}(\tilde{P}_{W_{w,t,s}}) = k_{O_w}\int_{0}^{P_{R_w}}\left(\tilde{P}_{W_{w,t,s}} - w\right) f_{P_{W_{w,t,s}}}(w)dw \tag{25}$$

Earlier research shows that the wind speed follows Weibull distribution function, as [47]:

$$f_{P_{W_{w,t}}}(W) = \frac{K_t l v_{in}}{c_t} \phi^{K_t - 1} e^{-\phi^{K_t}},\ 0 < W_t < W_R \tag{26}$$

where the constants k_t, c_t and ϕ are determined as:

$$K_t = \left(\sigma_t / \mu_t\right)^{-1.086}, \tag{27}$$

$$c_t = \frac{\mu_t}{\Gamma(1 + K_t^{-1})} \tag{28}$$

$$\phi = \frac{(1 + (W/W_R)l)v}{c_t} \tag{29}$$

$$\text{where } l = \frac{v_r - v_{in}}{v_{in}} \tag{30}$$

where μ_t and σ_t are the mean and standard deviations of the wind speed at the t^{th} hour, respectively.

3.3.2. Constraints

The load demand, capacity and ramp constraints are considered in a wind-thermal DEED problem, as:

$$\sum_{i=1}^{N_T} P_{T_{i,t,s}} + \sum_{w=1}^{N_W} \tilde{P}_{W_{w,t,s}} = \tilde{P}_{D_{t,s}}\ s \in N_S \tag{31}$$

$$P_{T_i}^{\min} \leq P_{T_{i,t,s}} \leq P_{T_i}^{\max}\ i \in N_T,\ t \in T\ s \in N_S \tag{32}$$

$$0 \leq \tilde{P}_{W_{w,t,s}} \leq P_{R_w}\ w \in N_W,\ t \in T\ s \in N_S \tag{33}$$

$$-DR_i \leq \left(P_{i,t,s} - P_{i,(t-1),s}\right) \leq UR_i, \text{ if } P_{i,(t-1),s} > P_i^{\min} \tag{34}$$

$$-DR_i^0 \leq \left|P_{i,t,s} - P_{i,(t-1),s}\right| \leq UR_i^1, \text{ if } 0 < P_{i,(t-1),s} < P_i^{\min} \tag{35}$$

$$\begin{aligned} \left[T_{i,(t-1),s}^{on} - T_{\min_i}^{on}\right]\left[U_{T_{i,(t-1),s}} - U_{T_{i,t,s}}\right] \geq 0 \\ \left[T_{i,(t-1),s}^{off} - T_{\min_i}^{off}\right]\left[U_{T_{i,t,s}} - U_{T_{i,(t-1),s}}\right] \geq 0 \end{aligned} \tag{36}$$

Eqn. (34) shows the conventional ramp limits between two consecutive hours, and Eqn. (35) is the initial/final ramp limits when a generating unit is in the process of startup or shutdown, in which UR^1 and DR^0 are the initial ramp up and down respectively. Eqn. (36) represents the minimum on/off time

of a thermal generator, where $T^{on}_{min_i}$ and $T^{off}_{max_i}$ are the minimum on and off time of the i^{th} unit, respectively. $T^{on}_{i,(t-1),s}$ and $T^{off}_{i,(t-1),s}$ are the continuous on and off time of the i^{th} unit at the t^{th} time interval in the s^{th} scenario, respectively, and $U_{T_{i,t,s}}$ are the operational status of the i^{th} thermal unit at the t^{th} time interval in the s^{th} scenario, i.e., 0 - unit off, 1 - unit on.

4. Proposed Approach

In this research, we design an MMBA for solving a wide range of bi-objective optimization problems. In the design, we consider three efficient population-based algorithms (NSGA-II, MODE, and MOPSO), as they have shown good performance in solving many real-world optimization problems, particularly, bi-objective DEED problems [4, 50]. Each algorithm uses its own sub-population and each sub-population size is updated self-adaptively during the evolutionary process. An overview of the proposed framework is discussed below.

4.1. Overview of MMBA

The MMBA starts with N_P random individuals, generated using Eqn. (41). Then, their corresponding f values and CV are evaluated, based on the problems objective functions and constraints. For solving real-life DEED problems, which have a number of complex constraints, a heuristic was used to improve the solution quality, as described in subsection 4.9. Then, all the individuals are randomly distributed under three sub-populations of equal size, *i.e.*, $N_{P_1} = N_{P_2} = N_{P_3}$, where N_{P_1}, N_{P_2} and N_{P_3} are the sub-population size for NSGA-II, MODE and MOPSO, respectively. In the subsequent generations, all these three algorithms generate N_{P_1}, N_{P_2} and N_{P_3} new individuals. In this evolutionary step, each algorithm uses random individuals from the whole population (N_P) instead of its own individuals. The reason for this is to exchange useful information from other good individuals, that might have been generated by the other algorithms.

Once the new individuals are generated, the new values of f and CV are evaluated. As uncertainty may exist, the expected values of f and CV are determined, that is $n-$number of random scenarios of the renewable production and load demand are generated. Then, each individual from each sub-population is evaluated, based on the generated scenarios. Subsequently the average of these values is considered the final expected value. More details are given in sub-section 4.4. In the cases when we solve a problem without any uncertainty, f and CV are directly calculated from the problem's objective function and constraints.

After that, a non-dominated sorting approach [14] is applied to both parents and offspring, with the best individuals being selected for the next generation. Subsequently, the values of N_{P_1}, N_{P_2} and N_{P_3} are updated, based on their performance in previous generations, i.e., based-on the success rate (SR) of the generating successful offspring from their parents, as [47]:

$$SR = \frac{\text{Number of offspring selected for the next generation}}{\text{Total number of offspring (i.e., population size)}} \tag{37}$$

A higher value of SR of an algorithm, indicates that higher numbers of successful offspring were found after a generation. To calculate N_{P_1}, N_{P_2} and N_{P_3}, the normalized SR (NSR) is used, as [47]:

$$N_{P_i} = \max\left[N_P^{min}, \min\left\{N_P^{max}, NSR_i \times SR_{i,k}\right\}\right], i \in 3 \tag{38}$$

where

$$NSR_i = \frac{SR_{i,k} + \Delta_i}{\sum_{i=1}^{3}\left(SR_{i,k} + \Delta_i\right)} \tag{39}$$

and N_P^{min} and N_P^{max} are the minimum and maximum sub-population sizes respectively, and $SR_{i,k}$ is the SR of the i^{th} algorithm at the k^{th} generation. The values of $N_{P_i}(i = 1, 2, 3)$ remain the same when all $SR_{i,k}(i = 1, 2, 3)$ are zero. However, if one is a very small value and the others zero, there is a possibility to overly bias towards one algorithm. To avoid such a situation, a small $\Delta_i(i = 1, 2, 3)$ (here, we set it to 0.001) is added to all $SR_{i,k}(i = 1, 2, 3)$ [47].

This process is continued until a cycle is completed. A cycle (we call it as N_{G_C}) is a predetermined number that allows the algorithm to update N_{P_1}, N_{P_2} and N_{P_3} using Eqn. (38). Once a cycle is completed, the average SR (ASR) of each algorithm is determined, as:

$$ASR_i = \text{average}\left\langle SR_{i,k}\right\rangle k = cycle_{start}, \ldots, (cycle_{start} + N_{G_c} - 1), i \in 3 \tag{40}$$

where $cycle_{start}$ is the particular generation number when a cycle has been started. Based on the maximum ASR, only the best algorithm is performed in the next cycle, in which the algorithm considers all the updated N_P individuals, while the other two algorithms remain off until the cycle is finished. Afterwards, the updated individuals are randomly and equally distributed to all three algorithms, which restart to perform for the next cycle, with the sizes of their sub-populations updated using Eqn. (38). Once the cycle is completed, the best algorithm again performs in the next cycle. The process is continued until the algorithm reaches to its maximum number of generations (N_G). A flowchart of the proposed algorithm is shown in A.12 in Appendix, while its Pseudo-code is discussed in Algorithm 1. The details operators of the proposed algorithm are described in the subsequent subsections.

4.2. Initialization and Encoding

The MMBA algorithm starts with a random initial population, and each individual ($\vec{x_i}$) is encoded, as:

$$\vec{x_i} = \vec{x}_{min} + (\vec{x}_{max} - \vec{x}_{min})\, LHS(N_P) \forall \in N_P \tag{41}$$

Algorithm 1 Pseudo-code of the proposed MMBA algorithm

Require: N_G, N_P, N_P^{min} and N_P^{max}.

Set, $count_1 = count_2 = 0$.
Randomly generate initial population, as in subsection 4.2.
Evaluate the f and CV of each individual, as discussed in Algorithm 3.
Randomly distribute N_P individuals in three subpopulations with sizes of N_{P_1}, N_{P_2} and N_{P_3} (i.e., $N_{P_1} = N_{P_2} = N_{P_3}$) for GA, DE and PSO, respectively.
for $k = 1 : N_G$ **do**
    if $count_1 \leq N_{G_C}$ **then**
        Set, $count_1 = count_1 + 1$
        Generate N_{P_1}, N_{P_2} and N_{P_3} offspring from all parents (i.e., N_P) using NSGA-II, MODE, and MOPSO respectively.
        Repeat step 3 for all N_{P_1}, N_{P_2} and N_{P_3}.
        Select best individuals based on a non-dominated approach described in subsection 4.8.
        Update the best N_P individuals, based on the selected individuals, as: $N_P \leftarrow N_{P_1} + N_{P_2} + N_{P_3}$.
        Calculate $SR_{i,k}, i = 1, 2, 3$ using Eqn. (37)
        Update $N_{P_i} \forall i$, based on normalized SR, using Eqns. (38) and (39).
    else
        Set, $count_2 = count_2 + 1$
        if $count_2 \leq N_{gc}$ **then**
            if $Count_2 = 1$ **then**
                Calculate average SR (ASR) of all three algorithms using Eqn. (40).
                Determine the best algorithm, based on the maximum ASR.
            end if
            switch Best Algorithm **do**
                case GA
                    Generate new individuals using GA operators while considering, $N_{P_1} \leftarrow N_{P_1} + N_{P_2} + N_{P_3}$.
                case DE
                    Generate new individuals using DE operators while considering, $N_{P_2} \leftarrow N_{P_1} + N_{P_2} + N_{P_3}$.
                case PSO
                    Generate new individuals using PSO operators with considering, $N_{P_3} \leftarrow N_{P_1} + N_{P_2} + N_{P_3}$.
        end if
        if $count_2 \neq N_{G_C}$ **then**
            Perform the best algorithm by repeating steps 3 and 10.
        else
            Repeat from step 4, and set again, $count_1 = count_2 = 0$.
        end if
    end if
end for

Table 1: $\vec{x_i}$ and N_x for DEED problems

	Hydro-thermal	Wind-thermal	Solar thermal
$\vec{x_i}$	$P_{T_{i,t,}}$, $X_{j,t}$	$P_{T_{i,t,}}$, $P_{W_{w,t,s}}$	$P_{T_{i,t,}}$, $U_{s_{k,t,s}}$
N_x	$(N_T + N_H) \times T$	$(N_T + N_W) \times T$	$(N_T + N_S) \times T$

where $\vec{x_i} = \left\{x_i^1, x_i^2, ..., x_i^D\right\}$, D is the number of decision variables, $\vec{x}_{min} = \left\{x_{min}^1, x_{min}^2, ..., x_{min}^D\right\}$ and $\vec{x}_{max} = \left\{x_{max}^1, x_{max}^2, ..., x_{max}^D\right\}$ the lower and upper limit vectors, respectively. N_P is the population size, and LHS means that $\vec{x_i}$ is generated using the Latin Hypercube Sampling (LHS). LHS ensures that the generated samples are equally distributed over the search space.

The actual decision variables and their limits depend on the considered optimization problem. For example, Table 1 shows $\vec{x}$ and N_x for different DEED problems, where $i = 1, 2, \ldots, N_T$, $j = 1, 2, \ldots, N_H$, $w = 1, 2, \ldots, N_W$, $k = 1, 2, \ldots, N_S$, $t = 1, 2, \ldots, N_T$ and $s = 1, 2, \ldots, n$.

4.3. Uncertainty Considerations

In this research, the intermittent generation of the renewable sources (i.e., wind and solar) and forecasted load demand, are represented as n -number of random scenarios. As previously mentioned, a scenario represents a potential occurrence of an uncertain variable. For example, we want to schedule two thermal and one renewable generators to meet a certain load. The per unit cost of the thermal power plants is higher than that of the renewable one. However, the output of a renewable source is uncertain, which means a forecasted output with a standard deviation (σ), are given. Assume that the capacities of the two thermal power plants are 200 to 500 MW, the forecasted output of the renewable sources is 50 MW with $\sigma = 5$ MW. If the electricity demand at a certain hour is 500 MW, what would be the optimal decision from each power plant with minimizing the overall cost. Generally, when the renewable output is maximum, the cost is minimum, i.e., the optimal solution would be 50 from renewable and 450 from the two thermal plants. Nevertheless, the risk would be maximum, as there is a chance for the renewable source's output to deviate. A common approach is to consider the worst case for the uncertain variable. However, the worst-case approach is usually over-conservative. Therefore, the scenario-based-approach is very popular. In it, the uncertain variables (renewable and load demand for this study) are represented with a certain number of possible scenarios, which are generated using a normal distribution with the mean values (μ) and σ of their forecasted values and historical errors, respectively [25]. Then, $n-$number of problem instance are assumed for $n-$scenarios, with the f for each problem instance individually calculated. Finally, the solution of the optimization problem is determined from the expected f values, that is determined after averaging all fitness values obtained from the problem instances. A solution is said to be better if its expected f is minimum and consistent with other possible scenarios. Therefore, when the expected f of an individual is minimum, even after considering a higher number of scenarios, it is more likely to be stable for other

scenarios. However, selecting the number of scenarios is a trade-off between solution stability and computational time (further discussed in section 5.3.1). Earlier research demonstrated that the generated scenarios must be covered by following a $3\sigma-$rule, which is computationally burden for a large but practical renewable-based DEED system [34]. Therefore, the proposed algorithm considers a lower number of scenarios at the initial stage of the evolution, as most solutions are either infeasible and/or far away from the global optimum. Nevertheless, to increase the solution stability at the later stages of the evolution, the algorithm increases the number of scenarios over the generations, as:

$$n = \min\left[n^{max}, n^{min}\left(1 + \text{floor}\left\lfloor k/r \right\rfloor\right)\right] \tag{42}$$

where n^{min} and n^{max} are the minimum and maximum scenarios size, k the current generation number and r the period of generations to increment n. In other words, the algorithm runs $r-$ generations with the same number of scenarios to obtain a sub-optimal solution.

Algorithm 2 shows the process of generating $n-$scenarios, in which the value of n is calculated using Eqn. (42). Note that for simplicity, we assume the uncertainties of wind and solar sources are in their power outputs instead of their wind speeds and solar irradiance, respectively [46]. Also, the scenario generation scheme does not generate any infeasible scenarios that are impossible to solve by an algorithm.

Algorithm 2 Scenario Generation Approach

Require: μ, σ, and $T \triangleright T$ is the number of time periods of intermittent generations.
  for $t = 1 : T$ **do**
    for $r = 1 : R$ **do** $\triangleright R$ is the number of uncertain variables that to be represented in the form of scenarios.
      Get corresponding $(\mu^w_{r,t}, \sigma^w_{r,t})$, $(\mu^s_{r,t}, \sigma^s_{r,t})$ and $\mu^d_{r,t}$, $\sigma^d_{r,t}$ from the historical data of wind speed, solar irradiance and forecasted load demand, respectively.
      Generate $n-$ random variables using a normal distribution, as: $P_{S_{r,t,s}} = \mathbf{N}\left(\mu^s_{r,t}, \sigma^s_{r,t}\right)$, and $P_{D_{t,s}} = \mathbf{N}\left(\mu^d_{r,t}, \sigma^d_{r,t}\right)$; $s = 1, 2, ..., n$.
    end for
  end for

4.4. Evaluations

As mentioned before, the DEED problem has two objectives, 1) minimize the cost function and 2) minimize gas emission. When the uncertainties are included, the objective values are not deterministic, but probabilistic. In this paper, the f of a DEED problem is determined based on its objective functions, while the CV is calculated as:

$$CV_i = \sum_{p=1}^{P} \max\left(0, G_p\left(\overrightarrow{x_i}\right)\right) + \sum_{q=1}^{Q} \max\left(0, H_q\left(\overrightarrow{x_i}\right)\right) \tag{43}$$

where $\overrightarrow{x_i}$ are the decision variables of the i^{th} individual. The inequality and equality constraints are G and H respectively, while P and Q are their numbers, respectively. The procedure to calculate G and H of a DEED problem can be found in Appendix B. After calculating all G and H, if $CV = 0$, the individual is called feasible, otherwise it is infeasible.

For the uncertain DEED problem, a solution ($\overrightarrow{x}_i$) is considered infeasible, if it does not satisfy all the n scenarios considered in any generation, where the total CV for $\overrightarrow{x}_i$ is the sum of the CVs over all scenarios, as:

$$CV_i = \sum_{s=1}^{n} CV_{i,s} \forall i \in N_P \tag{44}$$

where $CV_{i,s}$ is the CV of the i^{th} individual for the s^{th} scenario. Algorithm 3 shows the detailed process to calculate f_i and CV_i of an individual, $\vec{x}_i, i \in N_P$ under uncertain conditions, with its flowchart given in Fig. A.13.

Algorithm 3 Evaluating the f and CV of an individual ($\vec{x}_i, i \in N_P$) under $n-$scenarios

Require: An individual, $\vec{x}_i, i \in N_P$

1: Set,$f_s = \{\}$ and $CV_{i,s} = \{\}$, are the fitness values and constraints violations for s^{th} scenario, respectively.
2: **for** $s = 1 : n$ **do**
3: Calculate, $CV_{i,s}$ of the i^{th} individual using Eqn. (43) with considering s^{th} scenario.
4: **if** $CV_{i,s} \neq 0$ **then**
5: Repair the $\vec{x}_i$ using the heuristic described in subsection 4.9, with considering the parameters for s^{th} scenario.
Update, $CV_{i,s}$.
6: **end if**
7: Calculate f_s of $\vec{x}_i$ with s^{th} scenario, as described in Section 3 and Eqn. (43).
8: **end for**
9: **Return** the expected fitness values, $f = \frac{\sum_{s=1}^{n} f_s}{n}$ and total constraints violations, $CV_i = \sum_{s=1}^{n} CV_{i,s}$.

Algorithm 3 starts with $n-$scenarios and each individual in the population is assessed based on all scenarios. First, $CV_{i,s}$ of the individual with the s^{th} scenario is calculated, based on Eqn. (43). If the solution is not feasible, i.e., $CV_{i,s} \neq 0$, the heuristic is used to repair it as much as possible, i.e., reducing $CV_{i,s}$ to reach 0 if possible (Algorithm 3, lines 2-6). Then, the corresponding fitness value is calculated (line 7). Once the i^{th} individual is evaluated on all scenarios, its expected (average) fitness and total CV_i are returned (line 9).

4.5. GA Search Operators

GA is a population-based meta-heuristic algorithm that has two main search operators, known as crossover and mutation. During crossover, a new offspring

is generated by exchanging chromosomes of its parents. Mutation helps to diversify the individuals in a population. Of the many search operators, simulated-binary-crossover (SBX) and non-uniform mutation (NUM) have shown excellent performance when solving real-value optimization problems [47]. In this research, they are considered and are briefly discussed below.

In SBX, firstly two parents are selected using a tournament-based selection scheme, in which a parent is randomly chosen from two based on their f and CV. Then, the two children are generated as:

$$\vec{y}_p^1 = 0.5\big[(1+\beta_{qp})\vec{x}_p^1 + (1-\beta_{qp})\vec{x}_p^2\big] \tag{45}$$

$$\vec{y}_p^2 = 0.5\big[(1-\beta_{qp})\vec{x}_p^1 + (1+\beta_{qp})\vec{x}_p^2\big] \tag{46}$$

where

$$\beta_{qp} = \begin{cases} \big(2u_p\big)^{1/\eta_c+1} & u_p \le 0.5, \\ \Big(\frac{1}{2(1-u_p)}\Big)^{1/\eta_c+1} & u_p > 0.5 \end{cases} \tag{47}$$

$u_p \in [0,1]$ is a uniform random number and η_c a distribution factor.

Non-uniform mutation, which has shown admirable performance for constrained optimization problems [51], is used to maintain diversity in the population. A mutated individual ($\vec{y}'_p$) is obtained from its original $\vec{y}_p$ in a generation k, as:

$$y'_{p,j,k+1} = y_{p,j,k} + \delta_{p,j}, j = 1, 2, \ldots, D \tag{48}$$

$$\delta_{p,j} = \begin{cases} \big(x_p^{max} - y_{p,j,k}\big)\Big(1 - [u]^{1-(k/N_G)^b}\Big) & u \le 0.5, \\ \big(x_p^{mpn} - y_{p,j,k}\big)\Big(1 - [u]^{1-(k/N_G)^b}\Big) & u > 0.5 \end{cases} \tag{49}$$

$$\forall p \in N_P \quad \text{and} \quad j \in D$$

where b is the speed of the step length, and it is set to 5, as in [47].

4.6. DE Search Operators

DE is another variant of EA which also has two search operators: mutation and crossover. In this research, we use two mutation operators: '$DE/rand/1$' and '$DE/rand-to-best/1$', and a binomial crossover, to ensure a good balance between diversity and convergence, as these operators have found to show good search capabilities [18]. At each generation, new individuals ($\vec{y}_{p,k}$) are generated from their parent ($\vec{x}_{p,k}$), as:

$$\vec{y}_{p,k+1} = \begin{cases} \vec{x}_{r_3,k} + F_p\big(\vec{x}_{r_1,k} - \vec{x}_{r_2,k}\big), \\ \qquad \text{if} \quad rand_1 \le Cr_p \text{ and } rand_2 \le prob_1, \\ \vec{x}_{p,k} + F_p\big((\vec{x}_{r_1,k} - \vec{x}_{r_2,k}) + (\vec{x}_{best,t} - \vec{x}_{p,k})\big), \\ \qquad \text{if} \quad rand_1 \le Cr_p \text{ and } rand_2 > prob_1, \\ \vec{x}_{p,k}, \quad \text{otherwise} \end{cases} \tag{50}$$

where r_1, r_2, and r_3 are three random integer values, such that $p \neq r_1 \neq r_2 \neq r_3$, the amplification factor for the mutation operator is F_P, the crossover rate is Cr_p and $prob_1$ is a predefined probability of choosing the mutation operator (here set to a value of 0.5).

The performance of DE depends highly on its control parameters, *i.e.*, the values of F_p and Cr_p [51]. To obtain the best performance from the DE, an adaptive mechanism for setting the values of F_P and Cr_P is used [48]. In it, they are initially generated using a normal distribution with a mean and standard deviation 0.5 and 0.1, respectively, *i.e.*, $\{\dot{F}, \dot{Cr}\} \in N(0.5, 0.1)$ [20]. In the following generations, they are updated as:

$$F_p = \begin{cases} \dot{F}_{r_1} + rand_1(\dot{F}_{r_2} - \dot{F}_{r_3}), \text{ if } (rand_2 < \tau_1), \\ rand_3, \text{ otherwise} \end{cases} \tag{51}$$

$$Cr_p = \begin{cases} \dot{Cr}_{r_1} + rand_4(\dot{Cr}_{r_2} - \dot{Cr}_{r_3}), \text{ if } (rand_5 < \tau_1), \\ rand_6, \text{ otherwise} \end{cases} \tag{52}$$

where $\tau_1 = 0.75$ and $rand_k \in [0, 1], k = 1, 2, .., 6$ [20]. However, the values of both F_p and Cr_p must lie between 0.1 and 1. So, if one is less than 0.1 or greater than 1, it is fixed to 0.1 and 1, respectively. Once the best individuals are selected for the next generation using the selection operator (discussed in subsection (4.8)), the corresponding best set of F_p and Cr_p are also selected.

4.7. PSO Search Operators

The PSO method is also a population-based optimization technique, and has been used for a wide range of real-world problems [13]. It is simple in structure with its individuals being called particles, which have a position defined by $\vec{x} = \{x_p^1, x_p^2, \ldots, x_p^D\}$, and velocity, defined by $\vec{v} = \{v_p^1, v_p^2, \ldots, v_p^D\}$. At each generation, $\vec{v}$ and $\vec{x}$ are updated as [13]:

$$\vec{v}_{k+1} = w\vec{v_k} + c_1 r_1 \left(\vec{P}_k - \vec{x}_k\right) + c_2 r_2 \left(\vec{G}_k - \vec{x}_k\right) \tag{53}$$

$$\vec{x}_{k+1} = \vec{x}_k + \chi \vec{v}_{k+1} \tag{54}$$

where w, c_1, c_2, $\chi \geq 0$. The inertia of a particle is w; c_1 and c_2 are two constants that set in the velocity towards the local $\left(\vec{P}\right)$ and the global best $\left(\vec{G}\right)$, χ is a shift factor (here, set to 1) to update the overall position, and r_1 and r_2 are two random values set between 0 and 1. Selecting the values of the inertia factor (w) is important when updating a particle, as it should be set dynamically to maintain a balance between convergence and diversity, as:

$$w = w_{max} - (w_{max} - w_{min}) \times \frac{k}{N_G} \tag{55}$$

where w_{max} and w_{min} are the maximum and minimum values of w, which are set to 0.9 and 0.4, respectively.

4.8. Selection Operator

In order to facilitate the bi-objective approach, the solutions of each generation are sorted using the well-known non-dominated and crowding distance (CD) mechanism [14]. In it, the individuals are ranked by following one of three criteria: (a) all solutions are infeasible (i.e., $CV_i \neq 0, i = 1, 2, \ldots, N_P$), (b) some of the them are feasible and the rest are infeasible, and (c) all solutions are feasible (i.e., $CV_i = 0, i = 1, 2, \ldots, N_P$). For (a), the individuals with lower CVs are selected, regardless of their f values, for (b), the feasible individuals are always preferred over infeasible ones, and for (c), the individuals are sorted by examining their f and CD neighboring solutions, *i.e.*, solutions with the same f but that are non-crowded, are preferred over those with in a crowded area, as per Deb's [14] crowding rules. For uncertain bi-objective problems, both parents and children are evaluated under the same n scenarios, and their mean f and total CV are retained, as discussed in Algorithm 3. Then they are sorted, based on their f, CV, and CD, as discussed above.

4.9. Heuristic

As mentioned, the bi-objective DEED problem is a constrained complex optimization problem with a number of complex equality and inequality constraints. The individuals generated using an EA may not satisfy all the constraints, particularly the demand (equality type) ones. Therefore, the convergence rate of an EA method is very poor while solving a DEED problem [51]. The situation becomes worse when the load demand to be satisfied by considering the intermittent generation from renewable sources. Therefore, a heuristic is used to rectify an infeasible individual to a feasible one, or at least reduces its constraint violation (meaning coming closer to the feasible space). In it, the $T-$hour DEED problem is decomposed into $T-$hourly sub-problems. Then, the available productions is reallocated to the generators using a slack generation approach. The details of the heuristic can be found in our earlier research work [47].

5. Experimental Test

The performance of the proposed MMBA method is demonstrated by solving six standard test problems and three real-world bi-objective DEED problems, for a 24-hour planning horizon with one-hour time interval, such as for a hydro-thermal, wind-thermal and solar-thermal DEED. Furthermore, the uncertainties in renewable productions and forecasted load demand are considered in the DEED formulations. The problems are described as follows:

Benchmarks: six bi-objective test problems, such as Schaffer, Kursawe, ZDT1, ZDT2, ZDT3, and ZDT6 without uncertainties;

Case-Ia: seven-unit hydro-thermal DEED problems without uncertainties;

Case-Ib: seven-unit hydro-thermal DEED problems with uncertainties in load demand;

Case-IIa: seven-unit wind-thermal DEED problems without uncertainties;

Case-IIb: seven-unit wind-thermal DEED problems with uncertainties in load demand and solar productions;

Case-IIIa: nineteen-unit solar-thermal DEED problems without uncertainties;

Case-IIIb: nineteen-unit solar-thermal DEED problems with uncertainties in load demand and solar productions;

The uncertainties of forecasted load demand, wind and solar productions are represented with $n-$ scenarios in which the value of n is dynamically updated over generations using Eqn. (42), with the values of r, n^{min} and n^{max} being set to 10, 10 and 100, respectively. All the test problems are solved using the following algorithms:

- Decomposition-based EA (DBEA) [6]
- NSGA-II
- MODE
- MOPSO
- Proposed MMBA

As our earlier research [45] showed that algorithms without a heuristic perform very poorly when solving DEED problems, all the above algorithms consider the heuristic described in Section 4.9. Each algorithm is run 30 times and their feasible non-dominated solutions over $n-$scenarios are reported and compared with results from state-of-the-arts algorithms. The parameters of GA such as, crossover and mutation probabilities are set to 0.9 and 0.1, respectively, and the distribution index (η) is set to 3. The PSO parameters are set, as $c_1 = c_2 = 2$, $w = 0.8$ and mutation rate of 0.1. The DE parameters are set self-adaptively with the mutation and crossover rates varying within the range of 0.1 to 1.0.

All the algorithms are executed on an Intel Core i7 3.4 GHZ desktop computer with 16 GB of RAM in a Matlab (R2017a) environment. Based on the empirical experiments and our earlier research [47], we set the values of N_P, N_G, N_P^{min}, N_P^{max} and N_{G_C} of each problem, as listed in Table 2. Note that while solving the uncertain DEED problems (Case-Ib, -IIb and -IIIb), we set smaller values of N_P and N_G to reduce the computational burden.

To validate the obtained results from MMBA, all the DEED problems were also solved using a recent algorithm for many-objective optimization problems, namely decomposition-based evolutionary algorithm (DBEA) [6]. The code for DBEA was taken from an online source where all the parameters are set as default, and the maximum number of generations is calculated as: $(N_P \times N_G)/13$, so that all algorithms use the same number of FFEs. Note that, 13 is the reference point which used to solved all DEED problems.

Table 2: Different parameters used in different problems

Problem	N_P	N_G	N_P^{min}	N_P^{max}	N_{G_C}
Case-Ia	200	500	40	160	20
Case-Ib	60	100	12	48	
Case-IIa	100	200	20	80	
Case-IIb	100	200	20	80	
Case-IIIa	200	500	40	160	
Case-IIIb	60	100	12	48	

Table 3: Comparison of the performance of algorithms for the benchmark problem

Alg.	Average HV indicator (ref.: [1,1])					
	MMBA	NSGA-II	MODE	MOPSO	DBEA	MOPSO* [13]
Schaffer	0.83	0.83	0.83	0.83	0.79	0.83
Kursawe	0.40	0.40	0.40	0.40	0.36	0.40
ZDT1	0.67	0.66	0.66	0.66	0.59	0.67
ZDT2	0.33	0.33	0.33	0.33	0.26	0.33
ZDT3	0.52	0.51	0.52	0.52	0.43	0.52
ZDT6	**0.41**	0.26	0.38	0.38	0.32	0.40

5.1. Solving bi-objective benchmark problems

In this section, to verify the efficiency of our proposed MMBA, standard bi-objective test problems ('Schaffer', 'Kursawe', 'ZDT1', 'ZDT2', 'ZDT3', and 'ZDT6') [13], which have known Pareto-frontiers, are solved. The results obtained from the proposed algorithm are compared with the state-of-the-art algorithms. For a fair comparison, we use the same problem data, and N_P and N_G values, as those in [13]. All the problems are solved 30 times. To measure the performance of each algorithm, the hyper-volume (HV) values are calculated based on their normalized fitness values as [45]:

$$f_{norm} = \frac{f - f_{ideal}}{f_{Nadir} - f_{ideal}} \tag{56}$$

where f_{norm} and f are normalized and actual fitness values, respectively, while f_{ideal}, and f_{Nadir} are the ideal and nadir points respectively [26].

Based on the values of HV, reported in Table 3, it is found that the proposed MMBA outperforms the state-of-the-art ones for 'ZDT6', while it obtains the same solutions for all other problems. It is also noted that the values of HV for MOPSO* are calculated based on the given online results [30].

5.2. Real-world Bi-objective Optimization Problems

In this subsection, some of the real-world bi-objective optimization problems, such as hydro-thermal, wind-thermal and solar-thermal DEED problems, are considered. In addition, uncertainties of these problems are incorporated into their formulations and they are solved using the proposed MMBA and the state-of-the-art algorithms.

Table 4: Comparison of the performance of algorithms for the hydro-thermal DEED problem

	Alg.	HV indicator (ref.: [1,1])				Time
		Best	Avg.	Worst	σ	(min.)
Case-Ia	DBEA	0.78	0.78	0.78	**0.00**	**3.58**
	NSGA-II	0.86	0.82	0.79	0.03	8.59
	MODE	0.71	0.61	0.52	0.07	5.94
	MOPSO	0.60	0.46	0.32	0.11	6.70
	MMBA	**0.91**	**0.88**	**0.85**	0.02	7.62
Case-Ib	DBEA	0.61	0.53	0.21	0.18	**15.25**
	NSGA-II	0.74	0.62	0.52	0.08	18.33
	MODE	0.73	0.63	**0.57**	0.06	17.38
	MOPSO	0.37	0.30	0.21	0.06	20.11
	MMBA	**0.77**	**0.63**	0.38	**0.05**	17.99

5.2.1. Hydro-Thermal DEED Problems

The hydro-thermal DEED problem is comprised with three thermal and four hydro units [45]. Two cases (Case-Ia and Case-Ib) are considered for this problem and are solved using the NSGA-II, MODE, MOPSO and MMBA on the same platform. In Case-Ib, the uncertainty of the load demand is considered and presented with upto 100 random scenarios, in which the mean values are assumed as the forecasted load demand and the errors are the 10% of the mean values. To reduce the computational time, a smaller number of scenarios are considered at the early stages of evolution, and to enhance solution stability, a larger number of scenarios are used at the later stages of the evolutionary process. The process of dynamically generating the number of scenarios is discussed in Section 4.3.

Both cases are solved thirty times using the above five algorithms, and their best results are recorded. Then, their hyper-volume (HV) values of their obtained solutions are determined based on their normalized values, as shown in Eq (56). The minimum, maximum, average and σ of the HV values, and the computational time of the algorithms in both cases, are shown in Table 4, with the best values shown in **boldface**. It is seen that the proposed MMBA shows superiority over the other algorithms. It provides very consistent results in a rational computational time in both cases. The Pareto fronts of all the algorithms, based on the median runs, are shown in Figs. 1 and 2 for Case-Ia and Case-Ib respectively, in which it is seen that the proposed MMBA is the best algorithm, as it produces better and wider Pareto fronts.

5.2.2. Wind-Thermal DEED Problems

In this section, a seven-unit wind-thermal DEED problem without (*i.e.*, Case-IIa) and with (*i.e.*, Case-IIb) uncertainties of load demand and wind production are solved. The parameters of these two cases are illustrated in Table 5. The uncertainties are represented in the form of $n-$ scenarios in which the values of n are dynamically increased up to 100, with the mean and standard

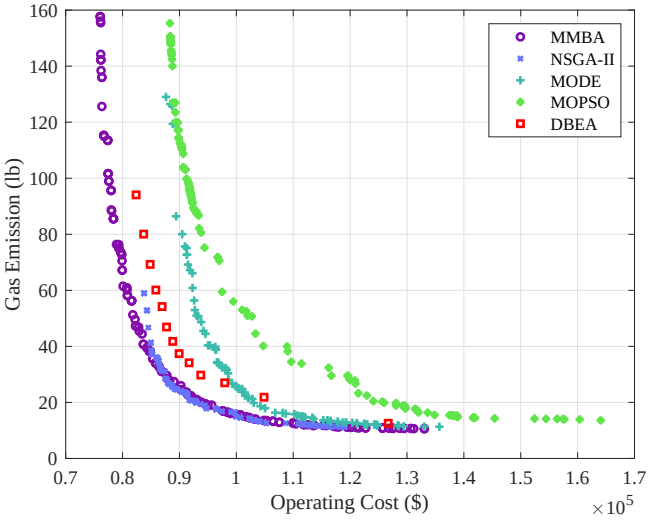

Fig. 1: Pareto fronts of the hydro-thermal problem (Case-Ia)

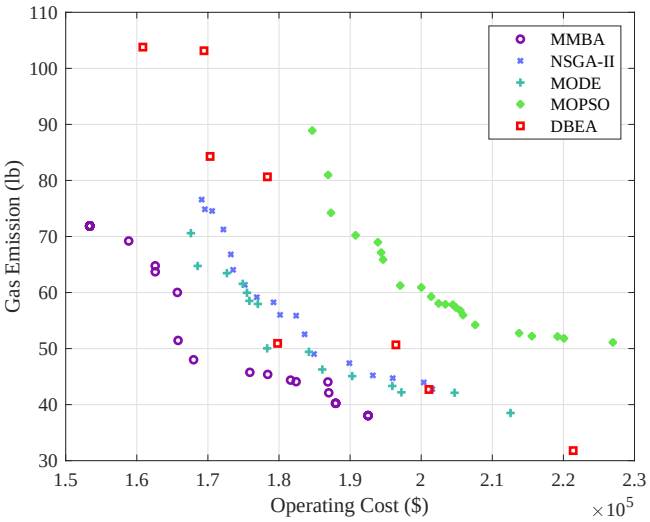

Fig. 2: Pareto fronts of the hydro-thermal problem (Case-Ib)

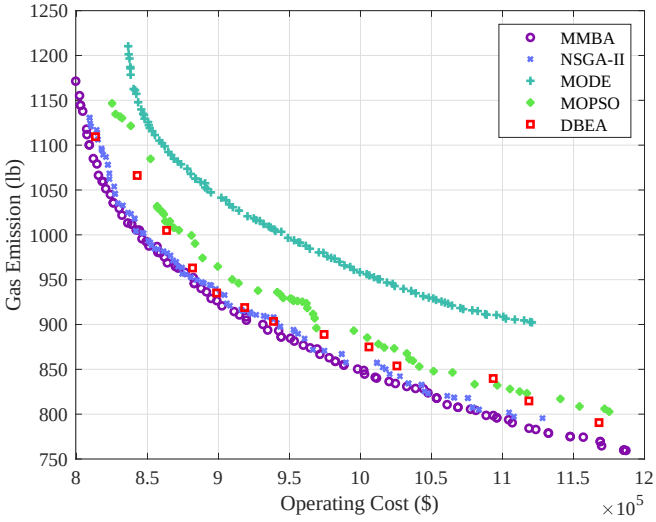

Fig. 3: Pareto fronts of the wind-thermal problem (Case-IIa)

deviations taken from the forecasted and historical errors, respectively. In this research, the historical errors are considered as 10% of the forecasted values [48].

The proposed MMBA and state-of-the-art algorithms are applied to solve both Case-IIa and Case-IIb, and their Pareto fronts are shown in Figs. 3 and 4, respectively. It is seen that the MMBA outperforms the other algorithms in terms of obtaining solutions of a higher quality. Eventually, MMBA obtains wider and better Pareto fronts than the state-of-the-art algorithms, in both the deterministic and the stochastic cases. Table 5 shows the statistics of the HV values and computational times of the algorithms in both cases. It is clear that MMBA obtained the best HV values (which are shown in **boldface**) in a reasonable computational time.

5.2.3. Solar-Thermal DEED Problems

In the mixed-integer solar-thermal DEED problem, 13-solar and 6-thermal units are considered from [45]. The problem is solved in two different cases: one without considering any uncertainty i.e., Case-IIIa; and another incorporating the uncertainties of solar generation and forecasted load demand, i.e., Case-IIIb. Both uncertain parameters (i.e., solar generation and load demand) are simultaneously represented by up to 100 random scenarios, with their mean values from forecasted solar production and load demand respectively, while their σ assume 50% of the forecasted solar irradiance and 10% of the load demand, respectively. The values of solar irradiance and load demand are taken from [49].

Both cases are solved using the state-of-the art and proposed algorithms with their Pareto fronts based on the median runs, are illustrated in Figs. 5 and

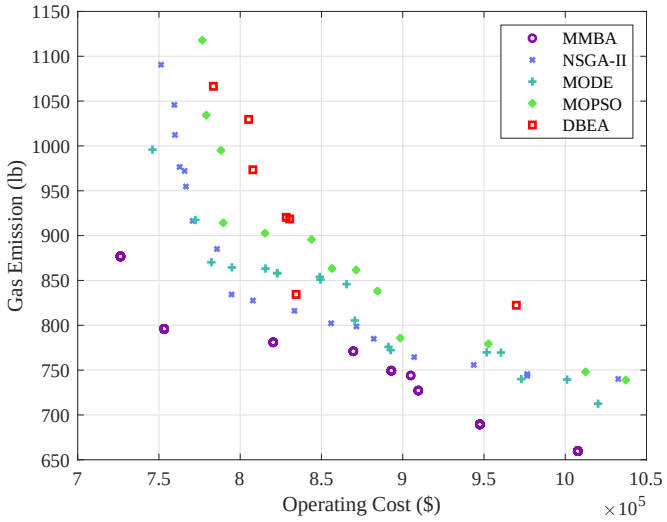

Fig. 4: Pareto fronts of the wind-thermal problem (Case-IIb)

Table 5: Comparison of the performance of algorithms for the wind-thermal DEED problem

	Alg.	HV indicator (ref.: [1,1])				Time (min.)
		Best	Avg.	Worst	σ	
Case-IIa	DBEA	0.42	0.37	0.35	0.03	**1.70**
	NSGA-II	0.73	0.72	0.70	0.01	3.17
	MODE	0.75	0.71	0.48	0.08	2.91
	MOPSO	0.67	0.65	0.63	0.01	2.84
	MMBA	**0.75**	**0.74**	**0.73**	**0.00**	2.73
Case-IIb	DBEA	0.52	0.51	0.51	0.02	**21.67**
	NSGA-II	0.32	0.30	0.26	0.03	48.31
	MODE	0.30	0.25	0.20	0.05	28.51
	MOPSO	0.19	0.15	0.11	0.04	27.91
	MMBA	**0.52**	**0.45**	**0.36**	**0.02**	40.94

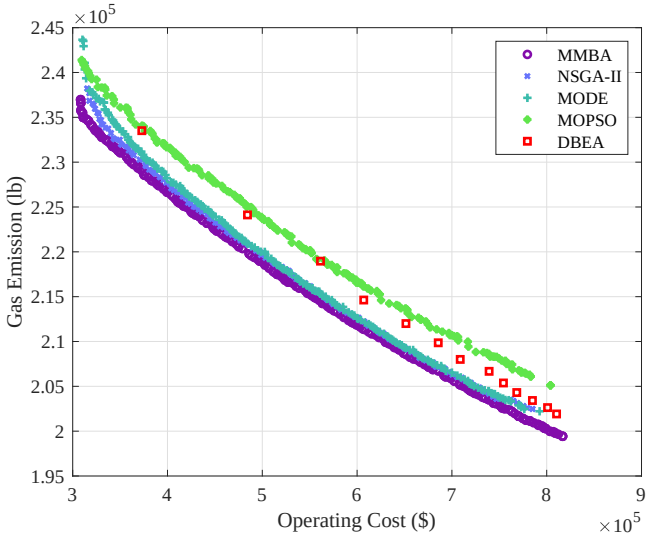

Fig. 5: Pareto fronts of the solar-thermal problem (Case-IIIa)

6 for Case-IIIa and Case-IIIb, respectively. Both figures show that the proposed MMBA is the best.

The results of the HV values for both cases are shown in Table 6, which shows that the proposed one obtains the maximum HV in both cases. This means the our proposed algorithm obtains a better spread set of solutions along the Pareto fronts as shown in Figs. 5 and 6 for Case-IIIa and Case-IIIb, respectively.

5.3. Discussion - Stochastic vs. Deterministic

In this section, we discuss the quality of the obtained solutions from MMBA under stochastic and deterministic approaches. We first discuss the effects of selecting the dynamic number of scenarios in the various stages of the evolutionary process, and then we analyze the effect of using per MW renewable sources and different approaches for evaluating f under uncertain environments.

5.3.1. Effect of the Dynamic Sizes of Scenarios

As mentioned, the uncertain DEED problems are represented as scenario-based probabilistic problems, and the scenario size (n) is dynamically increased with solution process. This method helps to significantly reduce computational time. In this section, we analyze the quality of solutions and the computational time required for different values of n. The quality of a solution is determined by evaluating its errors, as [39]:

$$error = z\frac{\Psi}{\sqrt{n}} \tag{57}$$

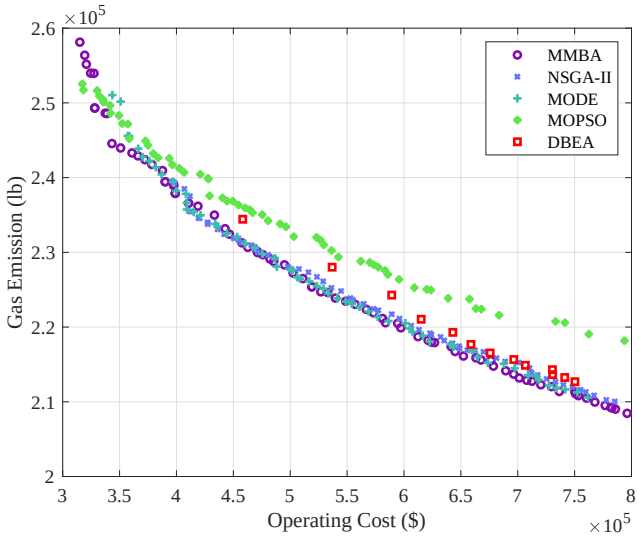

Fig. 6: Pareto fronts of the solar-thermal problem (Case-IIIb)

Table 6: Comparison of the performance of algorithms for solar-thermal DEED problem

	Alg.	HV indicator (ref.: [1,1])				Time (min.)
		Best	Avg.	Worst	σ	
Case-IIIa	DBEA	0.42	0.37	0.35	0.03	10.97
	NSGA-II	0.61	0.58	0.57	0.02	10.47
	MODE	0.60	0.59	0.57	0.02	**10.32**
	MOPSO	0.51	0.50	0.48	0.01	16.29
	MMBA	**0.62**	**0.59**	**0.58**	**0.01**	11.51
Case-IIIb	DBEA	0.53	0.52	0.50	0.02	**27.05**
	NSGA-II	0.66	0.64	0.61	0.02	38.73
	MODE	0.67	0.66	0.64	0.02	39.87
	MOPSO	0.64	0.59	0.58	0.02	40.14
	MMBA	**0.68**	**0.67**	**0.65**	**0.01**	39.77

Table 7: Errors and computational time for different n values

n	10	50	100	200	500	1000	Dynamic*
error	19.44	7.9	5.53	3.89	2.45	1.72	1.73
Time(min)	0.74	4.24	10.35	20.66	52.31	105.99	60.59

*Dynamic means the value of n is dynamically increased from 10 to 1000

Table 8: Errors and computational time for different r values

r	Error	Time	HV indicator (ref.: [1,1])			
			Best	Avg.	Worst	σ
5	2.21	42.43	0.43	0.42	0.39	0.03
10	**2.75**	**39.77**	**0.68**	**0.67**	**0.65**	**0.01**
20	2.81	36.37	0.66	0.64	0.61	0.03

where Ψ is the σ of the uncertain parameters of the generated scenarios, n, and z is a predefined value fora given confidence level. We set $z = 1.96$ for the confidence level of 95% [39]. The error value measures the quality of a solution; the smaller the error, the more accurate the output f of the considered scenarios.

Table 7 shows the values of errors and computational time for solving a sample solar-thermal problem (Case-IIIb) with different values of n. It is seen, that lower values of n produce a higher error of the obtained solutions. Conversely, higher values of n produce a lower error of the obtained solutions, but also results in a significantly higher computational time. On the other hand, when the value of n is dynamically increased, both the error and computational time are found to be satisfactory.

Now, we discuss the effect of the parameter r, which is used to update the value of n in Eqn. (42). Three different values of r are used to solve the solar-thermal DEED problem and their corresponding error, computational time (in minutes) and HV values are shown in Table 8. It is seen, that with increasing values of r, the errors increase but the computational time decreases. Furthermore, based on the values of HV, $r = 10$ is the best value.

5.3.2. Effect of Costs and Emissions to Use Renewable Energy

We analyze the values of costs and gas emissions of the test problems, under both stochastic and deterministic renewable production and load demand. From the above discussion of the simulation results shown in Table 7, scheduling generators using a lower number of scenarios, have a larger error value. In contrast, scheduling the generators using a deterministic approach considering a single forecasted scenario does not give an acceptable solution by itself [8]. Therefore, solving the problems using a dynamic scenario-based approach is vital.

Nevertheless, as the renewable productions and load demand are unmanageable and hard to forecast accurately, further production from the conventional thermal generators are taken to keep a balance between security and stability of the power network. This raises the total operating costs and gas emissions,

which can be seen from Figs. 1 to 6, where the expected f with uncertainties are higher than those without uncertainties. To demonstrate the percentage increase of both cost and gas emission, we evaluate two metrics as follows [8]:

- Average uncertainty costs (AUC): This matrix in $\$/MWh$ is the measure of the increase in the total electrical energy costs when an MW of uncertain renewable energy grows is scheduled:

$$AUC = \frac{\frac{\mathbb{E}\langle F_c\rangle_{min} - F_{c_{min}}}{\sum_{t=1}^{T}\sum_{ren=1}^{N_{ren}} P_{ren_{ren,t}}}}{T} \tag{58}$$

where $\mathbb{E}\langle F_c\rangle_{min}$ and $F_{c_{min}}$ are the minimum operating cost of stochastic and deterministic test problems respectively, P_{ren} is the power production from renewable energy, N_{ren} is the number of renewable sources. Note that the minimum costs are taken as extreme minimum costs from the corresponding Pareto fronts.

- Average uncertainty gas emissions (AUE): This matrix in lb/MWh indicates the changes of gas emissions when an additional MW of renewable energy is scheduled:

$$AUC = \frac{\frac{\mathbb{E}\langle F_E\rangle_{min} - F_{E_{min}}}{\sum_{t=1}^{T}\sum_{ren=1}^{N_{ren}} P_{ren_{ren,t}}}}{T} \tag{59}$$

where $\mathbb{E}\langle F_E\rangle_{min}$ and $F_{E_{min}}$ are the extreme minimum gas emissions of the Pareto fronts for the stochastic and deterministic test problems, respectively.

AUC and AUE are (5.98,0.0027) and (2.15,0.95) for the wind-thermal and solar-thermal problems respectively. It is seen that both metrics (AUC and AUE) are positive, but different for cost and emission, which indicate that their increasing rates vary with the per MW renewable energy usage.

5.3.3. Effect of Different Approaches to Evaluate f

As mentioned before, the two objective functions of the stochastic DEED problem are evaluated by determining the expected (e.g., average) values of cost and gas emission, respectively. However, the expected objective functions do not always give the true solution. Sometimes, the expected f result in increasing computational time. In this subsection, we analyze the Pareto fronts obtained from the MMBA by determining the f based on different approaches. So far, we evaluated the f of both operating cost and gas emission by determining their expected values i.e., the f of a median scenario of the given number of scenarios. Now, we solve the problems by evaluating the f based on three instances:

- Ins.-A: the f are evaluated for each individual, based on best scenario of a given number of scenarios;

- Ins.-B: the f are evaluated for each individual, based on the median scenario of a given number of scenarios;

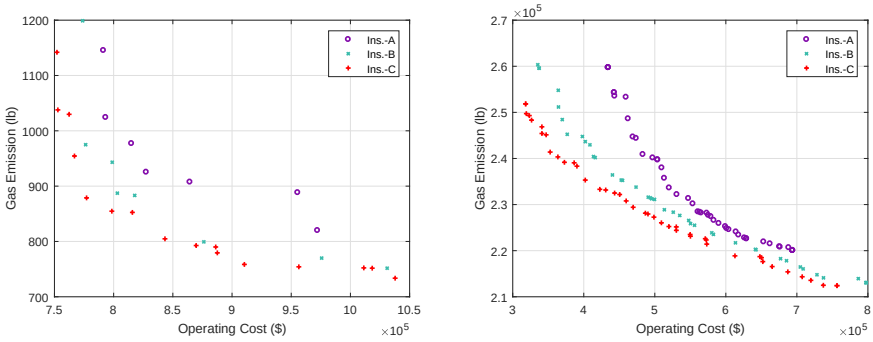

Fig. 7: Pareto fronts of wind thermal DEED (Case-IIb) and solar-thermal (Case-IIIb) problems (left to right) with $N_P = 60$ and $N_G = 100$.

- Ins.-C: the f are evaluated of each individual, based on the worst scenario of a given number of scenarios;

Fig. 7 shows the Pareto fronts of the stochastic wind- and solar-thermal problems, respectively when they are solved using the MMBA algorithm while considering the three different approaches to evaluate their f. It is seen that the final solutions widely varied with the f evaluation selection, i.e., when the f are evaluated based on Ins.-A, the algorithm obtains the best Pareto fronts. In addition, the Pareto fronts from the different approaches are not identically varied, i.e., their variations are different at different points. This is because when the algorithm starts with a worst solution, the quality of solutions is inferior, even in the final generation. On the other hand, when the algorithm starts with a best scenario, it quickly finds the best solutions which appeared in Fig. 7. However, if the maximum number of generations is increased in the case of worst scenario selection (i.e., Ins.-B and Ins.-C), the Pareto fronts improve, as shown in Fig. 8 for wind- and solar-thermal DEED problems, respectively. In this case, we allow the algorithm to evaluate up to 1000 generations for the worst case (i.e., Ins.-C), while it stops at 100 generations in the best case (i.e., Ins.-A). In conclusion, a stochastic DEED problem can be solved by evaluating its f in any of the three approaches. However, when the f are evaluated based on the worst scenario, the algorithm requires a large number of FFEs to obtain a Pareto front of good quality, whereas such front is found quickly when the f are evaluated based on the best scenario.

5.4. Statistical Tests

In this subsection, our proposed MMBA is statistically evaluated with state-of-the-art approaches. Based on the values of HV of different problems, two non-parametric tests (Wilcoxon sign test and Friedman rank test) are applied. Table 9 shows the results of the Wilcoxon tests, based on both the best and average values of HV found in 30 runs, with a 5% significance level for the above six cases. The last column shows a decision of the test with indicating

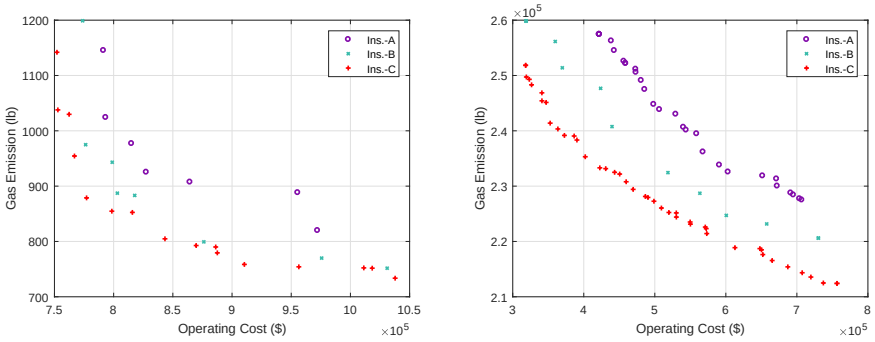

Fig. 8: Pareto fronts of wind thermal DEED (Case-IIb) and solar-thermal (Case-IIIb) problems (left to right) with $N_P = 60$ while $N_G = 100, 100$, and 1000 for Ins. A, B and C, respectively.

Table 9: Wilcoxon test results for MMBA versus DBEA, NSGA-II, MODE and MOPSO

Algorithms	Criterion	Better	Similar	Worse	p	Decision*
MMBA vs DBEA	Best HV	6	0	0	0.028	+
	Mean HV	6	0	0	0.028	+
MMBA vs NSGA-II	Best HV	6	0	0	0.027	+
	Mean HV	6	0	0	0.027	+
MMBA vs MODE	Best HV	5	1	0	0.043	+
	Mean HV	4	2	0	0.068	−
MMBA vs MOPSO	Best HV	6	0	0	0.028	+
	Mean HV	6	0	0	0.027	+

**The MMBA algorithm is statistically better (i.e.,'+') when $p < 0.05$*

three signs: ' + ' , ' − ' and ' ≡ '. When MMBA is significantly better than the other algorithms, it represents '+', that '-' when it is significantly worse and ' ≡ ' when there is no significant difference between algorithms. From Table 9, it is seen that the MMBA approach obtains consistently better results than the state-of-the-art approaches in both cases. In addition, based on the p values, it can be said that MMBA is statistically better than other algorithms. The Friedman test results are shown in Table 10, from which, it can be seen, that the proposed MMBA algorithm is ranked first, based on both best and mean HV.

Table 10: Ranks of DBEA, NSGA-II, MODE, MOPSO and MMBA from Friedman test results

Criteria	DBEA	NSGA-II	MODE	MOPSO	MMBA
Best HV	3.75	2.50	2.92	4.67	**1.17**
Mean HV	3.83	2.67	2.67	4.50	**1.33**

5.5. Effect of number of algorithms

In this subsection, we show the performance of each algorithm, considered in the framework of MMBA, during the solution process. In addition, the proposed framework is tested by including another population-based algorithm, namely bat algorithm for multi-objective optimization (MOBA) which is also known as an efficient algorithm for real-world problems [27, 28].

Firstly, the performances of all the considered algorithms in MMBA are demonstrated, based on their SR on generating successful offspring. As mentioned, MMBA considers three algorithms, and their sub-population sizes are dynamically changed, with the best algorithm run for a full cycle to evolve all individuals while the other two are kept off. In Appendix C, Figs C.14 to C.16 show the NSR (normalized success rate) for the first 200 generations of different algorithms for solving deterministic DEED problems, i.e., hydro-thermal (Case-Ia), wind-thermal (Case-IIa) and solar-thermal (Case-IIIa) ones. It is seen that MOPSO initially performs better, while MODE and NSGA-II perform better in the later stages of the search process. Therefore, it can be said that no single algorithm performs the best, over all generations, for solving the wide range of DEED problems considered in this paper, and their performance can change during the solution process. Therefore configuring the best algorithm during evolutionary process is the best option. However, in the above configuration, the performance of MOPSO is comparatively inferior than those for MODE and NSGA-II. Therefore, we solve all these three problems considering only MODE and MOPSO, in addition to adding another algorithm, as shown below:

- **var1**: it considers NSGA-II and MODE;
- **var2**: based on the proposal, it considers NSGA-II, MODE and MOPSO; and
- **var3**: it considers NSGA-II, MODE, MOPSO and MOBA.

The above three variants of the proposed framework are used to solve three deterministic DEED problems and their best solutions are shown in Figs. 9 to 11 for hydro-thermal (Case-Ia), wind-thermal (Case-IIa) and solar-thermal (Case-IIIa), respectively. It is found that the proposed framework with an additional algorithm (i.e., var3) does not increase solution quality. On the other hand, when MOPSO is removed from the framework (i.e., var1), it shows almost similar performance as that found from MMBA (i.e., var2) . However, MMBA slightly increases the quality of solutions.

The performances of the above three variants are shown in Table 11, in which the average HV in 30 runs, are reported. It is seen that the proposed MMBA of var2 is the best algorithm, based on the quality of solutions and reasonable computational time.

5.6. Feasibility Check

In this subsection, we report the performance of different algorithms in producing feasible non-dominated solutions for uncertain DEED problems. As

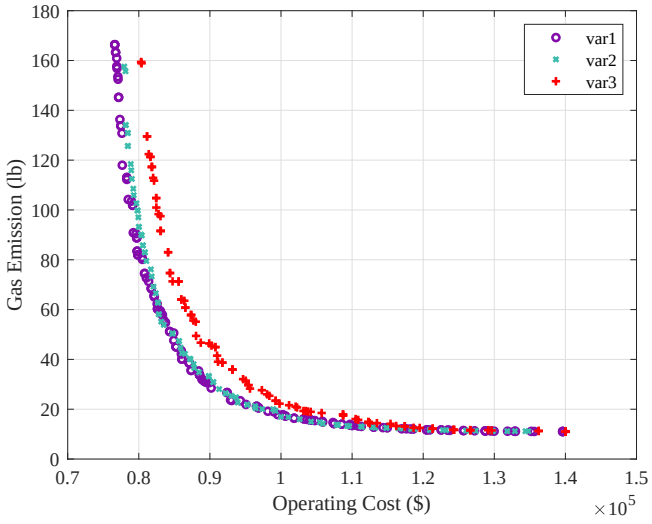

Fig. 9: Comparisons between the three frameworks for Case-Ia

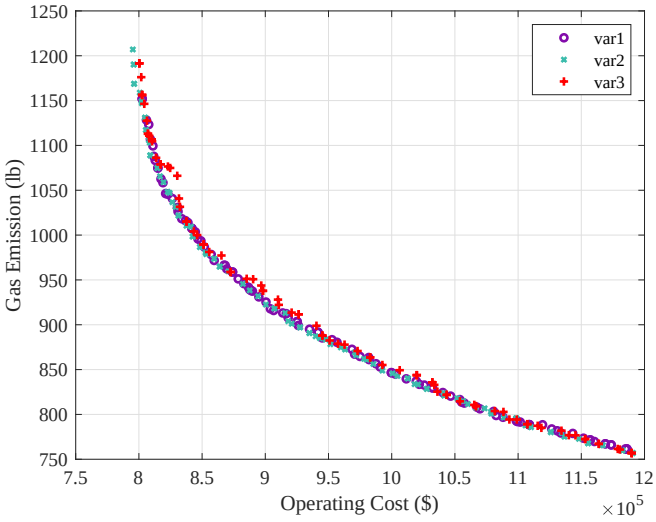

Fig. 10: Comparisons between the three frameworks for Case-IIa

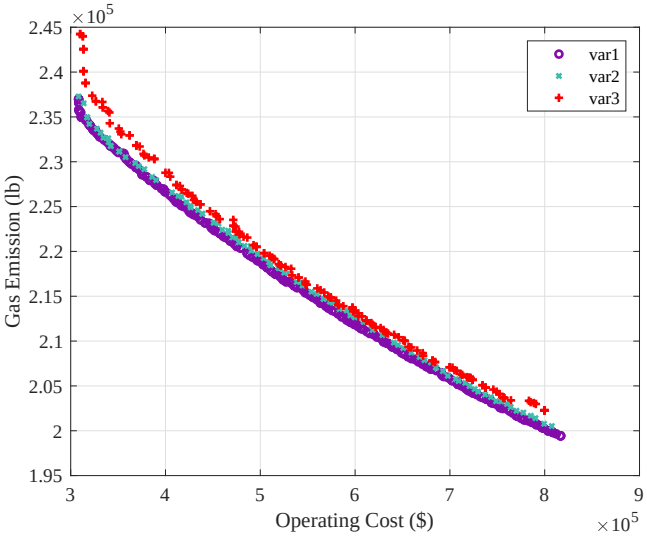

Fig. 11: Comparisons between the three frameworks for Case-IIIa

Table 11: Average HV and computational time for the three variants of the proposed framework

Problems	var1	var2	var3
hydro-thermal	0.8828	**0.9131**	0.8445
wind-thermal	0.7120	**0.7182**	0.7089
solar-thermal	0.5417	**0.5591**	0.4779
Time (sec.)	180.23	194.13	216.45

Table 12: Minimal, maximal and average rates of feasible non-dominated solutions over 30 runs for different algorithms

Algorithm	Hydro-thermal			Wind-thermal			Solar-thermal		
	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.
DBEA	0.32	0.62	0.43	0.62	0.62	0.62	1.00	1.00	1.00
NSGA-II	0.30	0.35	0.32	0.11	0.16	0.13	1.00	1.00	1.00
MODE	0.25	0.37	0.31	0.07	0.13	0.11	1.00	1.00	1.00
MOPSO	0.22	0.37	0.31	0.12	0.20	0.17	0.95	1.00	1.00
MMBA	0.39	0.78	0.62	0.96	0.97	0.97	1.00	1.00	1.00

previously mentioned, the f and CV values of individuals are calculated for $n-$scenarios, with an individual considered as feasible if it satisfies all the constraints over all $n-$scenarios, otherwise, it is recorded as infeasible.

Table 12 presents the minimum, maximum, and average feasibility rates of the non-dominated feasible solutions, over 30 runs, for all the algorithms considered in this research. Note that the number of non-dominated feasible solutions varies from generation to generation. The reported statistics is based on the final generation of the algorithms. As reported, MMBA obtained the highest numbers of feasible individuals, for solving all three problems, in comparison to those of other algorithms.

6. Conclusions and Recommendations for Future Research

In this paper, a MMBA for solving a wide range of benchmarks of deterministic and uncertain optimization problems, is developed. It contains three efficient population-based algorithms (NSGA-II, MODE and MOPSO), which are self-adaptively configured during the search process, based on their performance in previous generations. Although all three approaches initially consider the same number of individuals and evolve in their own sub-population, the sizes of these sub-populations vary dynamically and depend on their SRs for generating a better offspring from a parent. After a cycle, only the best algorithm of the three performs in the next cycle. Once that cycle is completed, the updated individuals are uniformly distributed to all the algorithms to perform for the following cycle. The process is repeated until it is terminated when a stopping criterion is met.

The performance of our proposed MMBA algorithm is illustrated by using it to solve six standard test problems and three real-world bi-objective DEED problems of which, each of them considers two cases, i.e., with and without uncertainties. In the uncertain DEED problem, both renewable generations and load demand are represented in the form of different random scenarios. The number of scenarios was dynamically increased over generations to keep a balance solution quality and computational time. The obtained results were analyzed with those from state-of-the-art algorithms, and the key outcomes found are as follows.

- For solving a wide range of bi-objective optimization problems, such as benchmarks, deterministic DEED and uncertain DEED problems, higher quality solutions are obtained with the proposed evolutionary framework, (i.e., MMBA-based-on NSGA-II, MODE and MOPSO) than with the state-of-the-art algorithms.
- For the deterministic DEED problems, the non-dominated solutions obtained by the MMBA are better, than the solutions from the state-of-the-art algorithms.
- For the uncertain DEED problems, scenarios of the uncertain parameters play an important role in obtaining a high quality solution. Higher number of scenarios gives more stable solutions but increases computational time. However, the proposed dynamic consideration of scenarios, helps to reduce the computational time, even after considering a higher number of scenarios.
- Statistic tests demonstrate that the solutions found from MMBA are significantly better than those from the state-of-the-art algorithms.
- For solving an optimization problem, it is found that an algorithm may perform better at early stages of evolution process, while another might be better in a later stage. It is also seen, that no single algorithm performs best throughout the generations.
- Therefore, combining them into a framework helps to obtain a high-quality solution. However, it does not always give benefits. For example, solutions are not significantly improved when three algorithms (i.e., NSGA-II, MODE and MOPSO) are considered instead of two, i.e, NSGA-II and MODE. In-fact, solution quality is degraded when the framework considers four algorithms, such as with NSGA-II, MODE, MOPSO and MOBA.

As part of our future work, we plan to improve the performance of the MMBA by proposing a method for exchanging information among the considered algorithms. Solving more uncertain problems by considering uncertainties of wind speed, solar irradiation and load demand over short time intervals, could be another possible research direction.

Acknowledgment

This research has been funded by Australian Research Council Discovery Project DP170102416.

References

References

[1] Abido, M. (2003a). Environmental/economic power dispatch using multiobjective evolutionary algorithms. *IEEE Transactions on Power Systems*, 18(4):1529–1537.

[2] Abido, M. (2003b). A novel multiobjective evolutionary algorithm for environmental/economic power dispatch. *Electric Power Systems Research*, 65:71 – 81.

[3] Abido, M. (2006). Multiobjective evolutionary algorithms for electric power dispatch problem. *IEEE Transactions on Evolutionary Computation,*, 10(3):315–329.

[4] Abido, M. (2009). Multiobjective particle swarm optimization for environmental/economic dispatch problem. *Electric Power Systems Research*, 79(7):1105 – 1113.

[5] Aghaei, J., Niknam, T., Azizipanah-Abarghooee, R., and Arroyo, J. M. (2013). Scenario-based dynamic economic emission dispatch considering load and wind power uncertainties. *International Journal of Electrical Power & Energy Systems*, 47:351–367.

[6] Asafuddoula, M., Ray, T., and Sarker, R. (2015). A decomposition-based evolutionary algorithm for many objective optimization. *IEEE Transactions on Evolutionary Computation*, 19(3):445–460.

[7] Azizipanah-Abarghooee, R., Niknam, T., Roosta, A., Malekpour, A., and Zare, M. (2012). Probabilistic multiobjective wind-thermal economic emission dispatch based on point estimated method. *Energy*, 37(1):322–335.

[8] Bahmani-Firouzi, B., Farjah, E., and Azizipanah-Abarghooee, R. (2013). An efficient scenario-based and fuzzy self-adaptive learning particle swarm optimization approach for dynamic economic emission dispatch considering load and wind power uncertainties. *Energy*, 50:232 – 244.

[9] Basu, M. (2010). Economic environmental dispatch of hydrothermal power system. *International Journal of Electrical Power & Energy Systems*, 32:711–720.

[10] Basu, M. (2011). Economic environmental dispatch using multi-objective differential evolution. *Applied Soft Computing*, 11(2):2845 – 2853. The Impact of Soft Computing for the Progress of Artificial Intelligence.

[11] Cai, J., Ma, X., Li, Q., Li, L., and Peng, H. (2009). A multi-objective chaotic particle swarm optimization for environmental/economic dispatch. *Energy Conversion and Management*, 50(5):1318 – 1325.

[12] Chakraborty, U. (2012). A new stochastic algorithm for proton exchange membrane fuel cell stack design optimization. *Journal of Power Sources*, 216:530 – 541.

[13] Coello, C. A. C., Pulido, G. T., and Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):256–279.

[14] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6:182–197.

[15] Ding, X., Lee, W.-J., Jianxue, W., and Liu, L. (2010). Studies on stochastic unit commitment formulation with flexible generating units. *Electric Power Systems Research*, 80:130–141.

[16] Eiben, A. E., Sprinkhuizen-Kuyper, I. G., and Thijssen, B. A. (1998). Competing crossovers in an adaptive ga framework. In *The IEEE International Conference on Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence.*,, pages 787–792.

[17] ElDesouky, A. (2013). Security and stochastic economic dispatch of power system including wind and solar resources with environmental consideration. *International Journal of Renewable Energy Research*, 3(4):951–958.

[18] Elsayed, S. M., Sarker, R. A., and Essam, D. L. (2011). Multi-operator based evolutionary algorithms for solving constrained optimization problems. *Computers & Operations Research*, 38:1877–1896.

[19] Elsayed, S. M., Sarker, R. A., and Essam, D. L. (2013). Adaptive configuration of evolutionary algorithms for constrained optimization. *Applied Mathematics and Computation*, 222:680–711.

[20] Elsayed, S. M., Sarker, R. A., and Essam, D. L. (2014). A new genetic algorithm for solving optimization problems. *Engineering Applications of Artificial Intelligence*, 27:57–69.

[21] Gonçalves, E., Balbo, A., da Silva, D., Nepomuceno, L., Baptista, E., and Soler, E. (2019). Deterministic approach for solving multi-objective non-smooth environmental and economic dispatch problem. *International Journal of Electrical Power and Energy Systems*, 104:880–897.

[22] Gong, W., Zhou, A., and Cai, Z. (2015). A multioperator search strategy based on cheap surrogate models for evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 19(5):746–758.

[23] Hetzer, J., Yu, D., and Bhattarai, K. (2008). An economic dispatch model incorporating wind power. *IEEE Transactions on Energy Conversion*, 23(2):603–611.

[24] Jin, Y. and Branke, J. (2005). Evolutionary optimization in uncertain environments-a survey. *IEEE Transactions on Evolutionary Computation*, 9(3):303–317.

[25] Khan, N. A., Awan, A. B., Mahmood, A., Razzaq, S., Zafar, A., and Sidhu, G. A. S. (2015). Combined emission economic dispatch of power system including solar photo voltaic generation. *Energy Conversion and Management*, 92:82 – 91.

[26] Kim, I. Y. and de Weck, O. L. (2006). Adaptive weighted sum method for multiobjective optimization: a new method for pareto front generation. *Structural and Multidisciplinary Optimization*, 31:105–116.

[27] Liang, H., Liu, Y., Li, F., and Shen, Y. (2018a). A multiobjective hybrid bat algorithm for combined economic/emission dispatch. *International Journal of Electrical Power and Energy Systems*, 101:103–115.

[28] Liang, H., Liu, Y., Shen, Y., Li, F., and Man, Y. (2018b). A hybrid bat algorithm for economic dispatch with random wind power. *IEEE Transactions on Power Systems*.

[29] Ma, H., Su, S., Simon, D., and Fei, M. (2015). Ensemble multi-objective biogeography-based optimization with application to automated warehouse scheduling. *Engineering Applications of Artificial Intelligence*, 44:79 – 90.

[30] Martínez, V. (2018). Multi-objective particle swarm optimization (mopso). *https://au.mathworks.com/matlabcentral/fileexchange/62074-multi-objective-particle-swarm-optimization-mopso*.

[31] Medhane, D. V. and Sangaiah, A. K. (2017). Search space-based multi-objective optimization evolutionary algorithm. *Computers & Electrical Engineering*, 58:126 – 143.

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

[33] Mohammadian, M., Lorestani, A., and Ardehali, M. (2018). Optimization of single and multi-areas economic dispatch problems based on evolutionary particle swarm optimization algorithm. *Energy*, 161:710 – 724.

[34] Nosair, H. and Bouffard, F. (2016). Economic dispatch under uncertainty: The probabilistic envelopes approach. *IEEE Transactions on Power Systems*, PP(99):1–1.

[35] OsArio, G., Lujano-Rojas, J., Matias, J., and CatalÃ£o, J. (2015). A probabilistic approach to solve the economic dispatch problem with intermittent renewable energy sources. *Energy*, 82:949 – 959.

[36] Pan, S., Jian, J., and Yang, L. (2018). A hybrid milp and ipm approach for dynamic economic dispatch with valve-point effects. *International Journal of Electrical Power and Energy Systems*, 97:290–298.

[37] Peng, C., Sun, H., Guo, J., and Liu, G. (2012). Dynamic economic dispatch for wind-thermal power system using a novel bi-population chaotic differential evolution algorithm. *International Journal of Electrical Power and Energy Systems*, 42(1):119–126.

[38] Purkayastha, B. and Sinha, N. (2010). Optimal combined economic and emission load dispatch using modified nsga ii with adaptive crowding distance. *International Journal of information Technology and Knowledge management*, 2(2):553–59.

[39] Rumsey, D. J. (2017). How to calculate a confidence interval for a population mean when you know its standard deviation. http://www.dummies.com/education/math/statistics/base-r-statistical-functions. (Accessed on 04/03/2017).

[40] Spears, W. M. (1995). Adapting crossover in evolutionary algorithms. In *Evolutionary programming*, pages 367–384.

[41] Tian, H., Yuan, X., Ji, B., and Chen, Z. (2014). Multi-objective optimization of short-term hydrothermal scheduling using non-dominated sorting gravitational search algorithm with chaotic mutation. *Energy Conversion and Management*, 81:504 – 519.

[42] Wang, Q., Guan, Y., and Wang, J. (2012). A chance-constrained two-stage stochastic program for unit commitment with uncertain wind power output. *IEEE Transactions on Power Systems,*, 27:206–215.

[43] Wu, G., Mallipeddi, R., and Suganthan, P. N. (2018). Ensemble strategies for population-based optimization algorithms - a survey. *Swarm and Evolutionary Computation*.

[44] Yu, H., Chung, C., Wong, K., Lee, H., and Zhang, J. (2009). Probabilistic load flow evaluation with hybrid latin hypercube sampling and cholesky decomposition. *IEEE Transactions on Power Systems,*, 24(2):661–667.

[45] Zaman, F., Elsayed, S. M., Ray, T., and Sarker, R. A. (2017a). *An Evolutionary Framework for Bi-objective Dynamic Economic and Environmental Dispatch Problems*, pages 495–508. Springer International Publishing, Cham.

[46] Zaman, F., Sarker, R., and Chang, G. (2017b). Dynamic scenario-based solution approach for scheduling solar-thermal generators. In *2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA 2017), Beijing, China*.

[47] Zaman, M., Elsayed, S., Ray, T., and Sarker, R. (2016a). Configuring two-algorithm-based evolutionary approach for solving dynamic economic dispatch problems. *Engineering Applications of Artificial Intelligence*, 53:105–125.

[48] Zaman, M., Elsayed, S. M., Ray, T., and Sarker, R. A. (2016b). Evolutionary algorithms for power generation planning with uncertain renewable energy. *Energy*, 112:pp. 408–419.

[49] Zaman, M. F., Elsayed, S., Ray, T., and Sarker, R. (2015). An evolutionary approach for scheduling solar-thermal power generation system. In *International Conference on Computers & Industrial Engineering (CIE45)*, volume 45, Metz, France,.

[50] Zaman, M. F., Elsayed, S. M., Ray, T., and Sarker, R. A. (2016c). A double action genetic algorithm for scheduling the wind-thermal generators. In Ray, T., Sarker, R., and Li, X., editors, *Artificial Life and Computational Intelligence: Second Australasian Conference, ACALCI 2016, Canberra, ACT, Australia, February 2-5, 2016, Proceedings*, pages 258–269. Springer International Publishing.

[51] Zaman, M. F., Elsayed, S. M., Ray, T., and Sarker, R. A. (2016d). Evolutionary algorithms for dynamic economic dispatch problems. *IEEE Transactions on Power Systems,*, 31:1486–1495.

Appendix A. Flowchart

For better understanding, algorithms 1 and 3 are shown in Figs. A.12 and A.13, respectively.

Appendix B. G and H for DEED problems

In this subsection, we show the process to determine values of G and H for the equality and inequality constraints, respectively, for the hydro-thermal, solar-thermal and wind-thermal DEED problems.

Appendix B.1. Hydro-thermal

To evaluate G and H, the constraints of the hydro-thermal DEED problem can be represented, as:

$$H\left(P_{T_{i,t,s}}, X_{j,t,s}\right) = \left| \left(\sum_{i=1}^{N_T} P_{T_{i,t,s}} + \sum_{j=1}^{N_H} P_{H_{j,t,s}} \right) - \tilde{P}_{D_{t,s}} \right| - tolerance \; t \in T, \; s \in n \tag{B.1}$$

$$G_1\left(P_{T_{i,t,s}}, X_{j,t,s}\right) = P_{H_j}^{\min} - P_{H_{j,t,s}} \; j \in N_H, \; t \in T, \; s \in n \tag{B.2}$$

$$G_2\left(P_{T_{i,t,s}}, X_{j,t,s}\right) = P_{H_{j,t,s}} - P_{H_j}^{\max} \; j \in N_H, \; t \in T, \; s \in n \tag{B.3}$$

$$G_3\left(P_{T_{i,t,s}}, X_{j,t,s}\right) = P_{T_i}^{\min} - P_{T_{i,t,s}} \; i \in N_T, \; t \in T, \; s \in n \tag{B.4}$$

$$G_4\left(P_{T_{i,t,s}}, X_{j,t,s}\right) = P_{T_{i,t,s}} - P_{T_i}^{\max} \; i \in N_T, \; t \in T, \; s \in n \tag{B.5}$$

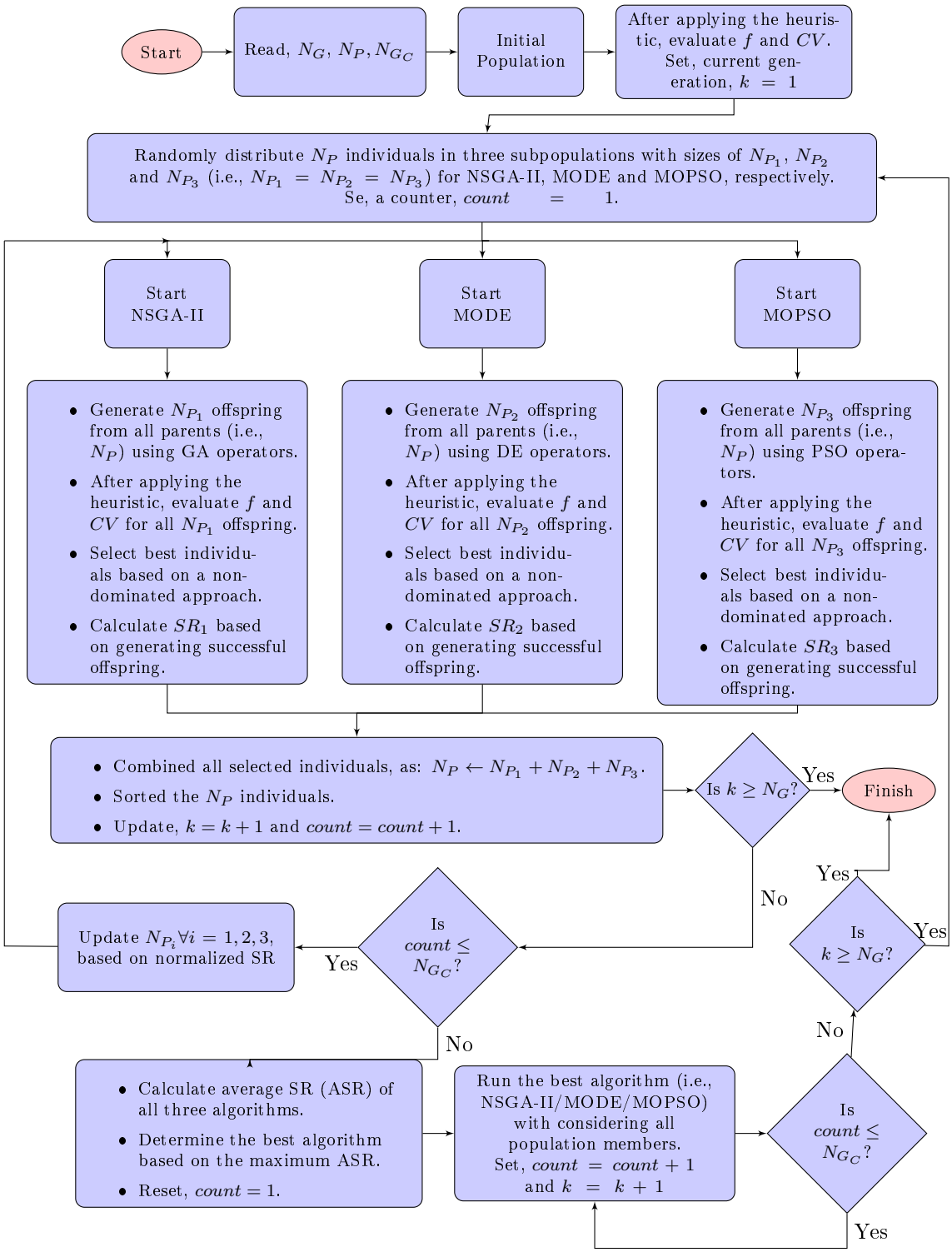

Fig. A.12: Flowchart of Algorithm-1

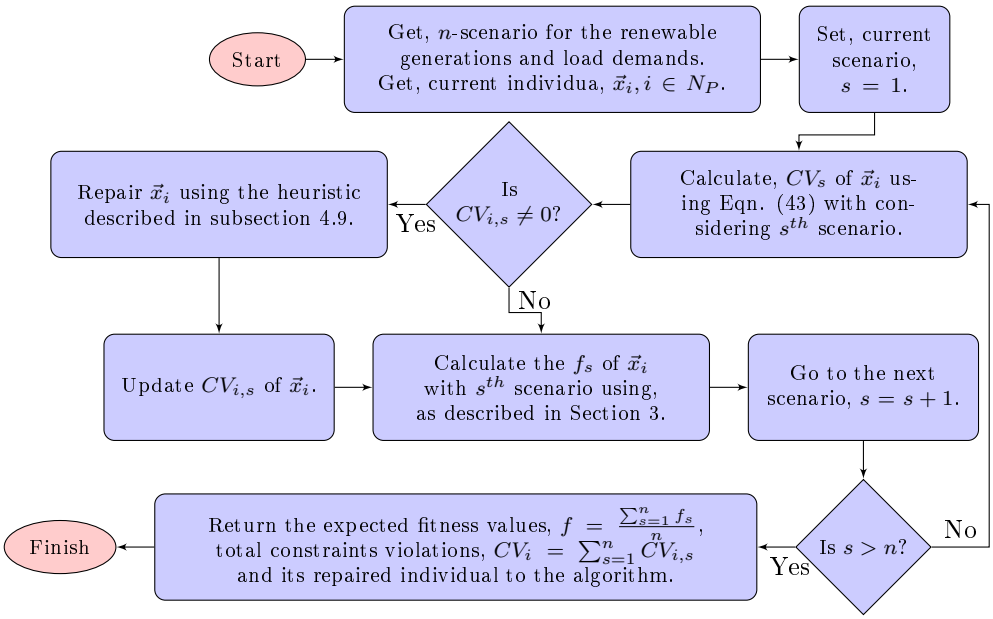

Fig. A.13: Flowchart of Algorithm-3

$$G_5\left(P_{T_{i,t,s}}, X_{j,t,s}\right) = V_{H_j}^{\min} - V_{H_{j,t,s}} \; j \in N_H, \, t \in T, \; s \in n \tag{B.6}$$

$$G_6\left(P_{T_{i,t,s}}, X_{j,t,s}\right) = V_{H_{j,t,s}} - V_{H_j}^{\max} \; j \in N_H, \, t \in T, \; s \in n \tag{B.7}$$

$$G_7\left(P_{T_{i,t,s}}, X_{j,t,s}\right) = X_{H_j}^{\min} - X_{H_{j,t,s}} \; j \in N_H, \, t \in T, \; s \in n \tag{B.8}$$

$$G_8\left(P_{T_{i,t,s}}, X_{j,t,s}\right) = X_{H_{j,t,s}} - X_{H_j}^{\max} \; j \in N_H, \, t \in T, \; s \in n \tag{B.9}$$

$$G_9\left(P_{T_{i,t,s}}, X_{j,t,s}\right) = \left| V_j^{ini} - |V_{j,t,s}|^{t=0} \right| \; j \in N_H, \, s \in n \tag{B.10}$$

$$G_{10}\left(P_{T_{i,t,s}}, X_{j,t,s}\right) = \left| V_j^{end} - |V_{j,t,s}|^{t=T} \right| \; j \in N_H, \, s \in n \tag{B.11}$$

where *tolerance* is a small value for the equality constraint relaxation, we set it to 0.001 which is acceptable in power system [51]. Note that the output of a hydro-generator, $P_{H_{j,t,s}}$ depends on the water storage rate, $X_{j,t,s}$, and its volume, $V_{H_{j,t,s}}$. Their relationships are already defined in Eqns. (4) and (5). The details of the above constraints are discussed in subsection 3.1.

Appendix B.2. Solar-thermal

To determine the values of G and H, the equality and inequality constraints for the solar-thermal DEED are represented, as:

$$H\left(P_{T_{i,t,s}}, \tilde{U}_{S_{k,t,s}}\right) = \left|\left(\sum_{i=1}^{N_T} P_{T_{i,t,s}} + \sum_{k=1}^{N_S} \tilde{P}_{S_{k,t,s}} U_{S_{k,t,s}}\right) - \tilde{P}_{D_{t,s}}\right| - tolerance,\ t \in T,\ s \in n \tag{B.12}$$

$$G_1\left(P_{T_{i,t,s}}, \tilde{U}_{S_{k,t,s}}\right) = P_{T_i}^{\min} - P_{T_{i,t,s}}\ i \in N_T,\ t \in T,\ s \in n \tag{B.13}$$

$$G_2\left(P_{T_{i,t,s}}, \tilde{U}_{S_{k,t,s}}\right) = P_{T_{i,t,s}} - P_{T_i}^{\max}\ i \in N_T,\ t \in T,\ s \in n \tag{B.14}$$

$$G_3\left(P_{T_{i,t,s}}, \tilde{U}_{S_{k,t,s}}\right) = \left(P_{T_{i,t,s}} - P_{T_{i,t-1,s}}\right) - UR_i\ \ i \in N_T\, t \in T,\ s \in n \tag{B.15}$$

$$G_4\left(P_{T_{i,t,s}}, U_{S_{k,t,s}}\right) = \left(P_{T_{i,t-1,s}} - P_{T_{i,t,s}}\right) - DR_i\ i \in N_T\, t \in T,\ s \in n \tag{B.16}$$

The descriptions of the above constraints can be found in subsection 3.2.

Appendix B.3. Wind-thermal

The values of G and H for the wind-thermal DEED problem are:

$$H\left(P_{T_{i,t,s}}, \tilde{P}_{W_{w,t,s}}\right) = \left|\left(\sum_{i=1}^{N_T} P_{T_{i,t,s}} + \sum_{w=1}^{N_W} \tilde{P}_{W_{w,t,s}}\right) - \tilde{P}_{D_{t,s}}\right| - tolerance\, s \in N_S \tag{B.17}$$

$$G_1\left(P_{T_{i,t,s}}, \tilde{P}_{W_{w,t,s}}\right) = P_{T_i}^{\min} - P_{T_{i,t,s}}\ i \in N_T,\ t \in T,\ s \in n \tag{B.18}$$

$$G_2\left(P_{T_{i,t,s}}, \tilde{P}_{W_{w,t,s}}\right) = P_{T_{i,t,s}} - P_{T_i}^{\max}\ i \in N_T,\ t \in T,\ s \in n \tag{B.19}$$

$$G_3\left(P_{T_{i,t,s}}, \tilde{P}_{W_{w,t,s}}\right) = 0 - \tilde{P}_{W_{w,t,s}}\ w \in N_W,\ t \in T\, s \in N_S \tag{B.20}$$

$$G_4\left(P_{T_{i,t,s}}, \tilde{P}_{W_{w,t,s}}\right) = \tilde{P}_{W_{w,t,s}} - P_{R_w} \tag{B.21}$$

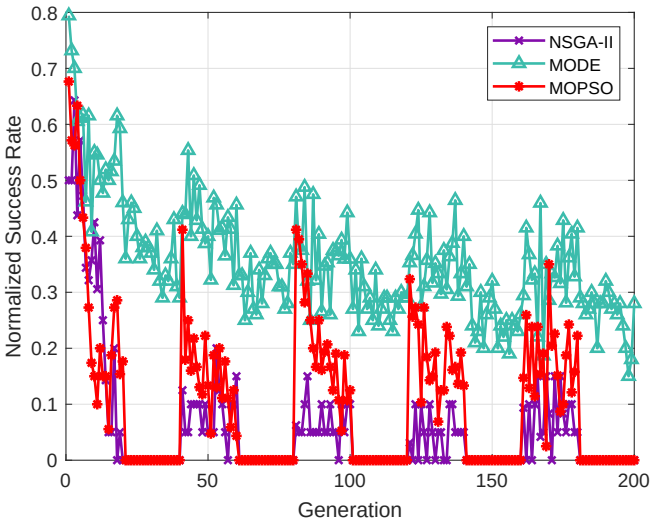

Fig. C.14: NSR of NSGA-II, MODE and MOPSO for the hydro-thermal DEED problems

$$G_5\left(P_{T_{i,t,s}}, \tilde{P}_{W_{w,t,s}}\right) = \begin{cases} UR_i - \left(P_{T_{i,t,s}} - P_{T_{i,t-1,s}}\right) & \text{if } P_{i,(t-1),s} > P_i^{\min} \\ UR_i^1 - \left(P_{T_{i,t,s}} - P_{T_{i,t-1,s}}\right) & \text{otherwise} \end{cases} \tag{B.22}$$

$$G_6\left(P_{T_{i,t,s}}, \tilde{P}_{W_{w,t,s}}\right) = \begin{cases} \left(P_{T_{i,t-1,s}} - P_{T_{i,t,s}}\right) - DR_i & \text{if } P_{i,(t-1),s} > P_i^{\min} \\ \left(P_{T_{i,t-1,s}} - P_{T_{i,t,s}}\right) - DR_i^0 & \text{otherwise} \end{cases} \tag{B.23}$$

$$G_7\left(P_{T_{i,t,s}}, \tilde{P}_{W_{w,t,s}}\right) = \left|\left[T_{i,(t-1),s}^{on} - T_{\min_i}^{on}\right] \times \left[U_{T_{i,(t-1),s}} - U_{T_{i,t,s}}\right]\right| \tag{B.24}$$

$$G_8\left(P_{T_{i,t,s}}, \tilde{P}_{W_{w,t,s}}\right) = \left|\left[T_{i,(t-1),s}^{off} - T_{\min_i}^{off}\right] \times \left[U_{T_{i,t,s}} - U_{T_{i,(t-1),s}}\right]\right|$$

All the above constraints for the wind-thermal DEED problem are discussed in subsection 3.3.

Appendix C. Illustrative Performance of Algorithm

In this subsection, the performances each algorithm in MMBA are shown in Figs. C.14, C.15 and C.16, for hydro-thermal, wind-thermal and solar-thermal DEED, respectively.

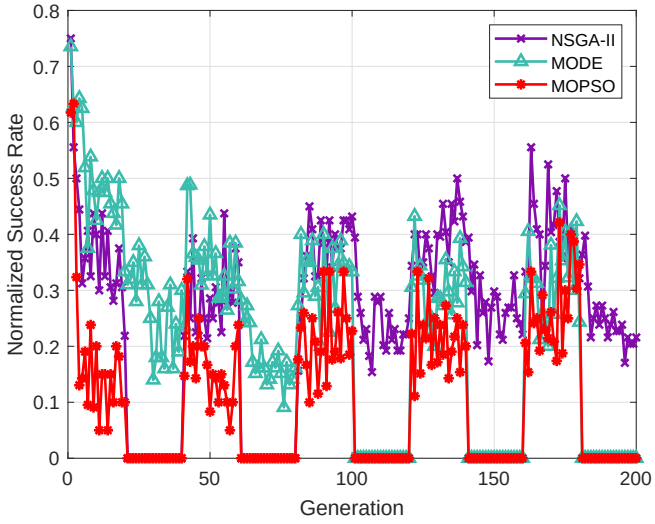

Fig. C.15: NSR of NSGA-II, MODE and MOPSO for the wind-thermal DEED problems

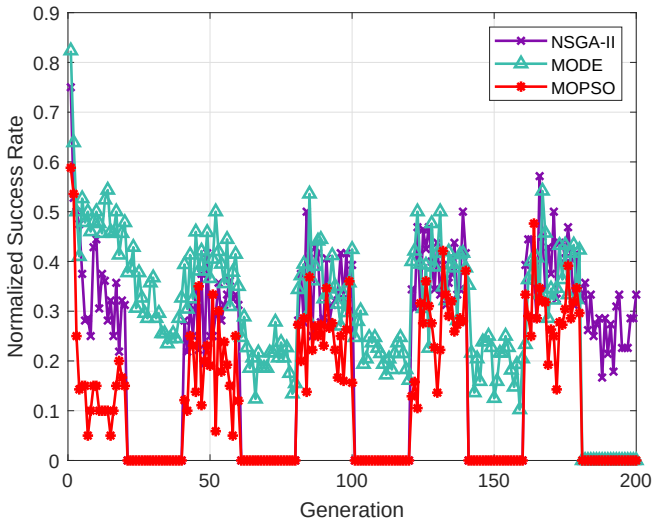

Fig. C.16: NSR of NSGA-II, MODE and MOPSO for the solar-thermal DEED problems