MULTI-OBJECTIVE POWER SYSTEM SCHEDULING USING EVOLUTIONARY ALGORITHMS

ANUPAM TRIVEDI

(*B. Tech, M. Tech, IIT Bombay*)

A THESIS SUBMITTED FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2015

I would like to dedicate this thesis to my father Mr. K.D. Trivedi, mother Mrs. Hemlata Trivedi, brother Mr. Aashwin Trivedi, my wife Mrs. Deepti Shrimali, and my daugther Tanvi Trivedi.

Declaration

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in this thesis.

This thesis has also not been submitted for any degree in any University previously.

ANUPAM Trivedi

23 January 2015

Acknowledgements

Foremost, I would like to express my sincere gratitude to my supervisor Assoc/P Dipti Srinivasan for giving me the opportunity to pursue PhD and her continuous support, guidance and encouragement throughout the period of research.

Besides my supervisor, I would like to thank my co-supervisor Dr. Thomas Reindl for his support and guidance during my research work. I really appreciate the financial support which I obtained from Solar Energy Research Institute of Singapore (SERIS).

Also, I would like to extend my sincere thanks to the members of the thesis advisory committee for their valuable comments during my CQE and OQE and the examination committee for reviewing this thesis.

I would like to thank my friends, Anirudh, Nimish, Suyash and Jayash.

I would like to thank my seniors, Dr. Balaji, Dr. Naran and Dr. Deepak for their help and suggestions on how to carry out good research. Contributions by Anurag, Rahul and intern students - Subhodip and Kunal are gratefully acknowledged.

I would like to sincerely thank my father Mr. K.D. Trivedi and my mother Mrs. Hemlata Trivedi for providing me the best childhood I could have ever got and supporting me through my education so that I could become able enough to pursue PhD. I would like to sincerely thank my brother Aashwin Trivedi whose principles developed my fighting spirit during my undergraduate days. I would like to sincerely thank my wife Deepti Shrimali for being extremely supportive and encouraging during the toughest days of my life. I am really grateful for her patience. I would also like to thank my newly born daughter, Tanvi Trivedi. I would like to dedicate this thesis to my parents, brother, my wife and my daughter.

Lastly, but most importantly, I am deeply grateful to the almighty God for giving me the strength and confidence to move on during the toughest obstacles and always showing me the way through.

Abstract

The day-ahead unit commitment (UC) problem is a nonlinear, high dimensional, highly constrained, mixed-integer NP-hard power system scheduling problem and is usually solved in the literature considering system operation cost as the single (economic) objective. However, the system operators would prefer to obtain the trade-off optimal solutions considering emission and reliability as the conflicting objectives along with system operation cost for better decision making. The primary objective of the thesis is to consider emission and reliability as objectives along with system operation cost and solve the UC problem as a multi-objective optimization problem using evolutionary algorithms (EAs).

Firstly, the UC problem in deterministic environment involving system operation cost as the single objective is tackled. An evolutionary optimization skeleton based on problem-specific: chromosome representation, genetic operators, and knowledge is developed. Furthermore, a hybrid framework synergizing genetic algorithm (GA) with differential evolution (DE) algorithm is proposed to efficiently solve the UC problem. The experimental results reveal that the proposed hybrid EA is superior to some of the best approaches proposed in the literature.

Subsequently, the bi-objective UC problem in deterministic environment considering system operation cost and emission as the conflicting objectives is tackled. The optimization skeleton developed for the single-objective UC problem is embedded within the domination and decomposi-

tion based multi-objective optimization frameworks. Non-dominated sorting genetic algorithm II (NSGA-II) and multi-objective evolutionary algorithm based on decomposition (MOEA/D-SBX and MOEA/D-DE) are selected as the representative algorithms from the domination and decomposition frameworks, respectively and efficiently customized. The proposed multi-objective evolutionary algorithms (MOEAs) are comprehensively compared among themselves and it is found that MOEA/D-DE significantly outperforms the contender algorithms. Thereafter, a non-uniform weight vector distribution (NUWD) strategy is proposed to bias the search direction and its effect on the performance of MOEA/D-DE is investigated. Finally, an ensemble optimizer based on combination of MOEA/D-DE with uniform and NUWD strategy is presented. The experimental results reveal that the proposed ensemble optimizer significantly outperforms the benchmark algorithms presented in the literature and returns a well-converged and uniformly distributed set of trade-off optimal solutions.

Thereafter, the bi-objective UC problem formulation is extended to include reliability as an additional objective along with system operation cost and emission. The uncertainties occurring due to thermal generator outage and load forecast error are captured using expected energy not served (EENS) reliability index and EENS cost is used to reflect the reliability objective. MOEA/D-DE developed for the bi-objective UC problem is applied to solve the three-objective UC problem. Thereafter, a non-uniform weight vector distribution (NUWD) strategy is proposed to enhance the performance of MOEA/D-DE. Furthermore, MOEA/D-DE with ϵ-dominance based external archive is presented and is found to return a well distributed set of trade-off optimal solutions.

Finally, the three-objective UC problem formulation (in uncertain environment) is further extended to include significant levels of wind penetration. The additional uncertainty due to wind forecast error is incorporated along with the uncertainties due to thermal generator outage and load forecast error using EENS reliability index. MOEA/D-DE with ϵ-dominance based external archive developed for the three-objective UC problem is implemented to solve the three-objective wind-thermal UC problem. The experimental results reveal that the proposed MOEA/D-DE is able to return a diverse set of trade-off optimal solutions for the wind-thermal UC problem.

List of publications

International Journal Publications

1. **A. Trivedi**, D. Srinivasan, D. Sharma and C. Singh, "Evolutionary Multi-objective Day-ahead Thermal Generation Scheduling in Uncertain Environment", *IEEE Transactions on Power Systems*, vol. 28 (2), pp. 1345 - 1354, 2013.

2. **A. Trivedi**, D. Srinivasan, S. Biswas and T. Reindl, "Hybridizing Genetic Algorithm with Differential Evolution for Solving the Unit Commitment Scheduling Problem", *Swarm and Evolutionary Computation* (Accepted for publication).

3. **A. Trivedi**, D. Srinivasan, K. Pal, C. Saha, and T. Reindl, "Enhanced Multi-objective Evolutionary Algorithm based on Decomposition for Solving the Unit Commitment Problem", *IEEE Transactions on Industrial Informatics* (Under 3rd review).

4. D. Sharma, **A. Trivedi**, D. Srinivasan and L. Thillainathan, "Multi-Agent Modeling for Solving Profit Based Unit Commitment Problem", *Applied Soft Computing*, vol. 13 (8), pp. 3751-3761, 2013.

5. A. Sharma, D. Srinivasan, and **A. Trivedi**, "A Decentralized Multi-Agent System Approach for Service Restoration Using DG Islanding", *IEEE Transactions on Smart Grids* (Accepted for publication).

6. **A. Trivedi**, D. Srinivasan, S. Biswas and T. Reindl, "A Hybrid Framework Synergizing Genetic Algorithm with Differential Evolution for the Unit Commitment Problem", *Information Sciences* (Submitted).

7. **A. Trivedi**, D. Srinivasan, and T. Reindl, "A Multi-objective Evolutionary Algorithm based on Decomposition for Solving the Unit Commitment Problem in Uncertain Environment", *IEEE Transactions on Smart Grid* (Submitted).

8. R. Mehta, D. Srinivasan, A. M. Khambadkone, and **A. Trivedi**, "Integration of Plug-in Electric Vehicle Charging Parks as Intelligent Virtual Power Plants into Distribution System", *IEEE Transactions on Smart Grid* (Submitted)

International Conference Publications

1. **A. Trivedi**, N. M. Pindoriya and D. Srinivasan, “Modified NSGA-II for Day-ahead Multi-objective Thermal Generation Scheduling”, *International Power and Energy Conference*, pp. 752 - 757, 2010.

2. **A. Trivedi**, N. M. Pindoriya, D. Srinivasan and D. Sharma, “Improved Multi-objective Evolutionary Algorithm for Day-ahead Thermal Generation Scheduling”, *Evolutionary Computation (CEC), IEEE Congress*, pp. 2170 - 2177, 2011.

3. **A. Trivedi**, D. Sharma and D. Srinivasan, “Multi-objectivization of Short-term Unit Commitment under Uncertainty using Evolutionary Algorithm”, *Evolutionary Computation (CEC), IEEE Congress*, pp. 1 - 8, 2012.

4. D. Sharma, D. Srinivasan and **A. Trivedi**, “Multi-Agent Approach for Profit Based Unit Commitment"”, *Evolutionary Computation (CEC), IEEE Congress*, pp. 2507 - 2513, 2011.

5. **A. Trivedi**, D. Srinivasan T. Reindl and C. Singh, “A Multi-objective Evolutionary Algorithm for Unit Commitment with Significant Wind Penetration”, *Evolutionary Computation (CEC), IEEE Congress*, 2015 (Accepted for publication).

Contents

List of Tables

List of Figures

Nomenclature

Variables

E_i^t	Pollutants produced by unit i at hour t in lb
f_i^t	Fuel cost of unit i at hour t in \$/h
P_i^t	Power generated by unit i at time t
$T_{OFF,i}^t$	Continuously off time of unit i up to hour t
$T_{ON,i}^t$	Continuously on time of unit i up to hour t
u_i^t	Unit commitment status of unit i at time t ($1 = ON$, $0 = OFF$)

Constants

a_{1i}, b_{1i}, c_{1i}	Emission coefficients of unit i
a_i, b_i, c_i	Fuel cost coefficients of unit i
CR	Crossover rate of Differential Evolution
CSC	Cold start-up cost of unit i
eta_c	SBX distribution index
HSC	Hot start-up cost of unit i
L^t	Load demand at hour t
max_gen	Maximum generations/iterations
MUT_i/MDT_i	Minimum up/down time of unit i
N	Number of generating units
NP	Population size
$P_{max,i}$	Rated upper limit generation of unit i
$P_{min,i}$	Rated lower limit generation of unit i
SD_i^t	Shut-down cost of unit i at hour t
SR_i^t	System spinning reserve requirement at hour t

SU_i^t	Start-up cost of unit i at hour t
T	Neighborhood size
$T_{cold,i}$	Cold start hour of unit i
T_{max}	Number of hours is the scheduling horizon

Index

i	Generating unit index
i	Hourly time index

Acronyms / Abbreviations

$COPT$	Capacity outage probability table
$hGADE/cur1$	hGADE/current-to-rand/1
$hGADE/cur2$	hGADE/current-to-rand/2
$EENS$	Expected energy not served
$EENS_{max}$	User-defined upper limit for solution's EENS cost
$EENS_t$	Expected energy not served at hour t
$EENS_{tot}$	Total expected energy not served for entire scheduling horizon
$Emis_{max}$	User-defined upper limit for solution's emission
$FLAC$	Full load average cost
PL	Priority list
$hGADE$	Hybrid of GA and DE
IGD	Inverted generational distance
$MOEA$	Multi-objective evolutionary algorithm
$MOEA/D$	Multi-objective evolutionary algorithm based on decomposition
$MOEE-UC$	Multi-objective economic/emission unit commitment
$MOEER-UC$	Multi-objective economic/emission/reliability unit commitment
$MOWT-UC$	Multi-objective wind-thermal unit commitment
MOP	Multi-objective optimization problem
$NSGA-II$	Non-dominated sorting genetic algorithm II

$hGADE/r1$	hGADE/rand/1
$hGADE/r2$	hGADE/rand/2
RPM	Real power matrix
SOC	System operation cost
SOC_{max}	User-defined upper limit for solution's system operation cost
UC	Unit commitment
UCM	Unit commitment matrix
$VOLL$	Value of lost load MW/h

Chapter 1

Introduction

Load demand in electric power systems varies according to the consumer behavior. The demand is generally higher during the daytime and early evenings when loads are high. On the other hand, the demand is lower during late nights and early mornings when most of the population is asleep. A sufficient number of generating units need to be committed to meet this variable demand. However, if enough generating units are committed to meet the peak load demand and these units are kept on at all times, then monetary losses may be incurred. Therefore, the load demands need to be satisfied while operating the power system economically. This process of determining optimal schedule of generating units over a particular time horizon, meeting the objective of minimizing the system operation cost, subject to unit and system operating constraints, is known as unit commitment (UC) [1]. It is a nonlinear, mixed-integer, combinatorial, high-dimensional and highly constrained optimization problem and belongs to the set of NP-hard problems [1]. In the literature, various extensions of the UC problem have been studied like security-constrained unit commitment (SCUC) [2], profit-based unit commitment (PBUC) [3], etc. However, the core of the problem remains to be unit commitment.

Thus, unit commitment is one of the most important problems in power system scheduling. Due to its economic importance, the UC problem has for long been a matter of concern for power system companies. Over the years, a lot of research has been conducted on developing efficient UC algorithms which can be mainly grouped as a) numerical optimization techniques and b) stochastic search based techniques. Numerical optimization techniques such as priority list (PL) [4], dynamic programming (DP) [5], mixed-integer linear programming (MILP) [6], branch and bound (BB) [7], and Lagrangian relaxation (LR) [8] have been proposed for the UC problem. These methods are simple and fast but most of them suffer from numerical convergence and solution quality problems. Stochastic search based techniques such as genetic algorithm (GA) [9, 10], evolutionary programming (EP) [11], memetic algorithm (MA) [12], particle swarm optimization (PSO) [13, 14], simulated annealing (SA) [15], quantum-inspired evolutionary algorithm (QEA) [16], differential evolution (DE) [17], artificial bee colony algorithm [18], firefly algorithm [19], and gravitational search algorithm [20] have been proposed for the UC problem. These stochastic search based techniques have attracted wide recognition from researchers due to their ease of implementation, capability of accommodating complex problem characteristics and attaining optimal/near-optimal solution. However, the stochastic search based techniques have the disadvantage that they do not guarantee convergence.

1.1 Motivation of Research

The UC problem is usually solved in the literature considering system operation cost as the single (economic) objective and the emission as well as reliability aspects are generally neglected. However, generation of electricity from fossil fuel releases several contaminants, such as sulfur dioxides, nitrogen oxides and carbon dioxide into the atmosphere. Due to the increasing environmental concerns that arise from the emissions and increasing awareness of environmental protection, the utilities have been pushed to improve their design and operational strategies, for reducing the emissions from the power plants. Thus, consideration of the environmental impacts of power generation in the UC problem is receiving intensive focus [21–24].

Further, the UC problem is generally solved in the literature considering a deterministic environment. However, the generation scheduling is subject to uncertainty due to deviations from load forecasts and outage of components such as generator, transmission line, etc. [25]. Thus, the system operators would prefer to obtain trade-off optimal solutions by incorporating various uncertainties and considering emission as well as reliability as additional objectives along with system operation cost for better decision making [23]. However, the UC problem considering system operation cost, emission and reliability as the multiple objectives is a nonlinear, mixed-integer, combinatorial, high-dimensional, highly constrained multi-objective optimization problem. Thus, it is a challenge to efficiently solve the UC problem as a multi-objective optimization problem and obtain different trade-off optimal solutions [23].

Further, renewable energy resources are being increasingly deployed in many countries to replace conventional generation due to concerns regarding global warming and air pollution. Amongst renewable energy resources, wind power generation has gained significant penetration in the power system because of the fast development of economical, reliable and efficient wind turbines [26]. However, the main challenge in integrating wind power generation with conventional generation is the intermittent and variable nature of wind [27]. Thus, in presence of wind generation, additional uncertainty due to wind forecast error needs to be considered. This makes the consideration of reliability objective even more important in presence of wind penetration [28].

The motivation of the research conducted in this thesis is to solve the UC problem as a multi-objective optimization problem so that the emission and reliability objectives can be simultaneously considered in the problem formulation along with system operation cost. The advantage of presenting a multi-objective approach to the UC problem is that a set of trade-off optimal solutions can be obtained which the system operator can use for enhanced decision making [23].

Several classical optimization methods like DP [5], MILP [6], LR [8], etc. have been proposed for the UC problem in the literature. However, the classical optimization methods are inherently single-objective optimizers and cannot be used to efficiently solve the problem as an ideal multi-objective optimization problem and obtain the trade-off optimal solutions [29]. On the other hand, in the last decade or so, evolutionary algorithms (EAs) have gained lot of popularity for solving multi-objective optimization problems [30]. Such EAs are called multi-objective evolutionary algorithms (MOEAs). Because of their population based nature, EAs can be used to efficiently obtain a set of trade-off optimal solutions for a complex multi-objective optimization problem in a single simulation run [29, 30].

Scheduling problems are generally complex, highly constrained, large scale and NP-hard in nature. EAs are ideal for solving scheduling problems because of their robustness in handling complex constraints, high-dimensional search space and ability to produce near-optimal solutions (if not exact optimal solution) [31]. Further, EAs are very flexible in nature and problem-specific information can be easily integrated within the EA framework in the form of heuristics. Generally, EAs enhanced with problem-specific heuristics are very efficient in solving complex scheduling problems [32]. Thus, in the literature, EAs have been widely implemented for solving several real-world scheduling problems.

EAs have found application in several single-objective real world scheduling problems like project scheduling [33, 34], storage tank scheduling [35], scheduling in steel making-continuous casting production [36], railway timetabling [37], jobshop scheduling [38], traveling salesman problem [39, 40], multiprocessor scheduling [41], etc.

Furthermore, MOEAs have been proposed for several multi-objective real world scheduling problems like flowshop scheduling [42], jobshop scheduling [43, 44], optimal energy consumption scheduling [45], traveling salesman [46, 47], berth allocation [48], examination timetabling [49], vehicle routing [50–52], resource investment project scheduling [53] etc.

This motivated us to choose EAs as the optimization tool to tackle the multi-objective UC scheduling problem. Thus, the motivation of the research conducted in this thesis is to design MOEAs for solving the UC problem as a multi-objective optimization problem.

1.2 Objectives of the Research

The main objective of the thesis is to to consider emission and reliability as objectives along with system operation cost and solve the UC problem as a multi-objective optimization problem using evolutionary algorithms (EAs). However, the main goal has been systematically split into different subgoals and the complexity of the problem has been progressively expanded.

1. **Single-objective UC problem in deterministic environment** - In order to efficiently solve the multi-objective UC problem using EA, the first aim was to develop an evolutionary framework to solve the

UC problem considering system operation cost as the only objective. This is because the multi-objective UC problem is an extension of the single-objective UC problem.

2. **Bi-objective UC problem in deterministic environment** - The second aim was to extend the single-objective UC problem to bi-objective UC problem (in deterministic environment) and consider minimizing emission as an additional objective along with minimizing system operation cost. The goal was to embed the optimization skeleton developed for the single-objective UC problem within some of the most popular MOEAs presented in the literature and suggest enhancements in order to efficiently solve the bi-objective UC problem.

3. **Three-objective UC problem in uncertain environment** - The third aim was to extend the bi-objective UC problem in deterministic environment to three-objective UC problem in uncertain environment and consider maximizing reliability as an additional objective along with minimizing system operation cost and emission. The goal was to apply the MOEAs proposed for the bi-objective UC problem (corresponding to second goal discussed above) and develop variants of the MOEA in order to efficiently obtain the trade-off optimal solutions for the three-objective UC problem.

4. **Three-objective wind-thermal UC problem in uncertain environment** - The fourth aim was to extend the three-objective UC problem (in uncertain environment) to include significant wind penetration. The goal was to implement the MOEAs proposed for the three-objective UC problem (corresponding to third goal discussed

above) in order to efficiently obtain the trade-off optimal solutions for the three-objective wind-thermal UC problem.

The specific goals are as summarized below:

- To design an evolutionary framework for solving the UC problem considering system operation cost as the single objective.
- To consider emission as an additional objective in the UC problem formulation along with system operation cost and propose MOEA for solving the bi-objective UC problem.
- To include reliability as an additional objective in the UC problem formulation along with system operation cost and emission and present MOEA for solving the three-objective UC problem.
- To incorporate significant wind penetration in the UC problem and suggest MOEA for solving the problem with system operation cost, emission and reliability as the multiple objectives.

1.3 Organization of the Thesis

- Chapter 2 briefly presents the basics of single-objective optimization, multi-objective optimization, evolutionary algorithms and multi-objective evolutionary algorithms.
- Chapter 3 presents a hybrid framework based on combination of genetic algorithm and differential evolution to solve the single-objective unit commitment problem.
- Chapter 4 extends the single-objective UC problem addressed in Chapter 3 to bi-objective UC problem by considering minimizing emis-

sion as an additional objective along with minimizing system operation cost. Non-dominated sorting genetic algorithm II (NSGA-II) and multi-objective evolutionary algorithms based on decomposition (MOEA/D-SBX and MOEA/D-DE) are selected as the representative algorithms from the domination and decomposition frameworks, respectively. The MOEAs are efficiently customized and applied to the multi-objective economic/emission unit commitment (MOEE-UC) problem. Further, variants of MOEA/D-DE are suggested to effectively solve the bi-objective UC problem.

- Chapter 5 extends the bi-objective UC problem (in deterministic environment) addressed in Chapter 4 to three-objective UC problem in uncertain environment by considering maximizing reliability as an additional objective along with minimizing system operation cost and (minimizing) emission. The MOEAs developed for the bi-objective generation scheduling problem in deterministic environment in Chapter 4 i.e., NSGA-II-SBX, MOEA/D-SBX and MOEA/D-DE are efficiently extended in this Chapter to solve the three-objective UC problem in uncertain environment. Furthermore, variants of MOEA/D-DE are presented to effectively solve the three-objective UC problem.

- Chapter 6 extends the three-objective UC problem (in uncertain environment) addressed in Chapter 5 to three-objective UC problem in presence of significant wind penetration. The multiple objectives considered remain the same as that in Chapter 5 i.e., minimizing system operation cost, minimizing emission and maximizing reliability. MOEA/D-DE and its variants proposed in Chapter 5 are implemented to solve the three-objective wind-thermal UC problem.

- Finally, Chapter 7 presents the conclusions and the contributions of thesis and discusses future work.

Chapter 2

Evolutionary Optimization

In this Chapter, the basics of single-objective optimization, multi-objective optimization, evolutionary algorithms and multi-objective evolutionary algorithms have been presented.

2.1 Single-Objective Optimization

When an optimization problem involves only one objective function, the task of finding the optimal solution is called single-objective optimization. Mathematically, a single-objective optimization problem (SOP) can be formulated (in the minimization case) as follows:

$$Minimize\ \mathbf{f}(\mathbf{x}) \tag{2.1}$$

subject to:

$$\mathbf{g_i}(\mathbf{x}) \geq 0;\ \ i = 1, 2, ..., m \tag{2.2}$$

$$\mathbf{h_j}(\mathbf{x}) = 0;\ \ j = 1, 2, ..., p \tag{2.3}$$

where $\mathbf{f}(\mathbf{x})$ is the objective function, $\mathbf{x} = (x_1, x_2, ..., x_n)$ is the vector of decision variables, $x \in \mathbb{R}^n$, $\mathbb{R}^n$ is the decision space, n is the number of decision variables, $\mathbf{g}$ is the set of m inequality constraints and $\mathbf{h}$ is the set of p equality constraints.

2.2 Multi-Objective Optimization

When an optimization problem involves more than one objective function, the task of finding one or more optimal solutions is called multi-objective optimization. Mathematically, a multi-objective optimization problem (MOP) can be formulated (in the minimization case) as follows:

$$Minimize\ \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_M(\mathbf{x})] \tag{2.4}$$

$$\mathbf{g_i}(\mathbf{x}) \geq 0;\ \ i = 1, 2, ..., m \tag{2.5}$$

$$\mathbf{h_j}(\mathbf{x}) = 0;\ \ j = 1, 2, ..., p \tag{2.6}$$

where $\mathbf{f}(\mathbf{x})$ is the set of objective functions, $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^M$, $\mathbb{R}^M$ is the objective space, M is the number of objective functions, $\mathbf{x} = (x_1, x_2, ..., x_n)$ is the vector of decision variables, $\mathbf{x} \in \mathbb{R}^n$, $\mathbb{R}^n$ is the decision space, n is the number of decision variables, $\mathbf{g}$ is the set of m inequality constraints and $\mathbf{h}$ is the set of p equality constraints.

2.2.1 Fundamental Difference Between Single-Objective Optimization and Multi-Objective Optimization

The fundamental difference between a single-objective and a multi-objective optimization problem is that in single-objective optimization problem there exists a single optimum solution to the problem while in multi-objective optimization problem there exists no single optimum solution but multiple optimal (trade-off) solutions. In multi-objective optimization, no solution from the set of optimal solutions can be said to be better than the other.

2.2.2 Main Approaches to Multi-Objective Optimization

The two main approaches to solving a multi-objective optimization problem are weighted multi-objective optimization approach and ideal multi-objective optimization approach. These approaches are briefly discussed as follows:

1. **Weighted multi-objective optimization approach**: In this approach [29], at first weights are chosen for each objective based on higher-level information (which requires problem information and experience). The multi-objective optimization problem is then converted into a single-objective optimization problem by forming a composite function as the weighted sum of objective functions. A single trade-off optimal solution is then found by employing a single-objective optimization algorithm. The schematic diagram of this approach is shown in Fig 2.1.

   The advantages of this approach are that it is simple and adequate when a reliable weight vector is known. However, the disadvantages are that estimating a reliable weight vector is difficult without any knowledge of possible consequences. Further, the trade-off solution obtained is largely sensitive to the weight vector.

2. **Ideal multi-objective optimization approach**: In this approach

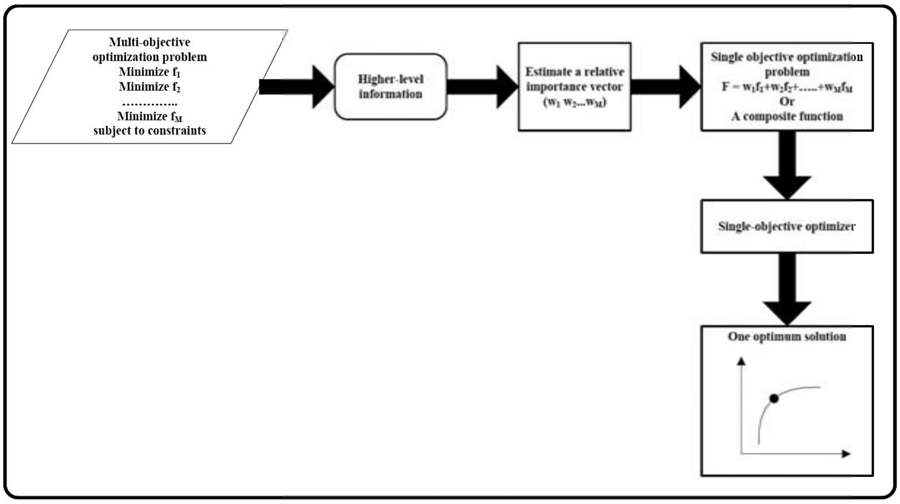


Fig. 2.1 Weighted multi-objective optimization approach.

[29], a multi-objective optimization algorithm is first implemented to find multiple trade-off solutions by considering all objectives to be important. Thereafter, higher-level information is used to choose one solution from the multiple trade-off solutions found. The schematic diagram of this approach is shown in Fig 2.2.

The advantages of this approach are that it is more methodical, practical and less subjective as compared to the weighted multi-objective optimization approach. Further, problem information is used in the second step to evaluate and compare each of the obtained trade-off solutions to choose one solution (unlike in weighted multi-objective optimization approach where problem information is used in the first step). However, this approach has its own challenges which are - a) to find a set of solutions as close as possible to the optimal trade-off front (known as Pareto-optimal front) and b) to find a set of solutions as diverse as possible. These two challenges happen to be the two goals of ideal multi-objective optimization.

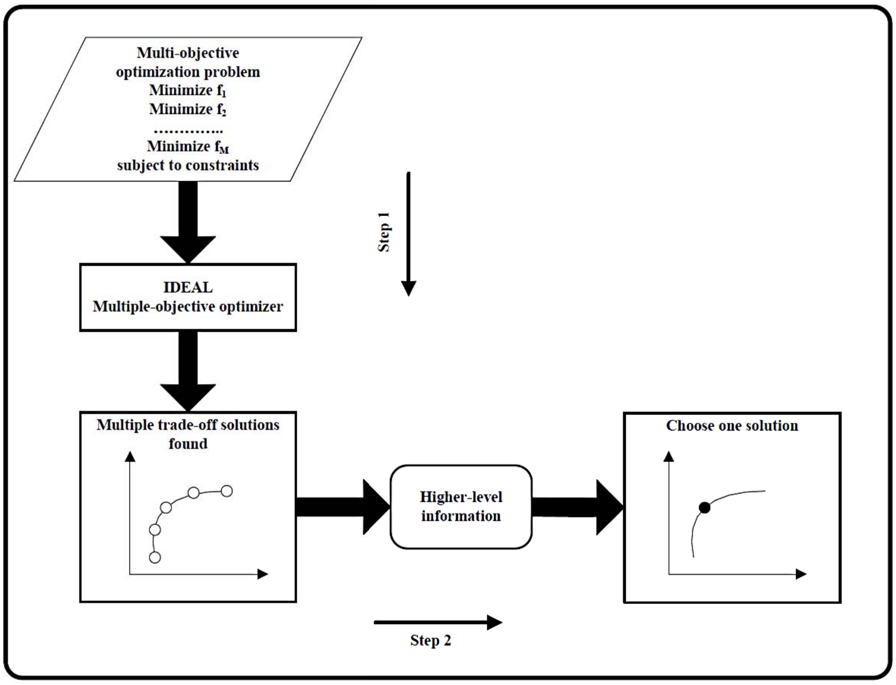


Fig. 2.2 Ideal multi-objective optimization approach.

2.3 Evolutionary Algorithms

Evolutionary algorithms (EAs) are algorithms which are inspired from nature and used for search and optimization problems. Genetic algorithms (GA) [54, 55], genetic programming (GP) [56], evolutionary programming (EP) [57], evolutionary strategies (ES) [58, 59], differential evolution (DE) [60, 61], ant colony optimization [62, 63], particle swarm optimization (PSO) [64], estimation of distribution algorithms [65, 66], etc. are some of the most popular evolutionary algorithms. Although ant colony optimization and particle swarm optimization fall under the category of swarm intelligence and differential evolution algorithms also do not exactly fall under the EA category, yet they can be loosely put under the same category as EAs. Interested readers are referred to [67] for a recent survey on EAs.

The basic characteristics of all EAs are as follows:

- Work with a population of solutions instead of a single solution.
- Do not require gradient or any auxiliary problem information except the objective function values and constraint violation values.
- Use probabilistic transition rules to guide their search and can therefore escape local optima.
- Efficient even with non-differentiable and discontinuous problems.

Algorithm 1 presents the pseudo-code of a typical EA. An EA starts with random initialization of the population. Thereafter, each solution in the population undergoes fitness evaluation which comprises of constraint violation evaluation and objective function evaluation. Subsequently, selection operation is invoked which selects only a set of fitter solutions which undergo variation operation i.e., crossover and mutation operation to form the offspring population. Next, the replacement operation takes place in which the new population is formed by selecting solutions from the parent population and the offspring population.

Algorithm 1: Pseudo-code of a typical EA

```
Begin
Initialization: Randomly initialize a population of solutions
Evaluation: Evaluate the fitness of each solution in the population
while ("Termination condition is not satisfied") do
    while ("Offspring population is not created completely") do
        Selection: Select a set of parent solutions
        Variation operation: Perform crossover and mutation operation on
        the parent solutions to create offspring solutions
    end
    Evaluation: Evaluate the fitness of each solution in the offspring population
    Replacement: Form the next population by selecting solutions from parent
    population and offspring population
end
End
```

In this thesis, two EAs which have been implemented for designing different power system scheduling algorithms are GA and DE. GA was originally developed in the early 1970s at the University of Michigan by John Holland and his students [54]. The basic GA is very generic and thus depending upon the problem, the basic GA can be easily modified. The strength of GA lies in the aspect that it can easily handle both binary variables as well as continuous variables. A binary coded GA is suitable for handling binary variables while a real-coded GA is more suitable for handling continuous variables. On the other hand, DE is a relatively new EA proposed by Price and Storn in 1995 [60] for real parameter optimization. The strength of DE is that it one of the most powerful real parameter optimizers [68] while the drawback is that it is inherently a real parameter optimizer [68]. Since DE is a very powerful real parameter optimizer, it has drawn the attention of many researchers over the last two decades resulting in many variants of the basic DE like - DE with self-adapting control parameters (jDE) [69], DE with global and local neighborhoods (DEGL) [70], adaptive DE with optional external archive (JADE) [71], self-adaptive DE (SaDE) [72], composite DE (CoDE) [73], etc. The DE variants have been applied to a plethora of real-world problems and classical benchmark problems. Interested readers are referred to [68, 74] for comprehensive review on DE including the major variants, and application of DE to different optimization problems.

It is noted that the above discussion of a typical EA (corresponding to Algorithm 1) matches more closely with the description of a GA. However, generally the exact steps of different EAs vary from each other. For example, in DE the selection of parent solutions to undergo variation operation is random whereas in GA only the fitter parent solutions undergo variation operation. Furthermore, in GA the order of variation operation is first crossover and then mutation while in DE the order is opposite i.e., first mutation and then crossover. DE is also different from traditional EAs in the sense that it perturbs the current generation population members with the scaled differences of randomly selected and distinct population members. Moreover, a GA generally involves combining the entire parent and offspring population and selecting the best solutions from the combine population to enter the next population. On the other hand, the replacement operation in DE is one to one comparison between the parent solution and the offspring solution and the fitter solution between the two enters the next population.

2.4 Multi-objective Evolutionary Algorithms

Multi-objective Evolutionary Algorithms (MOEAs) are the EAs which are designed to solve multi-objective optimization problems and obtain a set of multiple trade-off solutions in one single simulation run. MOEAs are based on ideal multi-objective optimization (discussed above) i.e., the goals of MOEAs are the same as that of ideal multi-objective optimization:

- To find a set of solutions as close as possible to the optimal trade-off front (known as Pareto-optimal front).
- To find a set of solutions as diverse as possible.

Because of the population based nature, EAs are ideal for multi-objective optimization since an approximation to the Pareto-optimal front can be obtained in a single simulation run. A MOEA shares a similar process flow as a typical EA illustrated in Algorithm 1. However, as a MOP involves multiple objectives, and the goals of a MOP are different from that of single-objective optimization, the assignment of fitness to a solution as well as selection is not straightforward as in single-objective optimization. Different MOEAs have been presented in the literature and the main distinction between different MOEAs lies in their basic framework. A recent survey of the state-of-art MOEAs is presented in [75]. Some of the popular evolutionary multi-objective optimization frameworks and widely used MOEAs are discussed below.

2.4.1 Domination-based framework

Domination based framework is one of the most widely employed framework for multi-objective optimization. In this framework, a MOP is optimized by simultaneously optimizing all the objectives. The assignment of fitness to solutions is based on Pareto-dominance principle which plays a key role in the convergence of domination-based MOEAs. Further, an explicit diversity preservation scheme is necessary in order to maintain a diverse set of solutions. Some of the most remarkable MOEAs based on this framework are multi-objective genetic algorithm (MOGA) [76], non-dominated sorting genetic algorithm (NSGA) [77], niched Pareto genetic algorithm (NPGA) [78], Pareto archived evolution strategy (PAES) [79], strength pareto evolutionary algorithm (SPEA) [80], non-dominated sorting genetic algorithm II (NSGA-II) [81], strength Pareto evolutionary algorithm 2 (SPEA2) [82], etc.

In this thesis, NSGA-II has been selected from the domination-based framework category and customized for solving different multi-objective UC problems. Thus, the methodology of NSGA-II [81] is discussed in detail below.

Brief Review of Non-Dominated Sorting Genetic Algorithm-II

The algorithm starts with a randomly generated population (of chromosomes or solutions) followed by evaluation of constraint violation and objective functions. Next, the constrained-domination principle is used to sort the population into non-dominated fronts and the constrained binary tournament selection is employed to form the mating pool. Thereafter, crossover and mutation operators are applied to form the offspring population followed by evaluation of constraint violation and objective functions (of the offspring population). The parent population and the offspring population are then combined and sorted into different non-dominated fronts. The elitism principle is implemented in which the next (i.e., new) population is filled with chromosomes of different non-dominated fronts, one at a time. The filling starts with the best non-dominated front and continues with chromosomes of the second non-dominated front, and so on. Since the population size is fixed, not all fronts can be accommodated in the next front. All fronts which cannot be accommodated are simply deleted. When the last allowed front is being considered, there may exist more chromosomes in the last front than the remaining slots in the next population. When such a situation happens, crowding distance is evaluated for the chromosomes in the last allowed front. The remaining slots for the next population are filled with chromosomes (from the last allowed front) according to the descending order of their crowding distance. Once the new population is formed, the above mentioned steps of population sorting, selection, variation and replacement based on elitism continue until the stopping criterion (as set by the user) is reached. When the algorithm stops, the last population represents the non-dominated trade-off solutions for the problem.

2.4.2 Decomposition-based framework

In this framework, a MOP is decomposed into several subproblems where a subproblem is constructed by using any aggregation-based methods. Thereafter, all the subproblems are optimized simultaneously using an EA. Unlike domination-based framework, it is not necessary to incorporate an explicit diversity preservation mechanism as the diversity is preserved implicitly because of pre-defined uniformly distributed weight vectors. The popular MOEAs based on this framework are multi-objective genetic local search (MOGLS) algorithm [83, 84] and multi-objective evolutionary algorithm based on decomposition (MOEA/D) [85].

In this thesis, MOEA/D has been employed for solving different multi-objective generation scheduling problems. Thus, the MOEA/D [85] methodology is discussed in detail below.

Brief Review of Multi-objective evolutionary algorithm based on decomposition

The basic concept of MOEA/D is based on decomposition of the target MOP into a number of scalar optimization subproblems and optimizing the subproblems simultaneously using an EA. Several decomposition approaches like weighted-sum approach, Tchebycheff approach, etc. have been proposed in [85] and any decomposition approach can be employed in the MOEA/D framework. Another concept at the core of MOEA/D framework is the neighborhood relation among the subproblems which is defined based on the distance between their aggregation coefficient weight vectors. Thus, each subproblem is optimized by using information from its neighboring subproblems only. The advantage of the MOEA/D framework is that it is generic and any EA can be incorporated to optimize the subproblems. In 2009, Zhang and Li, proposed MOEA/D-DE [86] in which the DE operators replaced the SBX operator earlier proposed in MOEA/D [85]. The basic principle of the decomposition concept involved in MOEA/D is explained below.

Suppose the target MOP is an m-objective minimization problem: *min-*

imize $F(x) = \{F_1(x), ..., F_m(x)\}$ where x is the decision variable. Let $\lambda_1, \lambda_2, ..., \lambda_N$ be a set of weight vectors where $\lambda_j = (\lambda_j^1, ..., \lambda_j^m)$ and $z = \{z_1, ..., z_m\}$ be the reference point where z_i is the best value found so far for objective F_i. According to the Tchebycheff decomposition approach, the solution to the target MOP is equivalent to optimizing N scalar optimization subproblems where the objective function of the jth subproblem is [85]

$$g^{te}(x|\lambda_j, z) = \max_{1 \leq i \leq m} \{\lambda_j^i |F_i(x) - z_i|\} \tag{2.7}$$

MOEA/D minimizes all these N objective functions simultaneously in a single run.

Since, MOEA/D was proposed by Zhang and Li in 2007 [85], it has drawn the attention of many researchers resulting in several studies on decomposition based multi-objective optimization approach and different variants of MOEA/D for improved performance [87–97].

Chapter 3

A Hybrid Framework Synergizing Genetic Algorithm with Differential Evolution for the Unit Commitment Problem

3.1 Introduction

In this Chapter, the single-objective unit commitment (UC) problem considering system operation cost as the only objective is tackled. A hybrid framework based on combination of genetic algorithm (GA) and differential evolution (DE) is proposed in this Chapter to solve the UC problem.

In Chapter 1, it was discussed that several evolutionary algorithms like genetic algorithm (GA) [9, 10], evolutionary programming (EP) [11], memetic algorithm (MA) [12], particle swarm optimization (PSO) [13, 14], differential evolution (DE) [17], etc. have been proposed for the UC problem in literature. However, simple evolutionary algorithms (EAs) are generally outperformed by hybrid EAs at solving complex optimization problems [98–100]. Hybridization, in context to EAs, essentially refers to the procedure of associating the best features of two or more algorithms together to form a superior algorithm. One of the most widely employed strategies to construct hybrid EA is by combining EA with local search algorithms. The class of such hybrid EAs is known as memetic algorithms. A comprehensive review on memetic algorithms is presented in [101–103]. Another strategy to construct hybrid EAs is by combining two or more EAs. In this Chapter, a hybrid framework based on integration of two powerful EAs - GA and DE is proposed to efficiently solve the UC problem.

The rest of the Chapter is organized as follows. The remainder of Section 3.1 throws light on EA-EA hybridization instances in the literature and discusses the motivation behind the research work. The background of the UC problem including problem formulation and related work is presented in Section 3.2. The description of the proposed hybrid evolutionary framework is presented in Section 3.3 followed by classification of the hGADE algorithm in Section 3.4. The experimental study is presented in Section 3.5 and the Chapter is summarized in Section 3.6.

3.1.1 Brief Review of EA-EA Hybrids for Solving Different Optimization Problems

Over the last decade, EAs like GA, DE, PSO, etc. have been successfully hybridized with other EAs for solving various complex optimization problems. Given below are some instances of EA-EA hybrids that have been developed and tested on challenging real-world applications and numerical benchmark problems.

Hybridization instances of GA with other EAs

In [99], recurrent network design by a hybrid of GA and PSO, named HGAPSO, is proposed in which (at each iteration) the top half best-performing individuals are marked as elites and the rest are discarded. The new population is then constituted by two parts: enhanced elites obtained after PSO enhancement form the first part while the offspring obtained after implementation of GA on enhanced elites form the other part. In [104], a hybrid of PSO and GA, called HPSOGA, is investigated for multi-UAV (unmanned aerial vehicle) formation reconfiguration problem. In HPSOGA, GA and PSO are hybridized in every iteration at sub-population level i.e., the population is probabilistically divided into two sub-populations, one of which is evolved using PSO and the other using GA. A hybrid of GA and API (a special class of ant colony optimization for continuous domains), termed GAAPI, is proposed in [105] for global continuous optimization. In GAAPI, API is the main optimizer while GA provides the escape mechanism from local optima when API is trapped.

Hybridization instances of DE with other EAs

A hybrid of DE and estimation of distribution algorithm (EDA), termed DE/EDA, is introduced in [106] for global optimization. In DE/EDA, the offspring generation scheme of DE is modified such that each component of the trial solution is probabilistically created either using differential mutation of DE or probability distribution model as in EDA. A hybrid of DE and covariance matrix adaptation evolutionary strategy (CMA-ES), named as differential covariance matrix adaptation evolutionary algorithm (DCMA-EA), is proposed in [107] for real parameter optimization. In DCMA-EA, the mutation, crossover and selection strategy of DE are embedded into the structure of a CMA-ES algorithm. In [108], a hybrid framework based on combining PSO with DE is suggested for real parameter optimization. In this framework, PSO is the main optimizer while after every iteration of PSO, DE evolves the personal-best positions of the swarm particles to enhance the convergence of PSO. A hybrid of DE and quantum PSO (QPSO), called DEQPSO, is presented for route planning of unmanned aerial vehicle in [100]. In DEQPSO, QPSO and DE are hybridized in sequential order at population level i.e., at every iteration, the population undergoes evolution first using QPSO and then using DE. In the literature, numerous hybrids based on DE and PSO have been proposed. A comprehensive review of such hybrid EAs is presented in [109].

Hybridization instances of GA with DE

In [110], a multi-method self-adaptive approach, termed AMALGAM, is proposed for multi-objective optimization problems. In AMALGAM, the population at each generation is evolved using GA, PSO, DE and adaptive metropolis search (AMS). The number of offspring an individual algorithm contributes at each generation is determined using a self-adaptive learning strategy which favor individual algorithms with highest reproductive success. The authors of [110], later extended AMALGAM to single-objective optimization and termed the algorithm as AMALGAM-SO [111]. An adaptive synergistic combination of multiple EAs, including GA, DE and EDA is proposed for multi-objective optimization in [112]. The approach has similar concept as AMALGAM [110] but differs in incorporating - progressively control paradigm with respect to the number of offspring an individual algorithm contributes at each generation, local search algorithm, and both domination and decomposition based frameworks of multi-objective optimization.

3.1.2 Motivation behind Proposing a Hybrid of GA and DE for the UC Problem

The survey shows that various EA-EA hybrids have been proposed in the literature for solving different optimization problems. However, rarely an EA-EA hybrid has been proposed for solving mixed-integer optimization problems. In order to efficiently solve a mixed-integer optimization problem, the algorithm employed should be able to effectively explore both discrete as well as continuous search space. As already stated, the UC problem is a mixed-integer optimization problem consisting of both binary UC variables and continuous power dispatch variables. Thus, the basic idea is to propose an EA-EA hybrid optimizer in which an EA which is well known for handling binary variables is hybridized with another EA which is popular for handling continuous variables. It is well known that there is no universal optimizer existing for both discrete optimization as well as continuous optimization. Thus, among the many powerful EAs existing in the literature for discrete and continuous optimization, two EAs had to be selected for hybridization.

DE algorithm is superior to other EAs like GA and PSO in solving real parameter optimization problems and has been shown to outperform the latter algorithms on several numerical benchmark problems [113]. Further, recently it was observed for real parameter multi-objective optimization problems that if SBX operator [114] is replaced by differential mutation operator of DE then the performance of NSGA-II [81] and MOEA/D [85] can be enhanced [86]. However, the limitation of DE is that it is inherently a real parameter optimizer and is yet to gain reputation in solving mixed-integer optimization problems [68]. On the other hand, among the existing EAs, GA is quite popular in the literature for its robustness on discrete optimization problems.

Motivated by the observation that GA and DE are capable of efficiently handling binary variables and continuous variables, respectively, in this Chapter, a hybrid framework based on integration of GA and DE, named hGADE, is proposed to solve the mixed-integer UC problem. In hGADE, at every generation, GA operates on the binary component of the solution (i.e., chromosome) while DE operates on the continuous component of the solution so that the hybrid optimizer is able to efficiently explore both binary search space and continuous search space. The fundamental idea behind developing hGADE algorithm is to combine GA and DE in such a way that they may complement the limitations of each other while maintaining their strengths in solving the mixed-integer UC problem.

3.2 Background

3.2.1 Unit Commitment Problem Formulation

In this Section, the UC problem formulation is presented.

Objective Function: System Operation Cost

The objective function of the UC problem is to minimize the system operation cost (SOC), where the SOC includes the fuel cost and the transition cost of all the generating units over the entire scheduling horizon [13].

The fuel cost ${f_i}^t$ of unit i is expressed as the quadratic function of its power output ${P_i}^t$ during hour t.

$$ {f_i}^t = a_i({P_i^t})^2 + b_i({P_i}^t) + c_i \tag{3.1} $$

where a_i, b_i, c_i are the fuel cost coefficients of unit i.

The transition cost is the sum of the start-up costs and the shut-down costs. In this Chapter, the shut-down costs have not been taken into consideration in accordance with the other approaches in literature [13, 14] while the start-up cost is modeled as follows:

$$SU_i^t = \begin{cases} HSC_i, & \text{if } MDT_i \leq T_{OFF,i}^t \leq MDT_i + T_{cold,i} \\ CSC_i, & \text{if } T_{OFF,i}^t > MDT_i + T_{cold,i} \end{cases} \tag{3.2}$$

where SU_i^t is the start-up cost of unit i at hour t, HSC_i and CSC_i represents the hot start cost and cold start cost of unit i, respectively, MDT_i represents the minimum down time of unit i, $T_{OFF,i}^t$ is the continuous off time of unit i up to hour t and $T_{cold,i}$ is the cold start cost of unit i.

Subsequently, the objective function of the UC problem is given by minimization of the following cost function [13].

$$F_1 = \sum_{t=1}^{T_{\max}} \sum_{i=1}^{N} \left(f_i^t\, u_i^t + SU_i^t\, (1 - u_i^{t-1}) u_i^t \right) \tag{3.3}$$

where u_i^t represents the unit commitment status of unit i at hour t ($1 = ON$, $0 = OFF$), T_{max} is the number of hours in the scheduling horizon and N is the number of thermal generating units in the system.

Constraints

1. *System power balance*: The total power generation at hour t must be equal to the load demand L^t for that hour.

$$\sum_{i=1}^{N} (P_i^t . u_i^t) = L^t, \quad t = 1, 2, ...T_{max} \tag{3.4}$$

2. *System spinning reserve requirements*: For reliable operation, the system must carry certain reserve capacity at every hour (SR^t) in order to meet unforeseen situations such as deviation in actual load demand

from forecast load demand or generator outage.

$$\sum_{i=1}^{N} (P_{max,i}.u_i^t) \geq L^t + SR^t, \quad t = 1, 2, ...T_{max} \tag{3.5}$$

where $P_{max,i}$ represents the rated upper limit generation of unit i.

3. *Unit minimum up/down time*: If a unit i is turned on/off, it must remain on/off for at least its minimum up/down time (MUT_i/MDT_i) duration.

$$\begin{aligned} T^t_{ON,i} &\geq MUT_i \\ T^t_{OFF,i} &\geq MDT_i \end{aligned} \tag{3.6}$$

where $T^t_{ON,i}$ and $T^t_{OFF,i}$ represent the continuous on and off time of unit i up to hour t, respectively.

4. *Unit generation limits*: For stable operation, the power output of each generator is restricted within its limits as follows:

$$P_{min,i} \leq P_i^t \leq P_{max,i} \tag{3.7}$$

where $P_{min,i}$ and $P_{max,i}$ represent the rated lower and upper limit generation of unit i, respectively.

3.2.2 Related Work

As mentioned earlier, the unit commitment problem consists of two tasks - one is determining the on/off status of the thermal units and the other is the power dispatch. In the literature, the stochastic search based UC algorithms have been found to outperform the numerical optimization algorithms. In the specialized literature, these algorithms can be further classified into two categories.

Category 1

The algorithms under this category employ stochastic algorithms for determining the best combination of on/off status of the thermal units while the power dispatch is carried out using problem-specific techniques like lambda iteration method or other economic dispatch methods.

A local search based binary GA, i.e., a memetic algorithm (MA) is presented in [12] to solve the UC problem in which the GA evolves the binary UC variables while the load dispatch is performed using the lambda iteration method. This approach incorporated heuristic operator for repairing minimum up/down time constraint and priority list (PL) based swap mutation operator. Further, at every generation, two local search operators are applied to improve the best solution. In [115], a quantum inspired evolutionary algorithm (QEA) based on certain principles of quantum computing such as quantum bits, quantum gates, etc. is proposed for combinatorial optimization problems. A method based on employing QEA for unit scheduling and lambda iteration for load dispatch is presented to solve the UC problem in [16]. A repair operator is applied for satisfying minimum up/down time constraint. Further, the approach employed PL based repair operators for satisfying spinning reserve constraint and handling over commitment. An enhanced PSO (EPSO) based on employing binary PSO for evolving binary UC variables and lambda iteration method for load dispatch is proposed in [14] to solve the UC problem. In [20], a binary gravitational search algorithm (BGSA) is suggested to handle the unit scheduling problem and lambda iteration method is applied for load dispatch. In both EPSO [14] and BGSA [20], PL based repair operators are used for satisfying spinning reserve constraint and decommitment of excess units (just like in the QEA approach [16]). The BGSA approach additionally employed local search based mutation strategies to prevent premature convergence. In [9], a binary GA is proposed to handle the unit scheduling problem while economic dispatch method is employed for dispatching power to the committed units. In this approach, problem-specific variation operators like swap window mutation and window mutation are applied. Additionally, two local search operators - swap mutation and swap-window hill climb operator are applied to the best chromosome at every generation in order to prevent premature convergence.

Category 2

The algorithms under this category employ stochastic algorithms for determining both the best combination of on/off status of the thermal units and the power dispatch. Recently, hybrid of binary-coded and real-coded PSO [13], hybrid of binary-coded and real-coded DE [17], hybrid of binary-coded and real-coded GA [10], hybrid of binary-coded and real-coded firefly algorithm [19], and hybrid of binary-coded and real-coded artificial bee colony algorithm [18] have been proposed to solve the UC problem. In all of these approaches, as discussed above in the category 1, heuristic based repair operator for satisfying minimum up/down time constraint, and PL based repair operators for satisfying spinning reserve constraint and decommitment of excess units are adopted. Further, unlike the algorithms in the category 1, the algorithms in this category additionally employ PL based repair operator for satisfying load demand equality constraint as well.

3.2.3 Incorporation of Domain Specific Knowledge

The literature review shows that the algorithms proposed in the literature for solving the UC problem incorporate substantial domain specific knowledge in the form of priority list (PL) for repairing load demand equality constraint, spinning reserve constraint and decommitment of excess units. Further, heuristic operators are also employed for repairing minimum up/down time constraint violation. Additionally, some of the approaches employ problem-specific variation operators and local search operators. This is because UC problem is a nonlinear, high-dimensional, highly constrained, NP-hard optimization problem. Thus, incorporation of domain specific knowledge is essential to efficiently solve the UC problem. The benefits of incorporating domain-specific knowledge in EAs for solving real world optimization problems are discussed in detail in [32].

3.2.4 Similarity and Difference of the Proposed Hybrid Algorithm hGADE to other EA-EA Hybrids

The hGADE framework proposed in this Chapter for solving the UC problem belongs to aforementioned category 2 as GA is used to evolve the binary UC variables while DE is used to evolve the continuous power dispatch variables. Thus, the proposed hGADE framework is similar to the framework of other EA-EA hybrids in the sense that two EAs are synergized in order to harmoniously boost up the strengths of each algorithm. Further, the hGADE framework also incorporates domain-specific knowledge in the form of problem-specific variation operators, PL based heuristic initialization and repair operation for satisfying load demand equality constraint.

However, hGADE is different from other EA-EA hybrids in the sense that GA and DE are hybridized such that at every generation, GA which is efficient in handling binary variables is employed to search the binary search space while DE which is more efficient in handling continuous variables is employed to search the continuous search space of the mixed-integer UC problem. To the best of our knowledge, this work presents a first attempt to hybridize two powerful EAs - GA and DE in the aforementioned manner to solve a challenging real-world mixed-integer optimization problem.

3.3 Proposed Hybrid Framework: hGADE

In this Section, the proposed hybrid framework hGADE is vividly outlined in context of application to the UC problem. A flowchart representing generic hGADE framework which can be applied to any single-objective mixed-integer optimization problem is shown in Fig. 3.1.

3.3.1 Chromosome Representation

For every chromosome, a $N \times T_{max}$ binary unit commitment matrix (UCM) is used to represent the thermal generator on/off status and a $N \times T_{max}$ real power matrix (RPM) is used to represent the corresponding power dispatch. The chromosome representation is depicted in Fig. 3.2. It is noted that a chromosome's actual generation schedule is represented by its resultant power matrix (Res.PM) which is obtained by multiplying the corresponding elements of UCM and RPM.

It is worthwhile noting that the on/off status of the thermal units and the power dispatch can be represented by a single matrix as well. However, in such a case, the variation of binary variables and continuous variables can be a complicated task because both the binary operators and the real parameter operators will be acting on a single matrix. Thus, for convenience, the binary variables and the continuous variables are represented by two separate matrices - unit commitment matrix (UCM) and real power matrix (RPM), respectively. Further, such double matrix chromosome representation has also been adopted by other researchers in the literature for solving the UC problem [10, 17].

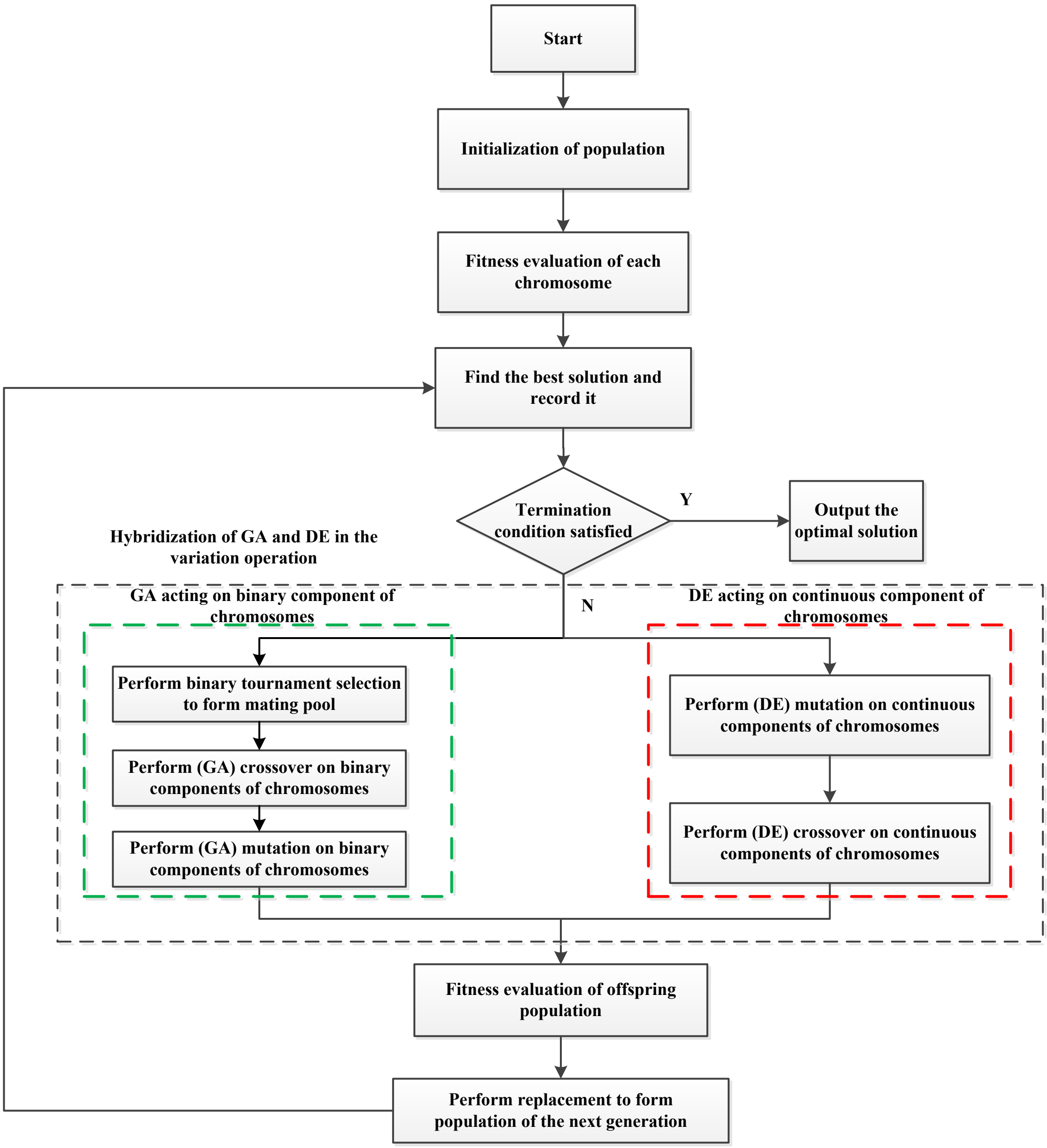


Fig. 3.1 Flowchart of the proposed hGADE framework.

3.3.2 Generation of Initial Population

The initial population is usually generated randomly. However, since UC is a highly constrained optimization problem, the algorithm always starts from the infeasible search space. A lot of time is wasted exploring the infeasible search space and the convergence is slow. Therefore, in this work, to provide some direction to the algorithm and speed up the convergence, a

	1	2	...	T_{max}-1	T_{max}
1	u_1^1	u_1^2	u_i^t	$u_1^{T_{max}-1}$	$u_1^{T_{max}}$
2	u_2^1	u_2^2	u_2^t	$u_2^{T_{max}-1}$	$u_2^{T_{max}}$
:	u_i^1	u_i^2	u_i^t	$u_i^{T_{max}-1}$	$u_i^{T_{max}}$
N-1	u_{N-1}^1	u_{N-1}^2	u_{N-1}^t	$u_{N-1}^{T_{max}-1}$	$u_{N-1}^{T_{max}}$
N	u_N^1	u_N^2	u_N^t	$u_N^{T_{max}-1}$	$u_N^{T_{max}}$

UCM

	1	2	...	T_{max}-1	T_{max}
1	P_1^1	P_1^2	P_i^t	$P_1^{T_{max}-1}$	$P_1^{T_{max}}$
2	P_2^1	P_2^2	P_2^t	$P_2^{T_{max}-1}$	$P_2^{T_{max}}$
:	P_i^1	P_i^2	P_i^t	$P_i^{T_{max}-1}$	$P_i^{T_{max}}$
N-1	P_{N-1}^1	P_{N-1}^2	P_{N-1}^t	$P_{N-1}^{T_{max}-1}$	$P_{N-1}^{T_{max}}$
N	P_N^1	P_N^2	P_N^t	$P_N^{T_{max}-1}$	$P_N^{T_{max}}$

RPM

Fig. 3.2 Structure of chromosome.

heuristic based initial population generation method is incorporated. In the heuristic initialization method, the initial population (i.e., UCM and RPM of chromosomes) is generated randomly except for one solution; UCM of which is generated using Priority list (PL) based on FLAC (full load average cost i.e., per MW cost at maximum power output of thermal units). Algorithm 2 shows the pseudo-code for creating PL. The reason behind seeding the initial random population with only one PL based solution is that one solution is enough to guide the algorithm towards the feasible space and our pilot experiments demonstrated that more PL based solutions were not found to improve the performance of the proposed algorithm. Algorithm 3 shows the pseudo-code for creation of the UCM of single heuristic solution in the initial population.

Algorithm 2: Pseudo-code for creating Priority List

input : $Pmax$ and N
output : $PriorityList$
begin
  for $unit = 1 : N$ **do**
    $FLAC = (a(Pmax(unit))^2 + b(Pmax(unit)) + c)/Pmax$;
  end
  Sort FLAC in ascending order and assign index (priority) starting from 1 and if two units have the same FLAC, then assign them the same priority;
end

Algorithm 3 shows that for the heuristic solution, corresponding to each hour, the units are turned on according to the ascending order of the PL (priority been ordered from 1 to 10 assuming there are 10 different units as in the base test system considered in this work) until the total capacity of committed units for that hour (i.e., P_{max}^{total}) is not greater than equal to summation of load demand and spinning reserve requirement. This condition ensures that the spinning reserve requirement constraint is met for the heuristic solution for each hour and the repair process (described later) ensures that for each hour of the heuristic solution, the load demand equality constraint is also satisfied. The UCM of the rest of the chromosomes in the initial population are randomly generated binary unit commitment matrices while the RPM of all the chromosomes in the initial population are generated as follows. Suppose the RPM of the kth chromosome of the population at generation G is denoted by $X_{k,G}$ (where $X_{k,G} = [x_{1,k,G}, x_{2,k,G}, ..., x_{D,k,G}]$, D being the number of decision variables). The jth decision variable of the kth chromosome is randomly initialized for the initial population (at $G = 1$) as

$$x_{j,k,1} = x_{j,min} + rand_{k,j}[0, 1].(x_{j,max} - x_{j,min}) \tag{3.8}$$

where $x_{j,min}$ and $x_{j,max}$ are the minimum and maximum bounds of the jth decision variable, respectively and $rand_{k,j}[0, 1]$ is a uniformly distributed random number lying between 0 and 1 and is generated independently for each decision variable of the kth chromosome.

3.3.3 Fitness Evaluation

Since UC is a highly constrained optimization problem, the performance of the algorithm depends on how the algorithm handles the constraints.

Boundary Constraint Handling - The generator limit constraints given by (3.7) are handled according to the bound handling approach known as set on boundary [116]. According to this approach, if a continuous variable

Algorithm 3: Generation of UCM (Heuristic solution)

```
   input  : Pmax, LoadDemand, priority, N and T_max
   output : UCM
1  begin
2      Initialize the UCM: UCM ⟵ zeros(N, T_max) ;
3      for time = 1 : T_max do
4          P_max^total ⟵ 0 ;
5          for p = 1 : 10 do
6              for unit = 1 : N do
7                  if priority(unit) == p then
8                      UCM(unit, time) ⟵ 1 ;
9                      P_max^total ⟵ P_max^total + Pmax(unit) ;
10                     if P_max^total > LoadDemand(time) + Reserve(time) then
11                         Go to Step 16;
12                     end
13                 end
14             end
15         end
16     end
17 end
```

corresponding to power dispatch of a generator exceeds the bounds (during variation operation), then the variable is set on the boundary.

Load Demand Equality Constraint Handling - In hGADE, the other constraints (i.e., minimum up/down time and minimum spinning reserve constraints) except for the load demand equality constraint get adequately handled over the generations by the replacement mechanism based on pushing the algorithm towards the feasible search space (described later). Therefore, a repair operator is applied to repair chromosomes that violate the load demand equality constraint [13]. In the repair procedure, the chromosome is repaired for load demand equality constraint violation at hour t using priority list (PL) of the thermal units. If the total power output of the committed thermal units at hour t is less than the load demand on the system at hour t than the power output of the committed thermal units is increased in ascending order of the PL otherwise the power output of the committed thermal units is decreased in descending order of the PL to meet the load demand. It is always ensured that the power output of the thermal units does not violate the generation limits given by (3.7). Algorithm 5 in the Appendix A shows the pseudo-code of the repair procedure.

Constraint Violation Evaluation

At first, all the constraints are normalized because different constraints may take different orders of magnitude. An inequality constraint of the form $g(x) \geq b$ is normalized using the following transformation:

$$(g(x))/b - 1 \geq 0 \tag{3.9}$$

Equality constraints are also normalized similarly [29]. Thereafter, all normalized constraint violations are simply added to calculate the overall constraint violation of a chromosome. A chromosome is considered feasible if the overall constraint violation is less than the tolerance limit (10^{-6}).

Objective Function Evaluation

The objective function system operation cost is calculated for each chromosome using its Res.PM (which is obtained by multiplying the corresponding elements of UCM and RPM as mentioned earlier).

3.3.4 Selection Operation

In the hGADE framework, the inherent structure of standard GA and DE is maintained except for the replacement mechanism (discussed later). Thus, the selection operation by GA acting on the binary component of the chromosomes proceeds by binary tournament selection based on the feasibility rules [117] to form the mating pool while no mating pool is formed corresponding to DE acting on the continuous component of the chromosomes (as per the classical DE algorithm).

3.3.5 Variation Operation – Hybridization of GA with DE

The variation operation is the step in which GA and DE are hybridized at every generation. In the variation operation, the binary unit commitment variables are evolved using GA operators while the continuous power dispatch variables are evolved using DE operators as described below.

GA Operators on Binary Component (i.e., UCM) of the Parent Chromosomes

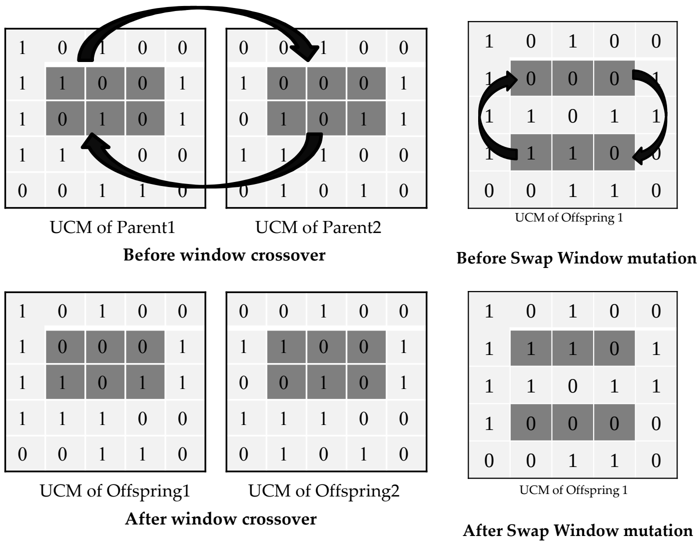
Since, in the hGADE algorithm, the binary variables are encoded in the form of matrix, problem-specific binary crossover and mutation operators which have been found in the literature to work well on matrix encodings have been adopted.

- *Window crossover* - A slightly modified version of the window crossover operator as mentioned in [12] is used as the binary crossover. It works by randomly selecting two parents and then randomly selecting a window size. The entries within the window portion are exchanged between the UCM of two parents to generate the UCM of two offspring. Fig. 3.3 (a) shows an example to illustrate how the window crossover works on a 5×5 UCM for a window size 2×3.

- *Swap window mutation* - It works on the UCM of a chromosome by randomly selecting: a) two units b) a time window of width w between 1 and T_{max} and c) a window position. The entries of the two units included in the window are then exchanged. This acts like a sophisticated mutation operator [9]. Fig. 3.3 (b) shows an example to illustrate how the swap window mutation works on a 5×5 UCM for a window size 1×3.

- *Window mutation* - This operator works on the UCM of a chromosome by randomly selecting: a) a unit, b) a time window of width w between 1 and T_{max} and c) a window position. Thereafter, it mutates all the bits included in the window, turning all of them to either 1's or 0's with an equal probability [9].

It is noted that both swap window mutation and window mutation operators are different from traditional binary flip GA mutation operator in which probabilistically a "0" is changed to "1" and vice-versa because the traditional GA mutation operator is not found to work well on matrix encodings for the UC problem [9].



(a) Window crossover

(b) Swap window operator

Fig. 3.3 Pictorial depiction of GA Operators on UCM of Parent Chromosomes.

DE Operators on Continuous Component (i.e., RPM) of the Parent Chromosomes

- *Mutation* - Corresponding to RPM of kth chromosome at generation G, $X_{k,G}$ (called target chromosome in DE literature) in the population, DE creates a mutant chromosome $V_{k,G}$ (where $V_{k,G}$ =

$[v_{1,k,G}, v_{2,k,G}, ..., v_{D,k,G}]$, D being the number of decision variables) through mutation. There are several DE variants in the literature and they differ mainly in the way mutation operation is executed. In this work, GA has been hybridized with 4 classical DE variants and 2 state-of-the-art self-adaptive DE variants. The equations for mutation operation corresponding to two of the classical DE variants [68, 74] employed in this work are as follows

$$DE/rand/1: \; V_{k,G} = X_{r_1^k,G} + F(X_{r_2^k,G} - X_{r_3^k,G}) \tag{3.10}$$

$$DE/current-to-rand/1: \quad V_{k,G} = X_{k,G} + F(X_{r_1^k,G} - X_{k,G}) \quad + \\ F(X_{r_2^k,G} - X_{r_3^k,G}) \tag{3.11}$$

where r_1^k, r_2^k and r_3^k are mutually exclusive and randomly chosen indices from $[1, NP]$ and are also different from the base index k (where NP is the population size). The scaling factor F is a control parameter for amplifying the difference of two chromosomes (for example the difference $(X_{r_2^k,G} - X_{r_3^k,G})$ in a vector sense) and lies in the range $[0, 2]$. A smaller value of F promotes exploitation while a larger value of F promotes exploration [60].

- *Crossover* – After generating the mutant chromosome $V_{k,G}$ through mutation, a crossover operation comes into play to further enhance the potential diversity of the population. In crossover, the mutant chromosome $V_{k,G}$ exchanges its components with the target chromosome $X_{k,G}$ with a probability $CR \in [0, 1]$ to form the trial chromosome $U_{k,G}$ (where $U_{k,G} = [u_{1,k,G}, u_{2,k,G}, ..., u_{D,k,G}]$, D being the number of decision variables). Although two crossover methods – *exponential* and *binomial* exist for the DE algorithm, the more preferred one is binomial crossover which has been adopted in hGADE algorithm as well. In binomial crossover, each component of the trial chromosome $U_{k,G}$ is inserted from either mutant chromosome or target chromosome according to the following condition:

$$u_{j,k,G} = \begin{cases} v_{j,k,G} & if\ (rand_{k,j}[0,1] \leq CR \text{ or } j = j_{rand}) \\ x_{j,k,G} & otherwise \end{cases} \tag{3.12}$$

where $rand_{k,j}[0,1]$ is a uniformly distributed random number and $j_{rand} \in [1, 2, ..., D]$ is a randomly chosen index which ensures that the trial chromosome gets at least one component from the mutant chromosome. It is noted that in DE/current-to-rand/1 and DE/current-to-rand/2, CR is always set at 1.0 as in these DE variants, the mutant chromosome becomes the offspring chromosome and there is no crossover between target chromosome and mutant chromosome.

3.3.6 Replacement

It has been shown in the literature that preserving infeasible solutions in the population of EAs can lead to an improved convergence as well as an improved convergence rate [118] for both single-objective and multi-objective constrained optimization problems. The need to preserve infeasible solutions in the population of EAs is also discussed in [119, 120] to preserve diversity, decrease the selection pressure and prevent premature convergence. Since, UC is a highly-constrained optimization problem, in this work, a replacement strategy which is based on preservation of infeasible solutions is implemented. Algorithm 4 shows the pseudo-code of the replacement procedure. According to this procedure:

- In the scenario when the number of feasible solutions in the combined parent and offspring population is less than the population size - then all the feasible solutions and the best infeasible solutions (in terms of lower total constraint violation) enter the next population. This condition ensures that the algorithm is pushed towards the feasible search space. In this manner, the replacement strategy is similar to the strategy based on superiority of feasible solutions [117].

- However, in the scenario when the the number of feasible solutions in the combined parent and offspring population is more than the population size - then the solutions with the best objective function values enter the next population. This condition ensures that fitter infeasible solutions (i.e., with better objective function values) are also preserved in the next population along with fitter feasible solutions. In this manner, the replacement strategy is different from the strategy based on superiority of feasible solutions [117].

Algorithm 4: Pseudocode of the replacement mechanism

```
input  : Parent population, Offspring population, popsize
output : Population of next generation
begin
    Combine the Parent population and Offspring population to form
    Combined population;
    Find the number of feasible solutions (Fnum) in Combined population;
    if Fnum < popsize then
        Preserve all the Fnum feasible solutions in the next population;
        Sort the remaining infeasible solutions in ascending order of total
        constraint violation values;
        Preserve (popsize − Fnum) number of infeasible solutions with lowest
        total constraint violation values.
    else
        Sort the Combined population in ascending order of objective
        function values;
        Select popsize number of solutions with lowest objective function values;
    end
end
```

Further, the replacement strategy implemented is quite similar to the replacement strategy presented in [118]. The only difference lies in the aspect that the user is not required to set a parameter corresponding to the infeasibility ratio to be maintained in the population. Thus, the replacement strategy implemented in hGADE algorithm is a parameter-less strategy unlike the one proposed in [118] and is more user-friendly. However, it is noted that the hGADE framework is flexible and other constraint handling techniques like for example, superiority of feasible solutions [117], ensemble of constraint handling techniques [121], may also be incorporated.

3.3.7 Termination Condition

For the hGADE algorithm, two termination conditions as shown below are employed. The algorithm is terminated if any of the termination condition is met.

- Condition 1 - If for 500 consecutive generations, the objective function of the best solution found so far does not improve by $ 5.
- Condition 2 - If the maximum allowed generation as summarized in Table 3.1 is reached.

Table 3.1 Maximum allowed generations for different test systems

Test System	10	20	40	60	80	100
Maximum Generations	3000	4000	6000	8000	8000	9000

3.4 Classification of the Proposed Hybrid Optimizer

In this Section, the proposed hybrid optimizer hGADE is classified according to a recently presented taxonomy [109] in which all the existing hybrid DEPSO algorithms in the literature were reviewed and classified. In [109], various hybridization factors were presented which can not only help the readers to better understand the hybridization strategy but can also act as reference for the interested researchers to design hybrid optimizers. The hGADE algorithm is classified according to various hybridization factors below.

- *Parent Relationship (PR)* - The relationship between the parent optimizers (i.e., the optimizers hybridized) can be of three types: 1) collaboration-based, 2) embedding-based, or 3) assistance-based. In collaboration-based relationship, the parents optimizers co-operate with each other in the search space, share or exchange accumulated information and their own operating steps in generating new sampling points in the search space are maintained. However, in the embedding-based relationship, the operating steps of the parent optimizers are changed and cannot be separated explicitly. Further, in the assistance-based relationship, one parent optimizer does not generate new sampling points in the search space and only acts as an assistant to the other parent optimizer.

  The relationship between GA and DE in hGADE is collaboration-based because at every generation, GA and DE work independently on binary (i.e., UCM) and continuous component (i.e., RPM) of the chromosomes, respectively and thus their own operating manners are maintained. Further, at every generation, once the variation operation of GA and DE are over, they share with each other the new information i.e., UCM and RPM of offspring. Thus, cooperation exists between GA and DE in the search space to seek the optimum solution.

- *Hybridization Level (HL)* - The hybridization level of a hybrid optimizer is highly dependent on the operating level (OL) of the parent optimizers. The OL of a parent optimizer refers to the level at which the parent optimizer acts and can be of four types: 1) component level, 2) individual level, 3) sub-population level, or 4) population level. If the parent optimizers work at the same OL, the hybridization is termed as homogeneous-level hybridization (HOLH) and the OL of the parent optimizers is termed as the HL of the hybrid optimizer. In contrast, if the parent optimizers work at different OLs, the hybridization is referred to as heterogeneous-level hybridization and the hybridization level between two parent optimizers is determined by the lower OL of parent optimizers.

Since, in hGADE, the binary component of the chromosomes are evolved using GA and the continuous component of the chromosomes are evolved using DE, the OL of both GA and DE is component level. Thus, homogeneous level hybridization exists in hGADE and the HL of hGADE is also component level.

- *Operating Order (OO)* - The operating order of the parents optimizers can be either - 1) sequential (alternate) or 2) parallel. In sequential OO type, the parent optimizers are applied one after another (i.e., sequentially), each working on the output of the previous. In contrast, in parallel OO type, the parent optimizers can act in a parallel fashion at their operating levels.

  In hGADE, at every generation, GA and DE act independently at the component level on binary component and continuous component of the chromosomes, respectively. Thus, the operating order in hGADE is parallel order. It is noted that although in this work GA and DE haven't been applied in parallel but the OO implies if parallel implementation of hybrid optimizers is possible or not.

- *Type of Information Transfer (TIT)* - The type of information transfer in a hybrid optimizer refers to the direction of information flow between two parent optimizers and can be either - 1) unidirectional (simplex TIT) or 2) bidirectional (duplex TIT). In hGADE, there exists duplex TIT because at every generation, once the variation operation of GA and DE are over, they share with each other the new information i.e., UCM and RPM of offspring and thereafter the offspring chromosomes undergo fitness evaluation.

- *Type of Transferred Information (TTI)* - In hybrid optimizers, the transferred information between parent optimizers can be of different types. For example, TTI can be 1) solutions (e.g., gbest or pbest if PSO is one of the parent optimizer, target vector or trial vector if DE is one of the parent optimizer), 2) solution components, 3) control parameters, etc. In hGADE, the type of transferred information is solution components because (as explained above) at every generation, once the variation operation of GA and DE are over, the information regarding the binary components of offspring flow from GA to DE and similarly the information regarding the continuous components of offspring flow from DE to GA.

3.5 Experimental Study

In this Section, the performance of the proposed hGADE framework is exhaustively evaluated on the UC problem. The experimental evaluation is systematically divided into 7 case studies:

1. In the first case study, the effect of incorporating the replacement scheme based on preserving infeasible solutions is demonstrated on the quality of results as well as the computational efficiency in comparison to the replacement schemes of standard GA and DE;

2. In the second case study, the effect of incorporating heuristic initialization is demonstrated on both the quality of results as well as the computational efficiency as compared to random initialization;

3. In the third case study, GA is hybridized with four classical DE variants namely, DE/rand/1, DE/rand/2, DE/current-to-rand/1 and DE/current-to-rand/2 [68, 74]. In this case study, the parametric tuning of the proposed hGADE variants is conducted;

4. Thereafter, in the fourth case study, GA is hybridized with two state-of-the-art self-adaptive DE variants namely, jDE [69] and JADE [71];

5. In the fifth case study, the hGADE variants proposed in case study 3 and 4 are statistically compared among themselves to determine the best hGADE variants;

6. Further, in the sixth case study, the impact of hybridization between GA and DE is presented by comparing the performance of the best hGADE variants (determined in case study 5) against a GA based approach in which both binary variables as well as continuous variables are evolved using GA;

7. In the seventh case study, the best hGADE variants (found in case study 5) are benchmarked against the other approaches proposed in literature for the UC problem.

The proposed hGADE variants are tested on UC problem for power systems with 10, 20, 40, 60, 80 and 100 units in a 24 hour scheduling horizon [9]. The dimensions of the different test systems are summarized in Table 3.2. The spinning reserve requirements are assumed to be 10% of the load demand [13, 14]. The generating unit data and the forecasted load demand data for the 10 unit system is summarized in Table

A.1 and A.2, respectively in the Appendix A. The hGADE variants have been named for example - hGADE/rand/1, hGADE/current-to-rand/1, hGADE/jDE, hGADE/JADE if the existing DE strategy in the hGADE variant is DE/rand/1, DE/current-to-rand/1, jDE and JADE, respectively. This nomenclature for referring hGADE variants is used in the rest of this Chapter. Further, for the ease of readers, the hGADE variants have been assigned acronyms as summarized in Table 3.3.

Table 3.2 Dimensions of different test systems

System size	10	20	40	60	80	100
Binary Variables	240	480	960	1440	1920	2400
Continuous Variables	240	480	960	1440	1920	2400

Table 3.3 Acronyms corresponding to hGADE variants

hGADE/rand/1	hGADE/r1
hGADE/rand/2	hGADE/r2
hGADE/current-to-rand/1	hGADE/cur1
hGADE/current-to-rand/2	hGADE/cur2

For each experiment, 20 independent simulation trials are conducted to verify the effectiveness of the hGADE variants. However, experimental results corresponding to only two or three representative test systems (out of six test systems) are presented for different case studies except for benchmarking where results obtained on all the test systems are presented. To make the comparison fair, the populations for all the hGADE variants (over all the test systems tested) were initialized using the same random seeds.

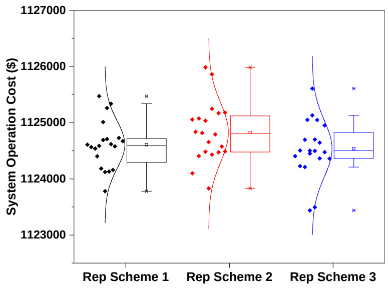
The initial experiments showed that (fixing F at 0.9 and CR at 0.9 for hGADE/r1 and fixing F at 0.9 and CR at 1.0 for hGADE/cur1), for hGADE/r1 and hGADE/cur1 variants, the GA crossover probability of 0.6 and the GA mutation probability of 0.25 worked well on all the test systems. Thus, the parameter settings of genetic operators i.e., (GA) crossover and mutation probabilities were kept fixed for all the simulation studies in this work as 0.6 and 0.25, respectively. It is noted that the GA mutation probability (of 0.25) is relatively high because the swap window and window mutation operators in hGADE probabilistically act on the entire (UCM part of the) chromosome unlike the standard flip (binary) GA mutation operator which probabilistically acts on each binary variable. The hGADE algorithm is developed on C++ platform and executed on PC with Intel Xeon 3.10 GHz processor and 4 GB memory.

3.5.1 Case Study 1 - Study on Efficacy of Replacement Scheme

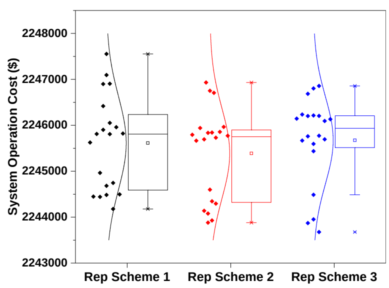
The replacement scheme in traditional DE is based on one to one comparison between the offspring and the parent solution and the fitter solution moving into the next generation. On the other hand, the replacement scheme in traditional GA is based on combination of offspring population and parent population and the fittest solutions entering into the next generation. In constrained optimization problems, both of these replacement schemes are generally based on the feasibility rules which always prefer a feasible solution over an infeasible solution [117]. However, as discussed earlier, the replacement scheme incorporated in the hGADE framework is based on preserving infeasible solutions. Thus, the effectiveness of the replacement scheme based on preserving infeasible solutions (named Rep Scheme 3) is investigated in this case study by comparing against the replacement schemes of traditional DE (Rep Scheme 1) and GA (Rep Scheme 2).

Effect on Quality of Results

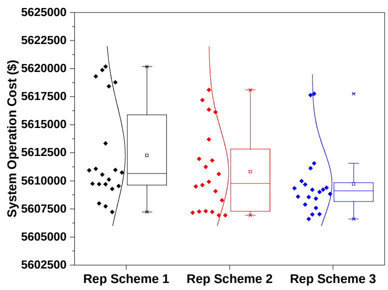
Fig. 3.4 illustrate the experimental results (corresponding to system operation cost) of 20 runs using box plots (along with distribution of solutions for better visualization) for hGADE/cur1 in presence of different replacement schemes. Fig 3.4 shows that hGADE/cur1 performed either comparable or better in presence of the replacement scheme based on preserving infeasible solutions as compared to the replacement schemes of DE and GA. For example, on the 40 unit system, the best cost obtained in presence of scheme 3 was \$ 502 and \$ 205 better than that obtained in presence of scheme 1 and 2, respectively while the average cost obtained was comparable. On the 100 unit system, the best cost obtained in presence of scheme 3 was \$ 625 and \$ 321 better than that obtained in presence of scheme 1 and 2, respectively. Further, the average cost obtained in presence of scheme 3 was \$ 2540 and \$ 1095 better than that obtained in presence of scheme 1 and 2, respectively.



(a) 20 unit system



(b) 40 unit system



(c) 100 unit system

Fig. 3.4 Comparison of replacement schemes with respect to system operation cost for hGADE/current-to-rand/1 on different test systems.

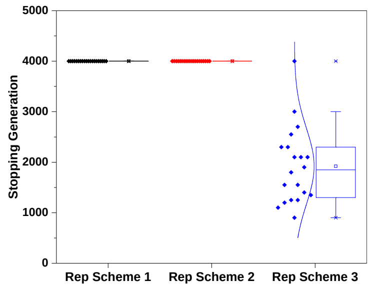
Effect on Computational Efficiency

To investigate if the proposed replacement scheme aids in improving the computational efficiency as well, experimental results (corresponding to stopping generation) of 20 runs for hGADE/cur1 were plotted using box plots in presence of different replacement schemes as shown in Fig. 3.5.

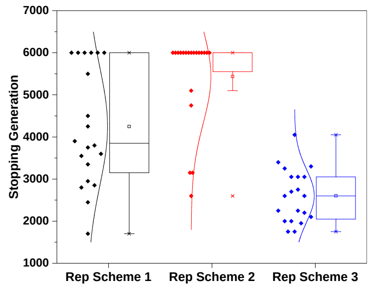
It is observed from the figure that hGADE/cur1 was able to converge remarkably faster in presence of scheme 3. For example, on the 20 unit system, in presence of scheme 1 and 2, hGADE/cur1 was not able to converge in a single run within maximum allowed generations (i.e., 4000). In contrast, in the presence of scheme 3, the average stopping generation was 1920. On the 40 unit system, hGADE/cur1 was not able to converge in some runs and most of the runs in maximum allowed generations (i.e., 6000) in presence of scheme 1 and 2, respectively. On the other hand, in presence of scheme 3, the average stopping generation of hGADE/cur1 was around 2600.

3.5.2 Case Study 2 - Study on Efficacy of Heuristic Initialization

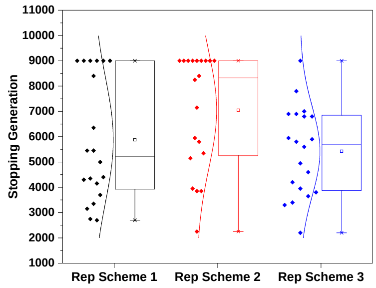
As mentioned earlier, a heuristic initialization strategy is incorporated to improve the effectiveness of the hGADE variants on the UC problem. To demonstrate this, experiments were conducted on different test systems by implementing hGADE/r1 and hGADE/cur1 in presence of random and heuristic initialization. It is noted that in this case study, the optimal parameter settings corresponding to F, CR and population size are set according to the experiments shown in the next case study.



(a) 20 unit system



(b) 40 unit system

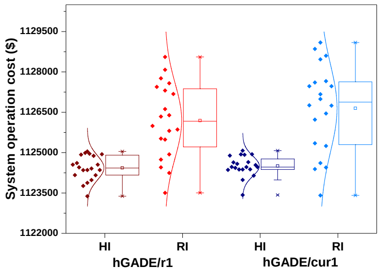


(c) 100 unit system

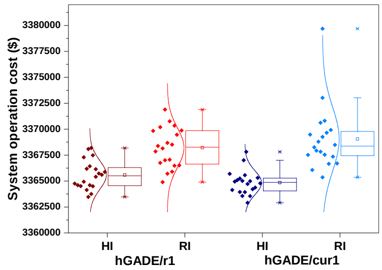
Fig. 3.5 Comparison of replacement schemes with respect to stopping generation for hGADE/current-to-rand/1 on different test systems.

Effect on Quality of Results

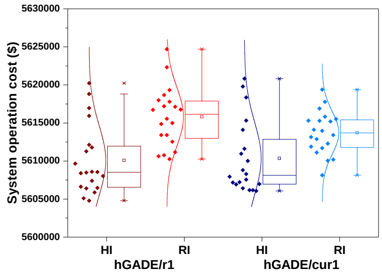
Fig. 3.6 illustrate the experimental results (corresponding to system operation cost) of 20 runs using box plots for hGADE/r1 and hGADE/cur1 in presence of random initialization (RI) and heuristic initialization (HI). It is observed that heuristic initialization had a remarkable effect on the quality of results by significantly reducing the best cost, mean cost, median and standard deviation (i.e., overall distribution of solutions) for both hGADE/r1 and hGADE/cur1. For example, considering hGADE/r1, as compared to random initialization, heuristic initialization could improve the mean cost by \$ 1755, \$ 4206 and \$ 3370 on 20, 60 and 100 unit system, respectively.



(a) 20 unit system



(b) 60 unit system

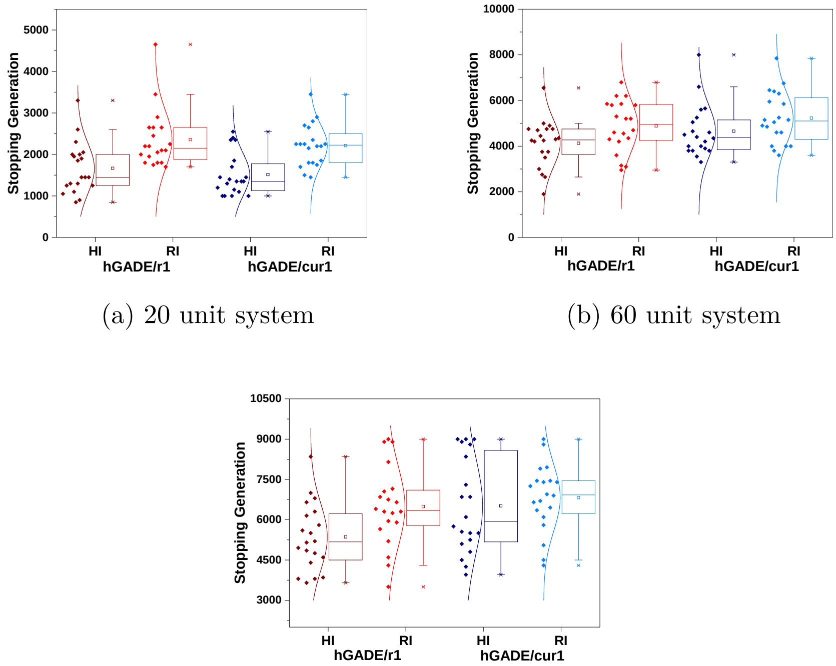


(c) 100 unit system

Fig. 3.6 Comparison of heuristic initialization (HI) and random initialization (RI) with respect to system operation cost for hGADE/rand/1 and hGADE/current-rand/1 on different test systems.

Effect on Computational Efficiency

To investigate if heuristic initialization helps in improving the computational efficiency as well in comparison to random initialization, experimental results with respect to stopping generation for 20 runs were plotted using box plots (refer Fig. 3.7) for hGADE/r1 and hGADE/cur1. Fig. 3.7 shows that heuristic initialization had an equally important effect on improving the computational efficiency as well in comparison to random initialization by significantly reducing the best, mean, median and standard deviation of stopping generation. For example, considering hGADE/r1, as compared to random initialization, heuristic initialization could reduce the mean stopping generation by 700, 760 and 1180 generations on 20, 60 and 100 unit system, respectively.



(a) 20 unit system

(b) 60 unit system

(c) 100 unit system

Fig. 3.7 Comparison of heuristic initialization (HI) and random initialization (RI) with respect to stopping generation for hGADE/rand/1 and hGADE/current-rand/1 on different test systems.

3.5.3 Case Study 3 - Study on Hybridization of GA with Classical DE variants

In this Section, a study on hybridizing GA with four classical DE variants namely, DE/rand/1, DE/rand/2, DE/current-to-rand/1 and DE/current-to-rand/2 is conducted and the optimal parameter settings F and CR are determined (for the DE operators acting on continuous variables) followed by determination of the optimal population size corresponding to each test system.

Determination of Parameter Settings F and CR

To determine the optimum values of scaling factor F and crossover probability CR, at first the hGADE/r1 variant was implemented on two test systems consisting of 20 and 40 units. The population size for 20 and 40 unit test systems was fixed as 200 and 300, respectively.

Initially for both the test systems, F was varied from 0.5 to 0.9 in steps of 0.2 and for each F, CR was also varied from 0.5 to 0.9 in steps of 0.2. Table 3.4 summarize the experimental results for hGADE/r1. It is observed from Table 3.4 that for hGADE/r1, for both the test systems and each value of scaling factor F, the results were best with CR at 0.9. It is also observed that with CR at 0.9, the results for both the test systems were best with F at 0.9.

Table 3.4 Effect of F and CR on hGADE/rand/1

		20 Unit System		
F	CR	Best Cost ($)	Avg. Cost ($)	Worst Cost ($)
	0.5	1,124,034	1,124,985	1,126,083
0.5	0.7	1,124,464	1,124,822	1,125,285
	0.9	**1,123,669**	**1,124,401**	**1,125,205**
	0.5	1,124,003	1,12,746	1,125,241
0.7	0.7	1,124,033	1,124,850	1,125,718
	0.9	**1,123,517**	**1,124,481**	**1,125,082**
	0.5	1,124,006	1,124,705	1,125,299
0.9	0.7	1,123,703	1,124,510	1,125,101
	0.9	**1,123,432**	**1,124,427**	**1,125,130**
		40 Unit System		
F	CR	Best Cost ($)	Avg. Cost ($)	Worst Cost ($)
	0.5	2,244,660	2,246,440	2,247,500
0.5	0.7	2,244,788	2,246,379	2,247,460
	0.9	**2,244,423**	**2,246,183**	**2,247,819**
	0.5	2,244,800	2,246,325	2,247,485
0.7	0.7	2,244,795	2,246,440	2,248,129
	0.9	**2,244,415**	**2,245,807**	**2,249,350**
	0.5	2,244,782	2,245,958	2,246,606
0.9	0.7	2,244,526	2,246,237	2,248,955
	0.9	**2,244,409**	**2,245,763**	**2,246,409**

Although, Storn and Price suggested in [60] that F can be in the range [0, 2] yet to the best of our knowledge, there have been hardly any studies in which $F \geq 1$ has been found to perform well. This may be because more number of numerical optimization studies have been conducted as compared to optimization studies on real-world problems and another reason may be that many researchers tend to utilize the parameter settings suggested by other research works. However, as among 0.5, 0.7 and 0.9, F at 0.9 produced the best results, F was varied further from 1.1 to 1.5 in steps of 0.2 while CR was fixed at 0.9 for hGADE/r1 and hGADE/r2 and 1.0 for hGADE/cur1 and hGADE/cur2. The reason behind trying $F > 1$ is that higher values of scaling factor F promote exploration and may prevent the algorithm from converging prematurely.

Tables 3.5, 3.6, 3.7 and 3.8 summarize the experimental results (on 40 and 60 unit test systems) for variation of scaling factor F from 0.9 to 1.3 in steps of 0.2 for hGADE/r1, hGADE/cur1, hGADE/r2 and hGADE/cur2, respectively. It is noted that the results with scaling factor F at 1.5 were not promising and hence have not been presented here.

Table 3.5 Effect of F ON hGADE/rand/1

		40 Unit System		
F	CR	Best cost ($)	Avg. cost ($)	Worst cost ($)
0.9	0.9	2,244,409	2,245,763	2,246,409
1.1	0.9	2,243,904	2,245,636	2,246,867
1.3	**0.9**	**2,243,724**	**2,245,582**	**2,247,130**
		60 Unit System		
F	CR	Best cost ($)	Avg. cost ($)	Worst cost ($)
0.9	0.9	3,363,823	3,365,744	3,367,551
1.1	0.9	3,363,610	3,365,632	3,369,860
1.3	**0.9**	**3,363,470**	**3,365,587**	**3,368,196**

It is observed from Table 3.5 and 3.6 that for hGADE/r1 and hGADE/cur1, on both the test systems, F at 1.3 produced the best results while Table 3.7 and 3.8 show that for hGADE/r2 and hGADE/cur2, F at 0.9 produced the best results. Thus, our intuition that higher values of scaling factor F i.e., $F > 1$ may work well turned out to be correct for hGADE/r1 and hGADE/cur1. However, the reason F at 0.9 turned out to be better than F at 1.3 for hGADE/r2 and hGADE/cur2 may be the fact that hGADE/r2

Table 3.6 Effect of F on hGADE/current-to-rand/1

		40 Unit System		
F	CR	Best cost ($)	Avg. cost ($)	Worst cost ($)
0.9	1	2,243,855	2,245,569	2,247,512
1.1	1	2,243,679	2,245,400	2,248,854
1.3	**1**	**2,243,522**	**2,245,321**	**2,246,540**
		60 Unit System		
F	CR	Best cost ($)	Avg. cost ($)	Worst cost ($)
0.9	1	3,363,086	3,365,259	3,367,143
1.1	1	3,363,094	3,365,140	3,368,515
1.3	**1**	**3,362,908**	**3,364,841**	**3,367,820**

Table 3.7 Effect of F on hGADE/rand/2

		40 Unit System		
F	CR	Best cost ($)	Avg. cost ($)	Worst cost ($)
0.9	**0.9**	**2,243,809**	**2,245,627**	**2,247,095**
1.1	0.9	2,244,133	2,245,921	2,247,417
1.3	0.9	2,244,221	2,246,239	2,248,250
		60 Unit System		
F	CR	Best cost ($)	Avg. cost ($)	Worst cost ($)
0.9	**0.9**	**3,363,225**	**3,365,641**	**3,368,207**
1.1	0.9	3,363,644	3,366,676	3,368,789
1.3	0.9	3,367,088	3,371,279	3,377,343

and hGADE/cur2 are more explorative than hGADE/r1 and hGADE/cur1, respectively and thus higher value of F may not be required.

Based on the observation from the experimental results, the optimum parameter settings common for all the test systems and employed for hybrid of GA and classical DE variants is summarized in Table 3.9.

Determination of Population Size

The population size was determined for the hGADE variants through experiments by implementing hGADE/r1 and hGADE/cur1 on different test systems. Fig 3.8 illustrate the experimental results using box plots for hGADE/r1 and hGADE/cur1 on selected test systems.

Tests were first carried out on 10 unit and 20 unit system considering population size of 100 and 200. It was observed that for both the test systems, better results were obtained with population size of 200. Thereafter, for 40 unit system, tests were carried out considering population size of 200 and 300 and it was observed that better results were obtained with population size of 300. Thus, as the system size i.e., dimensionality of the search space increased, the population size requirement also increased to 300. Further, for 60, 80 and 100 unit system, tests were carried out considering population size of 300 and 400. It was observed that in all the three test systems, population size of 400 performed better. Thus, as the system size increased, the population size requirement further increased to 400. The optimum population size obtained for different test systems and fixed for all the hGADE variants in this work is summarized in Table 3.10.

Table 3.8 Effect of F on hGADE/current-to-rand/2

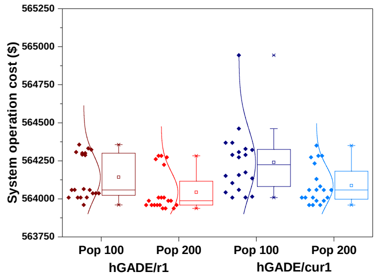
		40 Unit System		
F	CR	Best cost ($)	Avg. cost ($)	Worst cost ($)
0.9	**1.0**	**2,243,556**	**2,245,841**	**2,248,173**
1.1	1.0	2,243,609	2,245,971	2,247,368
1.3	1.0	2,244,575	2,246,560	2,247,655
		60 Unit System		
F	CR	Best cost ($)	Avg. cost ($)	Worst cost ($)
0.9	**1.0**	**3,363,067**	**3,365,128**	**3,366,851**
1.1	1.0	3,366,857	3,371,039	3,378,546
1.3	1.0	3,371,389	3,377,653	3,382,524

Table 3.9 Optimum parameter settings for hybrid of GA and classical DE variants

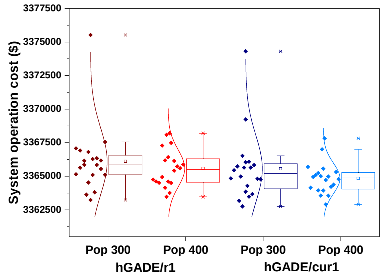
GA crossover rate	0.6
GA mutation rate	0.25
DE mutation rate (F)	1.3 (hGADE/r1, hGADE/cur1) 0.9 (hGADE/r2 and hGADE/cur2)
DE crossover rate (CR)	0.9 (hGADE/r1, hGADE/r2) 1.0 (hGADE/cur1 and hGADE/cur2)

Table 3.10 Optimum population size for different test systems

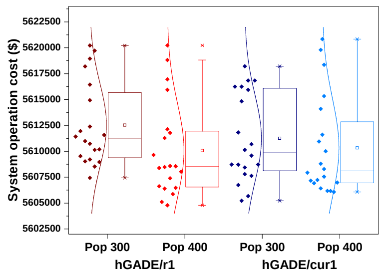
System size	10	20	40	60	80	100
Pop. Size	200	200	300	400	400	400



(a) 10 unit system



(b) 60 unit system



(c) 100 unit system

Fig. 3.8 Effect of population size with respect to system operation cost on hGADE/rand/1 and hGADE/current-rand/1 for different test systems.

3.5.4 Case Study 4 - Study on Hybrid of GA and Self-adaptive DE variants

In this case study, the flexibility of the proposed hGADE framework is demonstrated by substituting classical DE variants with self-adaptive DE variants in the framework. Here, GA is hybridized with two state-of-the-art self-adaptive DE variants namely, jDE [69] and JADE [71].

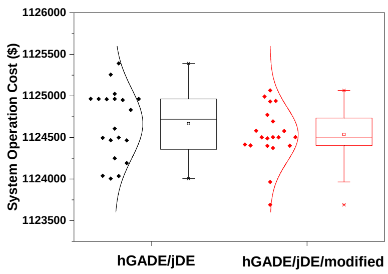
Study on Hybrid of GA and jDE

In the original study on jDE [69], it was observed that jDE is not sensitive to the initial F and CR values. However, it was mentioned that if suppose it is known that CR at 0.95 is good for the test problem at hand, then this knowledge may be utilized in the initialization. Since, the DE variant present in jDE is DE/rand/1 and it was observed in the parameter tuning study (presented in case study 3) that higher values of F and CR work better on the UC problem; an investigation was undertaken by modifying the initial F and CR values and the parameter range corresponding to self-adaptation for jDE. Thus, hGADE/jDE and hGADE/jDE/modified were put to comparison with the parameter settings as shown in Table 3.11.

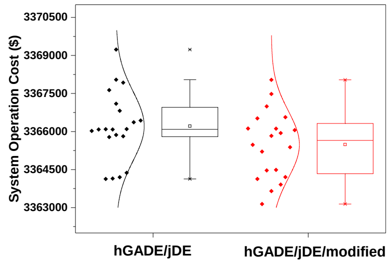
Table 3.11 Parameter settings for hGADE/jDE variant

hGADE Variant	F initial	CR initial	F new range	CR new range
hGADE/jDE	0.5	0.9	[0.1, 1.0]	[0, 1.0]
hGADE/jDE/modified	0.9	0.9	[0.7, 1.3]	[0.7, 1.0]

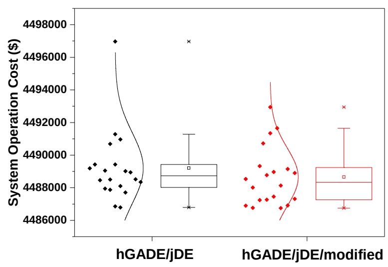
Fig. 3.9 illustrates the experimental results of 20 runs using box plots for hGADE/jDE and hGADE/jDE/modified. It is observed from Fig. 3.9 that hGADE/jDE/modified resulted in better distribution of solutions than hGADE/jDE. Therefore, hGADE/jDE/modified is selected for benchmarking (presented later) and is referred to as hGADE/jDE in the rest of the Chapter.



(a) 20 unit system



(b) 60 unit system



(c) 80 unit system

Fig. 3.9 Effect of parameter setting modification with respect to system operation cost on hGADE/jDE variant for different test systems.

Study on Hybrid of GA and JADE

As with jDE, in the original study on JADE [71], it was observed that JADE is not very sensitive to the initial μ_F and μ_{CR} values and an initial setting of $\mu_F = \mu_{CR} = 0.5$ was found to work well for all the standard test functions. However, as the modification in initial setting and the parameter range corresponding to self-adaptation worked well for hGADE/jDE, a similar investigation was undertaken for hGADE/JADE as well. Thus, hGADE/JADE and hGADE/JADE/modified were put to comparison with the parameter settings as shown in Table 3.12.

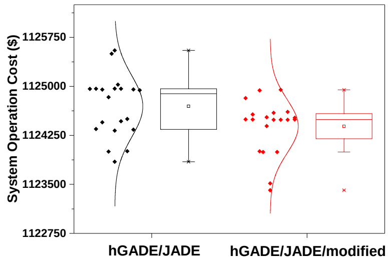
Table 3.12 Parameter settings for hGADE/JADE variant

hGADE Variant	μ_F initial	μ_{CR} initial	F new upper limit
hGADE/JADE	0.5	0.5	1.0
hGADE/JADE/modified	0.9	0.9	1.3

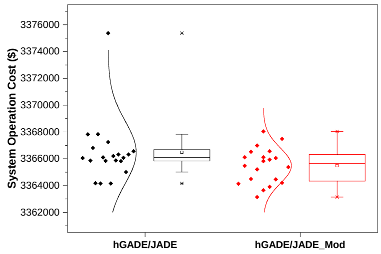
Fig. 3.10 illustrates the experimental results of 20 runs using box plots for hGADE/JADE and hGADE/JADE/modified. Fig. 3.10 shows that as observed in the hybrid of GA and jDE study, hGADE/JADE/modified also resulted in better distribution of solutions than hGADE/JADE. Therefore, hGADE/JADE/modified is selected for benchmarking (presented later) and is referred to as hGADE/JADE in the rest of the Chapter.

3.5.5 Case Study 5 - Comparison of hGADE variants Among Themselves

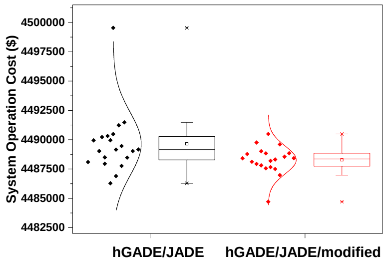
In this case study, to determine the best hGADE variants among the hybrid of GA and classical DE variants (presented in case study 3) and the hybrid of GA and self-adaptive DE variants (presented in case study 4), the hGADE variants were statistically compared among themselves. Friedman's test which can be employed for multiple comparisons [122] was ap-



(a) 20 unit system



(b) 60 unit system



(c) 80 unit system

Fig. 3.10 Effect of parameter setting modification with respect to system operation cost on hGADE/JADE variant for different test systems.

plied to the hGADE variants. Table 3.13 summarizes the individual ranking and the p-value obtained on different test systems while Table 3.14 summarizes the overall ranking across the 6 test systems obtained in terms of solution quality by the hGADE variants.

Table 3.13 Results obtained through Friedman's test for hGADE variants with respect to quality of results on different test systems

hGADE	10-unit system		20-unit system		40-unit system		60-unit system		80-unit system		100-unit system	
variant	Rank	p-value	Rank	p-value	Rank	p-value	Rank	p-value	Rank	p-value	Rank	p-value
hGADE/r1	2.85		3.2		3.35		3.9		3.9		3.25	
hGADE/r2	3.05		4.15		3.85		3.9		4.1		3.5	
hGADE/cur1	3.9	0.16929	3.25	0.525	2.85	0.16594	2.55	0.16102	2.5	0.10681	3.55	0.517
hGADE/cr2	4.25		3.6		3.75		3.5		3.7		3	
hGADE/jDE	3.4		3.65		4.2		3.85		3.45		4.15	
hGADE/JADE	3.55		3.15		3		3.3		3.35		3.55	

It is observed from Table 3.13 that according to the Friedman's test, at the 0.05 significance level, there were no significant differences among

Table 3.14 Overall rankings obtained through Friedman's test for hGADE variants with respect to quality of results for different test systems

hGADE variant	Overall rank
hGADE/r1	**20.45**
hGADE/r2	22.55
hGADE/cur1	**18.6**
hGADE/cr2	21.8
hGADE/jDE	22.7
hGADE/JADE	**19.9**

the hGADE variants on any of the test systems. This demonstrates the robustness of the hybridization strategy. However, according to the overall Friedman ranking across all the test systems summarized in Table 3.14, it is observed that hGADE/cur1, hGADE/JADE and hGADE/r1 are the best hGADE variants. Thus, these hGADE variants were selected for further comparisons in the rest of this Chapter.

3.5.6 Case Study 6 - Study on Efficacy of the Proposed Hybridization Strategy

To investigate the efficacy of the proposed hybrid GA-DE framework, it is essential to statistically compare the performance of the hGADE variants against a GA based approach in which both the binary as well as the continuous variables are evolved using GA. Thus, in this study, hGADE variants found to be the best in the previous case study i.e., hGADE/cur1, hGADE/JADE and hGADE/r1 were compared against a GA based approach in which the variation operators on binary variables i.e., window crossover, swap window mutation and mutation operator remain the same. However, the variation operators employed for evolving continuous variables are SBX crossover [114] and polynomial mutation [29] operator. The optimal parameters obtained through experiments corresponding to variation operators in GA based approach are summarized in Table 3.15.

Table 3.15 Optimum parameter settings for GA

η_c	η_m	Pcross_real	Pmut_real
5	10	0.6	0.1

It is noted that the rest of the parameters like crossover and mutation probabilities corresponding to variation operators acting on binary variables, population size corresponding to different test systems and termination condition remain the same for GA as set for hGADE variants. Further, the heuristic initialization is applied to GA as well. Moreover, in order to have a fair comparison, 2 variants of GA- a) GA-2 and b) GA-3 are implemented. GA-2 variant adopts the replacement scheme 2, i.e., the traditional scheme of GA while GA-3 variant adopts the replacement scheme based on preserving infeasible solutions i.e., scheme 3 (as discussed earlier).

At first, the Friedman's test was applied to statistically compare the performance of the hGADE variants and the GA variants with respect to the quality of solution and the results obtained are summarized in Table 3.16. It is observed from the p-values corresponding to the different test systems in the Table 3.16 that although on 10, 20 and 40 unit system, the algorithms under comparison are not statistically different at the 0.05 significance level but there is significant difference among the contender algorithms on larger systems i.e., 60, 80 and 100-unit. It is noted that in the previous case study, it was observed that according to the Friedman's test, the hGADE variants are not statistically different at the 0.05 significance level. This indicates that it is one or more of the hGADE variants which is/are statistically different from the GA variants.

Further, the overall ranking obtained through the Friedman's test for

Table 3.16 Results obtained through Friedman's test for hGADE variants and GA with respect to quality of results on different test systems

hGADE	10-unit system		20-unit system		40-unit system		60-unit system		80-unit system		100-unit system	
variant	Rank	*p*-value	Rank	*p*-value	Rank	*p*-value	Rank	*p*-value	Rank	*p*-value	Rank	*p*-value
hGADE/r1	2.8		2.75		3.35		2.95		2.95		2.2	
hGADE/cur1	3		3.15		2.95		2.05		2.05		2.55	
hGADE/JADE	3.05	0.3041	2.85	0.1712	3.05	0.2052	2.4	**4.24E-04**	2.3	**1.34E-04**	2.55	**4.40E-04**
GA-2	3.6		3.7		3.35		3.8		3.75		4	
GA-3	2.55		2.55		2.3		3.8		3.95		3.7	

hGADE variants and the GA variants across all the test systems in summarized in Table 3.17. The overall ranking shows that the three hGADE variants outperform the two GA variants. Thus, the Friedman's test comparison validates the efficacy of the proposed hybridization strategy. Further, the GA variant with the adopted replacement scheme i.e., GA-3 scored better rank than the GA variant with the traditional replacement scheme i.e., GA-2. This further signifies the efficacy of the replacement scheme based on preserving infeasible solutions.

Table 3.17 Overall rankings obtained through Friedman's test for hGADE variants and GA with respect to quality of results on different test systems

hGADE variant	Overall rank
hGADE/r1	**17**
hGADE/cur1	**15.75**
hGADE/JADE	**16.2**
GA-2	22.2
GA-3	18.85

However, to further investigate which of the hGADE variants are statistically different with respect to the GA variants on the larger system, Wilcoxon signed rank test [122] is applied to each of the hGADE variants and the two GA variants. Table 3.18, 3.19 and 3.20 summarize the results obtained through Wilcoxon signed rank test for hGADE/r1 and GA variants, hGADE/cur1 and GA variants and hGADE/JADE and GA variants, respectively.

Table 3.18 Results obtained through Wilcoxon signed rank test between hGADE/r1 variant and GA with respect to quality of results on different test systems

hGADE/r1	**60-unit system**			**80-unit system**			**100-unit system**		
v.s.	**R+**	**R-**	***p-value***	**R+**	**R-**	***p-value***	**R+**	**R-**	***p-value***
GA-2	139	71	0.216	149	61	0.105	187	23	**0.001**
GA-3	143	67	0.165	164	46	**0.027**	177	33	**0.006**

Table 3.19 Results obtained through Wilcoxon signed rank test between hGADE/cur1 variant and GA with respect to quality of results on different test systems

hGADE/cur1	**60-unit system**			**80-unit system**			**100-unit system**		
v.s.	**R+**	**R-**	***p-value***	**R+**	**R-**	***p-value***	**R+**	**R-**	***p-value***
GA-2	170	40	**0.014**	172	38	**0.01**	168	42	**0.017**
GA-3	181	29	**0.003**	179	31	**0.004**	171	39	**0.012**

It is observed from the *p*-values in the Table 3.18 that hGADE/r1 is statistically superior to both the GA variants on only 100 unit system. However, the *p*-values in the Tables 3.19 and 3.20 show that at the 0.05 level of significance, both hGADE/cur1 and hGADE/JADE are significantly superior to the two GA variants on 60, 80 and 100-unit system. The Wilcoxon signed rank test comparison further confirms the efficacy of the proposed hybridization strategy.

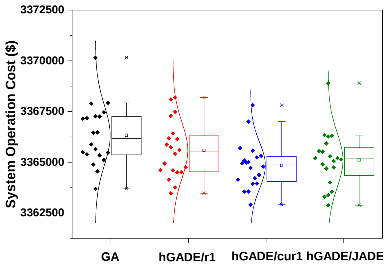
Next, the 3 best hGADE variants i.e., hGADE/cur1, hGADE/JADE and hGADE/r1 are compared with GA-3 variant to further (graphically) demonstrate the effect of hybridization on the quality of the results and the computational efficiency. It is noted that the GA-3 variant is chosen for further comparison and analysis as it was found to be better than the GA-2 according to the Friedman ranking.

Effect of Hybridization on Quality of Results

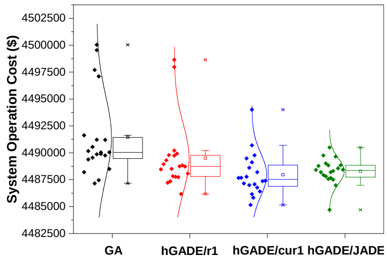
Fig 3.11 illustrate the experimental results using box plots for comparison between GA, hGADE/r1, hGADE/cur1 and hGADE/JADE. It is observed from Fig. 3.11 that in comparison to GA, all the 3 hGADE variants were

Table 3.20 Results obtained through Wilcoxon signed rank test between hGADE/JADE variant and GA with respect to quality of results on different test systems

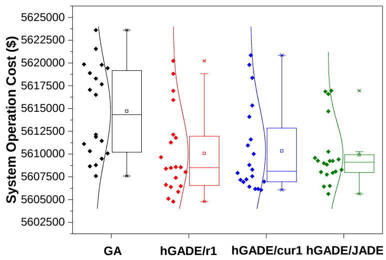
hGADE/JADE	**60-unit system**			**80-unit system**			**100-unit system**		
v.s.	**R+**	**R-**	***p-value***	**R+**	**R-**	***p-value***	**R+**	**R-**	***p-value***
GA-2	173	37	**0.009**	187	23	**0.001**	182	28	**0.003**
GA-3	174	36	**0.008**	201	9	**6.29E-05**	178	32	**0.005**



(a) 60 unit system



(b) 80 unit system



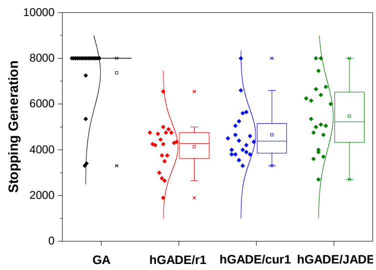
(c) 100 unit system

Fig. 3.11 Comparison of GA, hGADE/r1, hGADE/cur1 and hGADE/JADE with respect to system operation cost for different test systems.

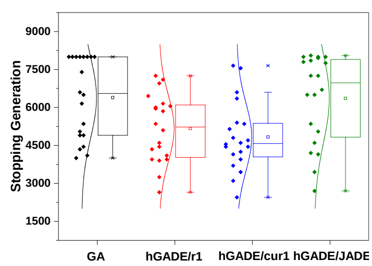
able to obtain significantly lower best cost, mean cost, median and standard deviation on different test systems. For example, as compared to GA, hGADE/cur1 could improve the best cost (and mean cost) by \$ 779 (and \$ 1492), \$ 2004 (and \$ 3482) and \$ 1520 (and \$ 4369) on 60, 80 and 100 unit system, respectively. This shows that the effect of the proposed hybridization strategy between GA and DE is remarkable on improving the quality of results as compared to GA.

Effect of Hybridization on Computational Efficiency

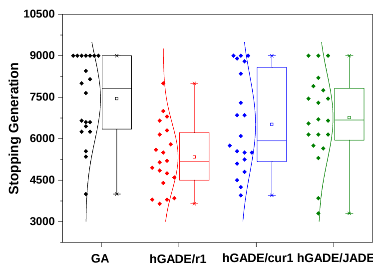
To investigate if the proposed hybridization strategy between GA and DE helps in improving the computational efficiency as well in comparison to the GA based approach, experimental results of stopping generation for 20 runs were plotted using box plots (refer Fig. 3.12) for GA, hGADE/r1, hGADE/cur1 and hGADE/JADE.



(a) 60 unit system



(b) 80 unit system



(c) 100 unit system

Fig. 3.12 Comparison of GA, hGADE/r1, hGADE/cur1 and hGADE/JADE with respect to stopping generation for different test systems.

Fig 3.12 shows that the proposed hybridization had an equally important effect on improving the computational efficiency as well in comparison to GA. For example, in almost all the runs on 60 unit system and many of the runs on 80 and 100 unit system, GA was not able to converge in the maximum allowed generations i.e., 8000, 8000 and 9000 generations, respectively. The mean stopping generation for GA and hGADE/cur1 on 60 unit system was 7365 and 4650, respectively; on 80 unit system was 6388 and 4833, respectively; and on 100 unit system was 7488 and 6515, respectively.

3.5.7 Case Study 7 - Comparison of hGADE variants against other benchmarks

The case study 5 revealed that the three hGADE variants with the best mean ranks (with respect to Friedman's test) for solving the UC problem are hGADE/cur1, hGADE/JADE and hGADE/r1. Thus, in this case study, the performance of these three hGADE variants is benchmarked against several algorithms, namely, EP [11], MA [12], EPSO [14], SA [15], QEA [16], DE [17], GA [10], BGSA [20], PL [4], LR [8] and DP [5]. It is noted that in order to have a thorough comparison, the benchmark algorithms are selected from both category 1 and 2 discussed in Section 3.2 (related work). The benchmarking involved comparing the best cost obtained corresponding to hGADE/cur1, hGADE/JADE and hGADE/r1 with those of the benchmark algorithms. This is because the UC being a day-ahead scheduling problem, the decision makers (i.e., system operators) have sufficient time to determine the solution. Thus, the best cost solution is usually preferred [17, 20]. The benchmarking in terms of best cost has been extensively employed on the UC algorithms in the literature [15, 16]. It is noted that the benchmark algorithms employed the same test systems and the results of the benchmark algorithms are directly obtained from the original publications.

Table 3.21 summarizes the comparative results (in terms of best cost) while Table 3.22 summarizes the ranking in terms of best cost and the total rank (denoted by rank sum) of the contender algorithms across different test systems. Table 3.21 shows that on almost all the test systems, the best cost solution obtained by hGADE/cur1, hGADE/JADE and hGADE/r1 is superior to that of the other approaches. Further, it is noted from Table

Table 3.21 Comparsion in terms of best cost ($) results obtained by different algorithms

Algorithm	10 Unit	20 Unit	40 Unit	60 Unit	80 Unit	100 Unit
EP [11]	564,551	1,125,494	2,249,093	3,371,611	4,498,479	5,623,885
MA [12]	566,686	1,128,192	2,249,589	3,370,595	4,494,214	5,616,314
SA [15]	565,828	1,126,251	2,250,063	N.A.	4,498,076	5,617,876
QEA [16]	**563,938**	1,123,607	2,245,557	3,366,676	4,488,470	5,609,550
EPSO [14]	**563,938**	1,123,773	2,244,772	3,364,250	4,487,742	5,608,055
BGSA [20]	**563,938**	1,123,996	2,246,445	3,364,665	4,488,039	5,607,838
PL [4]	**563,937**	1,124,369	2,246,508	3,366,210	4,489,322	5,608,440
LR [8]	**563,977**	**1,123,297**	2,244,237	3,363,491	**4,485,633**	5,605,678
DP [5]	**563,977**	**1,123,390**	2,244,334	3,366,975	4,490,844	5,610,217
GA [10]	**563,938**	1,124,290	2,246,165	3,365,431	4,487,766	5,606,811
DE [17]	**563,938**	1,124,290	2,246,274	3,365,784	4,488,450	5,607,900
hGADE/r1	**563,938**	**1,123,383**	**2,243,724**	**3,363,470**	4,486,180	**5,604,787**
hGADE/cur1	**563,959**	1,123,426	**2,243,522**	**3,362,908**	**4,485,158**	**5,605,075**
hGADE/JADE	**563,959**	1,123,410	**2,243,971**	**3,362,880**	**4,484,711**	**5,605,632**

3.21 that the proposed hGADE variants outperformed both DE [17] and GA [10] in which binary-real coded DE and binary-real coded GA were very recently proposed, respectively, to solve the UC problem.

The observations from Table 3.22 are as follows:

- On the 10 unit test system, hGADE/r1 secured the 1st rank while both hGADE/cur1 and hGADE/JADE secured the 2nd rank.
- On the 20 unit test system, hGADE/r1 secured the 2nd rank while hGADE/cur1 and hGADE/JADE secured the 4th and 5th rank, respectively.
- On the 40, 60, and 100 unit test systems, the hGADE variants secured the top 3 ranks.
- On the 80 unit test system, hGADE/JADE and hGADE/cur1 secured the 1st and 2nd rank, respectively while hGADE/r1 secured the 4th rank.
- The aforementioned observations as well as the total rank in the rank sum column prove that the hGADE variants are the most consistent algorithms across all the test systems. It is noticed from Table 3.21

Table 3.22 Ranking in terms of best cost obtained by different algorithms

Algorithm	10 Unit	20 Unit	40 Unit	60 Unit	80 Unit	100 Unit	Rank Sum
EP [11]	4	11	12	13	14	14	68
MA [12]	6	13	13	12	12	12	68
SA [15]	5	12	14	N.A.	13	13	57
QEA [16]	**1**	6	7	10	9	10	43
EPSO [14]	**1**	7	6	5	5	8	32
BGSA [20]	**1**	8	10	6	7	6	38
PL [4]	**3**	10	11	9	10	9	52
LR [8]	**3**	**1**	4	4	**3**	4	19
DP [5]	**3**	**3**	5	11	11	11	44
GA [10]	**1**	9	8	7	6	5	36
DE [17]	**1**	9	9	8	8	7	42
hGADE/r1	**1**	**2**	**2**	**3**	4	**1**	**13**
hGADE/cur1	**2**	5	**1**	**2**	2	**2**	**14**
hGADE/JADE	**2**	4	**3**	**1**	**1**	**3**	**14**

that the performance of the hGADE variants and LR [8] is comparable on all the test systems. However, Table 3.22 shows that the hGADE variants narrowly outperform LR [8] in the overall ranking (i.e., rank sum). Further, it is noticed from Table 3.22 that the difference in the overall ranking (i.e., rank sum) achieved by the hGADE variants and the other benchmark algorithms is significant.

This substantiates the proposed hybridization strategy between GA and DE and the efficiency of the hGADE algorithm in solving the UC problem.

3.6 Summary

In this Chapter, a novel framework based on hybridization of GA and DE was presented for solving the UC problem. The flexibility of the proposed hGADE framework was exhaustively demonstrated through hybridizing GA with 4 classical DE variants and 2 state-of-the-art self-adaptive DE variants. According to statistical comparison of the hGADE variants among themselves, hGADE/current-to-rand/1, hGADE/JADE and hGADE/rand-/1 were found to be the best hGADE variants. Further, the effectiveness of the proposed framework was highlighted through extensive comparative

Table 3.23 Best generation schedule obtained using hGADE/rand/1 for the ten-unit test system

	Generating Unit									
Hour	1	2	3	4	5	6	7	8	9	10
1	455	245	0	0	0	0	0	0	0	0
2	455	295	0	0	0	0	0	0	0	0
3	455	370	0	0	25	0	0	0	0	0
4	455	455	0	0	40	0	0	0	0	0
5	455	390	0	130	25	0	0	0	0	0
6	455	360	130	130	30	0	0	0	0	0
7	455	410	130	130	25	0	0	0	0	0
8	455	455	130	130	30	0	0	0	0	0
9	455	455	130	130	85	20	25	0	0	0
10	455	455	130	130	162	33	25	10	0	0
11	455	455	130	130	162	73	25	10	10	0
12	455	455	130	130	162	80	25	43	10	10
13	455	455	130	130	162	33	25	10	0	0
14	455	455	130	130	85	20	25	0	0	0
15	455	455	130	130	30	0	0	0	0	0
16	455	310	130	130	25	0	0	0	0	0
17	455	260	130	130	25	0	0	0	0	0
18	455	360	130	130	25	0	0	0	0	0
19	455	455	130	130	30	0	0	0	0	0
20	455	455	130	130	162	33	25	10	0	0
21	455	455	130	130	85	20	25	0	0	0
22	455	455	0	0	145	20	25	0	0	0
23	455	425	0	0	0	20	0	0	0	0
24	455	345	0	0	0	0	0	0	0	0

study with GA. The best hGADE variants i.e., hGADE/current-to-rand/1, hGADE/JADE and hGADE/rand/1 were found to be efficient on a range of test systems in achieving superior best cost solution and average cost when extensively compared with the other published approaches in the literature.

In the next Chapter, the single-objective UC problem is extended to bi-objective UC problem and minimizing emission is considered as an additional objective along with minimizing system operation cost.

Chapter 4

Multi-objective Day-Ahead Thermal Generation Scheduling in Deterministic Environment

4.1 Introduction

This Chapter extends the single-objective unit commitment problem formulation adopted in Chapter 3 to consider minimizing emission as an additional objective along with minimizing system operation cost. The generation scheduling problem considering both system operation cost and emission as the multiple objectives is a nonlinear, mixed-integer, combinatorial, high-dimensional, highly constrained, multi-objective optimization problem. The optimization skeleton developed for the UC problem in Chapter 3 (i.e., problem-specific: chromosome representation, genetic operators and knowledge) is efficiently embedded within the domination and decomposition based multi-objective optimization frameworks. Non-dominated sorting genetic algorithm II (NSGA-II) [81] and multi-objective evolutionary algorithms based on decomposition (MOEA/D-SBX [85] and MOEA/D-DE [86]) are selected as the representative algorithms from the domination and decomposition frameworks, respectively and efficiently customized and applied to the multi-objective economic/emission unit commitment (MOEE-UC) problem.

The rest of the Chapter is organized as follows. Section 4.2 discusses related work on considering emission along with system operation cost in the generation scheduling problem formulation. Section 4.3 presents the proposed work and the motivation. The multi-objective economic-emission UC problem formulation is presented in Section 4.4. The description of the proposed algorithms is presented in Section 4.5, 4.6 and 4.7. The experimental study is presented in Section 4.8 and the Chapter is summarized in Section 4.9.

4.2 Related Work

The generation scheduling problem considering both system operation cost and emission as the multiple objectives is a challenging multi-objective optimization problem and finding trade-off optimal solutions is a difficult task. For this reason, instead of treating cost and emission as competing objectives, many works have expressed the maximum allowable emission as constraints in the formulation of unit commitment and economic dispatch problems [123–125]. The drawback in this approach is that the information about the trade-off solutions cannot be obtained. Thus, many researchers have solved the generation scheduling problem considering both system operation cost and emission as the multiple objectives. These works are reviewed as follows:

4.2.1 Brief Review of Methods Proposed for Multi-objective Economic/Emission Dispatch Problem

Over the last decade, several multi-objective optimization algorithms based on evolutionary computation and swarm intelligence such as NSGA-II [126, 127], SPEA [128], NSGA and NPGA [129], MOPSO [130–132], multi-objective chaotic particle swarm optimization (MOCPSO) [133], multi-objective chaotic ant swarm optimization (MOCASO) [134], multi-objective differential evolution [135], multi-objective interactive honey bee mating optimization (IHBMO) [136], enhanced multi-objective cultural algorithm (EMOCA) [137], etc. have been proposed for solving the multi-objective economic/emission dispatch (EED) problem.

However, the limitation of such works is that the EED problem assumes all the available generating units to be committed and performs only the multi-objective economic/emission dispatch task i.e., the generator on/off determination task is neglected. Nevertheless, such extensive works on the multi-objective EED problem highlights the motivation of the researchers to consider emission as an independent objective along with the economic objective in the generation scheduling problem.

4.2.2 Brief Review of Methods Proposed for Multi-objective Economic/Emission Unit Commitment (MOEE-UC) Problem

Recently, the researchers have started focusing on solving the UC problem as a true MOP considering both economic and emission objectives. A modified NSGA-II based on problem specific genetic operators and priority list (PL) based heuristic initialization is presented in [21] for solving the MOEE-UC problem. In this approach, GA is employed to explore the binary search space while the lambda-iteration method is adopted for assigning the economic/emission power dispatch. In [22], a NSGA-II [81] based algorithm customized with problem specific genetic operators, Priority list (PL) based heuristic initialization and repair operation is presented for the MOEE-UC problem. The NSGA-II based algorithm proposed in [22] is extended in [23] and optimization models are presented in which reliability can be included as an additional constraint or objective along with economic and emission objectives. In both of these works i.e., [22] and [23], GA is employed to solve the tasks of determining the units to be turned on/off as well as the load dispatch. A two-level approach for UC considering bi-objective optimization model of system operation cost and emission and a three-objective optimization model of system operation cost, emission and transmission losses is presented in [24]. The applicability of the approach [24] is presented by integrating the approach within NSGA-II [81], SPEA-2 [82] and a simulated-annealing based multi-objective optimization algorithm (AMOSA) [138]. A memetic EA based on combination of NSGA-II and a problem specific local search algorithm is proposed in [139] to solve the MOEE-UC problem. The memetic algorithm is heuristically initialized using PL based solutions. Further, the on-off schedule is determined using NSGA-II (combined with local search algorithm) while the economic-emission dispatch problem is solved using weighted-sum lambda-iteration method. The main drawback of these MOEE-UC studies is that they lacked thorough validation and benchmarking, especially, with some recent state-of-the-art MOEAs and this leaves a scope for further improvement.

4.3 Proposed Work and the Motivation

Recognizing the importance of the multi-objective economic/emission UC (MOEE-UC) problem and observing that almost all the approaches proposed in the literature for solving this problem are based on NSGA-II [81], motivated us to select a recent state-of-the-art algorithm - multi-objective evolutionary algorithm based on decomposition (MOEA/D) [85] and investigate its performance in solving the particular problem. As discussed in Chapter 2, MOEA/D is a recently proposed evolutionary multi-objective optimization framework by Zhang and Li in 2007 [85]. Since its proposition, several MOEA/D variants have been proposed in the literature [87–97].

Inspired from the performance of MOEA/D and its variants in the literature, in this Chapter, the framework of MOEA/D [86] is chosen. MOEA/D-SBX and MOEA/D-DE are chosen as the representative algorithms from this framework for application to the MOEE-UC problem. Further, for comprehensive comparison, NSGA-II-SBX [81] is also tested on the MOEE-UC problem. The optimization skeleton developed for the single-objective UC problem in Chapter 3 is efficiently integrated within the domination and decomposition based multi-objective optimization frameworks.

In the original study, MOEA/D-SBX as well as MOEA/D-DE were proposed for continuous MOPs. However, as mentioned earlier, the UC problem is a mixed-integer optimization problem consisting of both binary UC variables and continuous power dispatch variables. Therefore, in MOEA/D-SBX as well as NSGA-II-SBX, GA is employed to explore both the binary search space as well as the continuous search space. On the other hand, since the hybridization of GA and DE was found to work very well on the (single-objective) UC problem in Chapter 3, the hybridization methodology is extended in this Chapter within the MOEA/D-DE framework. Thus, in proposed (hybrid) MOEA/D-DE, GA is employed to explore the binary search space while DE explores the continuous search space. To the best of our knowledge, this work presents a first attempt to hybridize two powerful EAs - GA and DE (in the aforementioned manner) within the MOEA/D framework to solve a challenging real-world multi-objective mixed-integer optimization problem.

A swap window real mutation operator which works on the continuous variables is also proposed in this Chapter to enhance the performance of the proposed MOEAs on the MOEE-UC problem.

Further, in the original study of MOEA/D-SBX [85] and MOEA/D-DE [86], the weight vectors corresponding to different scalar optimization subproblems are uniformly distributed. However, in this Chapter, a non-uniform weight vector distribution strategy is proposed to bias the search direction of MOEA/D-DE. Additionally, an ensemble algorithm based on combination of MOEA/D-DE with uniform and non-uniform weight vector distribution strategy is developed to enhance the overall performance of MOEA/D-DE on the MOEE-UC problem.

4.4 Problem Formulation

In this Section, the multi-objective economic/emission UC problem formulation is presented.

4.4.1 Objective Functions

1. System Operation Cost: The first objective function is to minimize the system operation cost (SOC), where SOC includes the fuel cost and the transition cost of all the generating units over the entire scheduling horizon

[139]. The fuel cost ${f_i}^t$ of unit i is expressed as the quadratic function of its power output ${P_i}^t$ during hour t.

$$ {f_i}^t = a_i {P_i^t}^2 + b_i {P_i}^t + c_i \tag{4.1} $$

where a_i, b_i, c_i are the fuel cost coefficients of unit i.

The transition cost is the sum of the start-up costs and the shut-down costs. In this Chapter, the shut-down costs have not been taken into consideration in accordance with the literature [139] while the start-up cost is modeled as follows:

$$ SU_i^t = \begin{cases} HSC_i, & \text{if } MDT_i \leq T_{OFF,i}^t \leq MDT_i + T_{cold,i} \\ CSC_i, & \text{if } T_{OFF,i}^t > MDT_i + T_{cold,i} \end{cases} \tag{4.2} $$

where SU_i^t is the start-up cost of unit i at hour t, HSC_i and CSC_i represents the hot start cost and cold start cost of unit i, respectively, MDT_i represents the minimum down time of unit i, $T_{OFF,i}^t$ is the continuous off time of unit i up to hour t and $T_{cold,i}$ is the cold start cost of unit i.

Subsequently, the first objective function (F_1) is given by minimization of the following cost function [139].

$$ F_1 = \sum_{t=1}^{T_{\max}} \sum_{i=1}^{N} \left(f_i^t . u_i^t + SU_i^t \, (1 - u_i^{t-1}) u_i^t \right) \tag{4.3} $$

where u_i^t represents the unit commitment status of unit i at hour t ($1 = ON$, $0 = OFF$), T_{max} is the number of hours in the scheduling horizon and N is the number of thermal generating units in the system.

2. Emission The second objective function F_2 is the reduction of emission of air-pollutants into the atmosphere [139].

$$F_2 = \sum_{t=1}^{T_{\max}} \sum_{i=1}^{N} \left(E_i^t . u_i^t \right) \tag{4.4}$$

where ${E_i}^t$ (lb) represents the quantity of pollutants produced by unit i at time t and is defined as

$${E_i}^t = a_{1i} {P_i^t}^2 + b_{1i} P_i^t + c_{1i} \tag{4.5}$$

and a_{1i}, b_{1i}, c_{1i} are the emission coefficients of unit i.

4.4.2 Constraints

1. System power balance: the total power generation at hour t must be equal to the load demand L^t for that hour.

$$\sum_{i=1}^{N} (P_i^t . u_i^t) = L^t, \quad t = 1, 2, ...T_{max} \tag{4.6}$$

2. System spinning reserve requirements: for reliable operation, the system must carry certain reserve capacity at every hour (SR^t) in order to meet unforeseen situations such as deviation in actual load demand from forecast load demand or generator outage.

$$\sum_{i=1}^{N} (P_{max,i} . u_i^t) \geq L^t + SR^t, \quad t = 1, 2, ...T_{max} \tag{4.7}$$

where $P_{max,i}$ represents the rated upper limit generation of unit i.

3. Unit minimum up/down time: if a unit i is turned on/off, it must remain on/off for at least its minimum up/down time (MUT_i/MDT_i)

duration.

$$\begin{aligned} T^t_{ON,i} &\geq MUT_i \\ T^t_{OFF,i} &\geq MDT_i \end{aligned} \tag{4.8}$$

where $T^t_{ON,i}$ and $T^t_{OFF,i}$ represent the continuous on and off time of unit i up to hour t, respectively.

4. Unit generation limits: for stable operation, the power output of each generator is restricted within its limits:

$$P_{min,i} \leq P^t_i \leq P_{max,i} \tag{4.9}$$

where $P_{min,i}$ and $P_{max,i}$ represent the rated lower and upper limit generation of unit i, respectively.

4.5 Proposed Algorithm MOEA/D-DE for the MOEE-UC problem

In this Chapter, the MOEAs proposed for the MOEE-UC problem are NSGA-II-SBX, MOEA/D-SBX and MOEA/D-DE. These MOEAs are developed by efficiently incorporating the optimization skeleton developed for the single-objective UC problem in Chapter 3 within the framework of the original versions of these MOEAs proposed in the literature. Thus, to avoid repetition, only MOEA/D-DE is discussed in detail, followed by the detailed pseudo-code of MOEA/D-DE in this Section. Thereafter, MOEA/D-SBX and NSGA-II-SBX are discussed briefly in the next Sections.

The proposed MOEA/D-DE is vividly outlined in the context of the MOEE-UC problem as follows. It is noted that although many components of MOEA/D-DE are similar to that of the hGADE algorithm proposed in

Chapter 3, yet they are again discussed in this section so as to present the MOEA/D-DE as a complete algorithm and enhance the readability.

4.5.1 Chromosome Representation

For every chromosome, a $N \times T_{max}$ binary unit commitment matrix (UCM) represents the thermal generator on/off status and a $N \times T_{max}$ real power matrix (RPM) represents the corresponding power dispatch. The chromosome representation is depicted in Fig. 4.1. It is noted that a chromosome's actual generation schedule is represented by its resultant power matrix (Res.PM) which is obtained by multiplying the corresponding elements of UCM and RPM.

$$
\begin{array}{c|ccccc|}
 & 1 & 2 & \dots & T_{max}\text{-}1 & T_{max} \\
\hline
1 & u_1^1 & u_1^2 & u_i^t & u_1^{T_{max}-1} & u_1^{T_{max}} \\
2 & u_2^1 & u_2^2 & u_2^t & u_2^{T_{max}-1} & u_2^{T_{max}} \\
: & u_i^1 & u_i^2 & u_i^t & u_i^{T_{max}-1} & u_i^{T_{max}} \\
\text{N-1} & u_{N-1}^1 & u_{N-1}^2 & u_{N-1}^t & u_{N-1}^{T_{max}-1} & u_{N-1}^{T_{max}} \\
\text{N} & u_N^1 & u_N^2 & u_N^t & u_N^{T_{max}-1} & u_N^{T_{max}} \\
\hline
\end{array}
\qquad
\begin{array}{c|ccccc|}
 & 1 & 2 & \dots & T_{max}\text{-}1 & T_{max} \\
\hline
1 & P_1^1 & P_1^2 & P_i^t & P_1^{T_{max}-1} & P_1^{T_{max}} \\
2 & P_2^1 & P_2^2 & P_2^t & P_2^{T_{max}-1} & P_2^{T_{max}} \\
: & P_i^1 & P_i^2 & P_i^t & P_i^{T_{max}-1} & P_i^{T_{max}} \\
\text{N-1} & P_{N-1}^1 & P_{N-1}^2 & P_{N-1}^t & P_{N-1}^{T_{max}-1} & P_{N-1}^{T_{max}} \\
\text{N} & P_N^1 & P_N^2 & P_N^t & P_N^{T_{max}-1} & P_N^{T_{max}} \\
\hline
\end{array}
$$

UCM RPM

Fig. 4.1 Structure of chromosome.

4.5.2 Generation of Initial Population

The UCM of the chromosomes in the initial population are randomly generated binary matrices while the RPM of all the chromosomes in the initial population are generated as follows. Suppose the RPM of the kth chromosome of the population at generation G is denoted by $X_{k,G}$ (where $X_{k,G} = [x_{1,k,G}, x_{2,k,G}, ..., x_{D,k,G}]$, D being the number of decision variables). The jth decision variable of the kth chromosome is randomly initialized for the initial population as

$$x_{j,k} = x_j^{min} + rand[0,1].(x_j^{max} - x_j^{min}) \tag{4.10}$$

where x_j^{min} and x_j^{max} are the minimum and maximum bounds of the jth decision variable, respectively and $rand_{k,j}[0,1]$ is a uniformly distributed random number lying between 0 and 1 and is generated independently for each decision variable of the kth chromosome.

4.5.3 Fitness Evaluation

Since UC is a highly constrained optimization problem, a key factor in the performance of the algorithm lies in how the algorithm handles the constraints.

Boundary Constraint Handling

The generator limit constraints given by (4.9) are handled according to the bound handling approach known as set on boundary [116]. According to this approach, if a continuous variable corresponding to power dispatch of a generator exceeds the bounds (during variation operation), then the variable is set on the boundary.

Load Demand Equality Constraint Repair Operator

In the proposed algorithm, the other constraints (i.e., minimum up down time and minimum spinning reserve constraints) except for the load demand equality constraint get adequately handled over the generations by the replacement principle based on feasibility rules (described later). Therefore, a repair operator is applied to repair chromosomes that violate the load demand equality constraint [13]. In the repair procedure, the chromosome is repaired for load demand equality constraint violation at hour t using priority list (PL) of the thermal units based on fuel cost coefficients. If the total power output of the committed thermal units at hour t is less than the load demand on the system at hour t then the power output of the committed thermal units is increased in ascending order of the PL otherwise the power output of the committed thermal units is decreased in descending order of the PL to meet the load demand. It is always ensured that the power output of the thermal units lies within their generation limits given by (4.9). Algorithm 5 in the Appendix A shows the pseudo-code of the repair procedure.

Constraint Violation Evaluation

At first, all the constraints are normalized because different constraints may take different orders of magnitude. An inequality constraint of the form $g(\mathbf{x}) \geq b$ is normalized using the following transformation:

$$\frac{g(\mathbf{x})}{b} - 1 \geq 0 \tag{4.11}$$

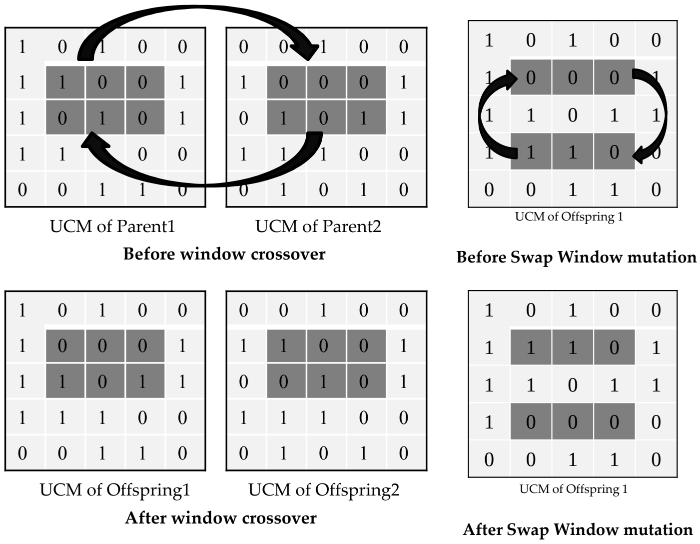
Equality constraints are also normalized similarly [29]. Thereafter, all normalized constraint violations are added to calculate the overall constraint violation of a chromosome. A chromosome is feasible if the overall constraint violation is less than the tolerance limit (10^{-6}).

Objective Function Evaluation

The objective function system operation cost and emission are calculated for each chromosome using its Res.PM (which is obtained by multiplying the corresponding elements of UCM and RPM as mentioned earlier).

4.5.4 Variation Operation: Hybridization of GA with DE

The variation operation is the step where the proposed hybridization between GA and DE occurs at every generation in MOEA/D-DE. In the variation operation in MOEA/D-DE (just like the hGADE algorithm proposed in Chapter 3), the binary UC variables are evolved using GA operators while the continuous power dispatch variables are evolved using DE operators as described below.



(a) Window crossover (b) Swap window operator

Fig. 4.2 A pictorial instance of GA operators acting on UCM of parent chromosomes.

GA Operators Acting on Binary Component (i.e., UCM) of the Chromosomes

Since, in the proposed algorithm, the binary variables are encoded in the form of matrix, problem-specific binary crossover and mutation operators which have been found in the literature to work well on matrix encodings are adopted.

- *Window crossover* - A slightly modified version of the window crossover operator as mentioned in [12] is used as the binary crossover [23]. It works by randomly selecting two parents and then randomly selecting a window size. The entries within the window portion are exchanged between the UCM of two parents to generate the UCM of two offspring. Fig. 4.2 (a) shows an example to illustrate how the window crossover works on a 5×5 UCM for a window size 2×3.

- *Swap window mutation* - It works on the UCM of a chromosome by randomly selecting: a) two units, b) a time window of width w

between 1 and T_{max} and c) a window position. The entries of the two units included in the window are then exchanged. This acts like a sophisticated mutation operator [9]. Fig. 4.2 (b) shows an example to illustrate how the swap window mutation works on a 5×5 UCM for a window size 1×3.

- *Window mutation* - This operator works on the UCM of a chromosome by randomly selecting: a) a unit, b) a time window of width w between 1 and T_{max} and c) a window position. Then it mutates all the bits included in the window, turning all of them to either 1's or 0's with an equal probability [9].

DE Operators Acting on Continuous Component (i.e., RPM) of the Chromosomes

- *Mutation* - Corresponding to RPM of kth chromosome at generation G, $X_{k,G}$ (called target chromosome in DE literature) in the population, DE creates a mutant chromosome $V_{k,G}$ (where $V_{k,G} = [v_{1,k,G}, v_{2,k,G}, ..., v_{D,k,G}]$, D being the number of decision variables) through mutation. In MOEA/D-DE, DE/rand/1 strategy is employed [86], the mutation operation of which takes place as follows:

$$DE/rand/1: \ V_{k,G} = X_{r_1^k,G} + F(X_{r_2^k,G} - X_{r_3^k,G}) \tag{4.12}$$

where r_1^k, r_2^k and r_3^k are mutually exclusive and randomly chosen indices probabilistically from either $[1, ..., T]$ or $[1, ..., NP]$ and are also different from the base index k (where T and NP are the neighborhood size and population size, respectively). The scaling factor F is a control parameter for amplifying the difference of two chromosomes (for example the difference $(X_{r_2^k,G} - X_{r_3^k,G})$ in a vector sense) and lies in the range $[0, 2]$. A smaller value of F promotes exploitation while a larger value of F promotes exploration [60].

- *Crossover* - After generating the mutant chromosome $V_{k,G}$ through mutation, a crossover operation comes into play to further enhance the potential diversity of the population. In crossover, the mutant chromosome $V_{k,G}$ exchanges its components with the target chromosome $X_{k,G}$ with a probability $CR \in [0, 1]$ to form the trial chromosome $U_{k,G}$ (where $U_{k,G} = [u_{1,k,G}, u_{2,k,G}, ..., u_{D,k,G}]$, D being the number of decision variables). In MOEA/D-DE, binomial crossover exists [86] in which each component of the trial chromosome $U_{k,G}$ is inserted from either mutant chromosome or target chromosome according to the following condition:

$$u_{j,k,G} = \begin{cases} v_{j,k,G} & if\ (rand_{k,j}[0,1] \leq CR \text{ or } j = j_{rand}) \\ x_{j,k,G} & otherwise \end{cases} \tag{4.13}$$

  where $rand_{k,j}[0, 1]$ is a uniformly distributed random number and $j_{rand} \in [1, 2, ..., D]$ is a randomly chosen index which ensures that the trial chromosome gets at least one component from the mutant chromosome.

Proposed Mutation Operator Acting on Continuous Component (i.e., RPM) of the Chromosomes

- *Swap window real mutation* - A swap window real mutation operator is proposed and incorporated within the algorithm. This mutation operator, as the name suggests, resembles the swap window mutation operator acting on UCM of the chromosomes as discussed above. The only difference lies in the aspect that swap window real mutation operator acts on RPM of the chromosomes unlike swap window mutation operator which acts on the UCM of the chromosomes. Thus, this operator works on the RPM of a chromosome by randomly selecting: a) two units, b) a time window of width w between 1 and T_{max}, and c) a window position. The entries of the two units included in the window are then exchanged.

4.5.5 Replacement

The original MOEA/D-SBX [85] and MOEA/D-DE [86] are proposed for unconstrained optimization problems. However, as the UC problem is a constrained optimization problem, the replacement component of the original MOEA/D is modified to incorporate constraint handling [140] as discussed below.

At every generation, once corresponding to an index i the variation operation is completed i.e., the child's (say $x_{child's}$) UCM and RPM are created using GA and DE, respectively; the UCM and RPM are combined to evaluate the fitness of the x_{child}. Thereafter, x_{child} is compared with a randomly picked solution in the neighborhood (say y) of index i and the replacement/update of neighborhood takes place according to the following rules based on superiority of feasibility [140].

- If both x_{child} and y are infeasible and $CV(x_{child}) < CV(y)$, then y is replaced by x_{child} (where CV denotes the total constraint violation).
- Else if x_{child} is feasible but y is infeasible, then y is replaced by x_{child}.
- Else if x_{child} is infeasible but y is feasible, then y is not replaced by x_{child}.
- Else if both x_{child} and y are feasible and $g(x_{child}|\lambda_j, z) \leq g(y|\lambda_j, z)$, i.e., if x_{child} is equal to or better than y with regard to Tchebycheff aggregation function, then y is replaced by x_{child}.

4.5.6 Stopping Criterion

The algorithm stops if the maximum number of generations (set as input) is reached.

4.5.7 Steps of the Proposed Algorithm MOEA/D-DE

Input

- NP: the number of subproblems considered in MOEA/D-DE i.e., the population size;
- $\lambda_1, \lambda_2, ..., \lambda_{NP}$: a set of NP weight vectors;
- T: the neighborhood size;
- δ: the probability that parent solutions are selected from the neighborhood;
- n_r: the maximal number of solutions that can be replaced by each child solution.
- z: the initial reference point $(z_1, z_2) = (10^{30}, 10^{30})$. The reference point initially has very large dimensions and is updated during the evolution of population.

At each generation, MOEA/D-DE maintains the following:

- A population of NP solutions $x_1, x_2, \ldots, x_{NP}$, where x_i is the current solution to the i_{th} subproblem.
- $F(x_1), F(x_2), \ldots, F(x_{NP})$, where $F(x_i) = \{F_1(x_i), F_2(x_i)\} \quad \forall i = 1, 2, \ldots, NP$.
- $CV(x_i)$ = total constraint violation of x_i $\forall i = 1, 2, \ldots, NP$.
- $z = (z_1, z_2)$, where z_1 and z_2 are the best values found so far for objective F_1 and F_2, respectively.

The steps executed are as follows.

- **Step 1: Initialization**
  - **Step 1.1** Compute the Euclidean distances between any two weight vectors and then calculate T closest weight vector to each λ_i. For all $i = 1, 2, \ldots, NP$, set $B(i) = \{i_1, i_2, \ldots, i_T\}$, where λ_j, $\forall j \in B(i)$ are T closest vectors to λ_i.
  - **Step 1.2** Randomly generate the initial population.

- **Step 1.3** For all $i = 1, 2, \ldots, NP$, repair x_i for load demand equality constraint violation.
- **Step 1.4** Calculate $CV(x_i)$ and $F(x_i)$ i.e., $\{F_1(x_i), F_2(x_i)\}$.
- **Step 1.5** Update $z = (z_1, z_2)$ according to the condition: $z_j = \min_{1 \le i \le NP} F_j(x_i)$ if x_i is feasible.

- **Step 2: Update**
For $i = 1, 2, \ldots, NP$, **do**
  - **Step 2.1 Selection of Mating/Update Range:** Uniformly generate random number $rand$ from [0,1]. Then,

$$P = \begin{cases} B(i), & \text{if } rand < \delta \\ 1, 2, \ldots, NP, & \text{otherwise} \end{cases}$$

  - **Step 2.2 Reproduction:**
    1. Randomly select three indices r_1, r_2, and r_3 from P which are different from i.
    2. Decode x_k in UCM_k and RPM_k, where $k = i, r_1, r_2, r_3$.
    3. Generate a solution UCM_{child} using GA recombination operators on UCM_k, where $k = i, r_1$.
    4. Generate a solution RPM_{child} using DE recombination operators on RPM_k, where $k = r_1, r_2, r_3$.
    5. Encode UCM_{child} and RPM_{child} in x_{child}.
  - **Step 2.3 Repair:** Repair x_{child} for boundary constraint violation and load demand equality constraint violation.
  - **Step 2.4:** Calculate $CV(x_{child})$ and $F(x_{child})$ i.e., $\{F_1(x_{child}), F_2(x_{child})\}$.
  - **Step 2.5 Update of z:** For $j = 1, 2$ do
    1. If x_{child} is feasible and $z_j > F_j(x_{child})$ then set $z_j = F_j(x_{child})$
  - **Step 2.6 Replacement/Update of Solutions:** Set $c = 0$ and then do
    1. Set $flag = 0$.
    2. If $c = n_r$ or P is empty, $i = i + 1$ and go to **Step 2.1**, else randomly pick an index j from P.
    3. Determine if x_{child} replaces x_j or not according to the replacement rules.
    4. If x_{child} replaces x_j then $flag = 1$ and $c = c + 1$.
    5. If $flag = 1$, remove j from P and go to **Step 2.6.1**.

- **Step 3: Stopping Criteria**
If termination criterion is satisfied, then stop else go to **Step 2**.

Output

- Approximation to Pareto-optimal solutions: $\{x_1, x_2, \ldots, x_{NP}\}$.
- Approximation to Pareto-optimal front: $\{F(x_1), F(x_2), ..., F(x_{NP})\}$.

4.6 Proposed MOEA/D-SBX for the MOEE-UC problem

The proposed algorithm MOEA/D-SBX for the MOEE-UC problem is similar to the algorithm MOEA/D-DE (described above) in each and every aspect other than the variation operation. As the name suggests, in MOEA/D-SBX, the continuous component of the chromosomes are evolved using SBX operator unlike DE operators in MOEA/D-DE. Since, MOEA/D-DE and MOEA/D-SBX have lot of resemblance, in order to avoid repetition, the pseudo-code of MOEA/D-SBX is not presented here.

4.7 Proposed NSGA-II-SBX for the MOEE-UC problem

The proposed algorithm NSGA-II-SBX for the MOEE-UC problem is based on the domination based multi-objective optimization framework of NSGA-II [81] (as discussed in Chapter 2). In NSGA-II-SBX, as the name suggests, GA evolves both the binary as well as the continuous component of the chromosomes. The algorithm is called NSGA-II-SBX instead of NSGA-II to emphasize that the continuous component of the chromosomes are evolved using SBX operator just like in MOEA/D-SBX. The optimization skeleton i.e., problem-specific chromosome representation, load demand equality constraint handling, genetic operators and swap window real mutation operator is the same in NSGA-II-SBX as that in MOEA/D-

DE and MOEA/D-SBX. Thus, NSGA-II-SBX is not discussed in detail and the algorithm is illustrated through a flowchart as shown in Fig. 4.3.

4.8 Experimental Study

In this Section, extensive case studies are undertaken to exhaustively demonstrate the effectiveness of the different algorithmic components and investigate the performance of the proposed algorithms on the MOEE-UC problem. The experimental evaluation is systematically divided into 7 case studies.

1. In the first case study, the effectiveness of the window crossover operator is demonstrated;

2. In the second case study, the efficacy of the binary mutation operators is presented;

3. In the third case study, the effectiveness of the proposed swap window real mutation operator is illustrated;

4. Thereafter, in the fourth case study, the proposed MOEAs i.e., NSGA-II-SBX, MOEA/D-SBX and MOEA/D-DE are exhaustively compared among themselves;

5. Further, in the fifth case study, the proposed MOEAs are benchmarked against the approaches presented in the literature;

6. In the sixth case study, a non-uniform weight vector distribution strategy is proposed for MOEA/D-DE and its effectiveness is investigated;

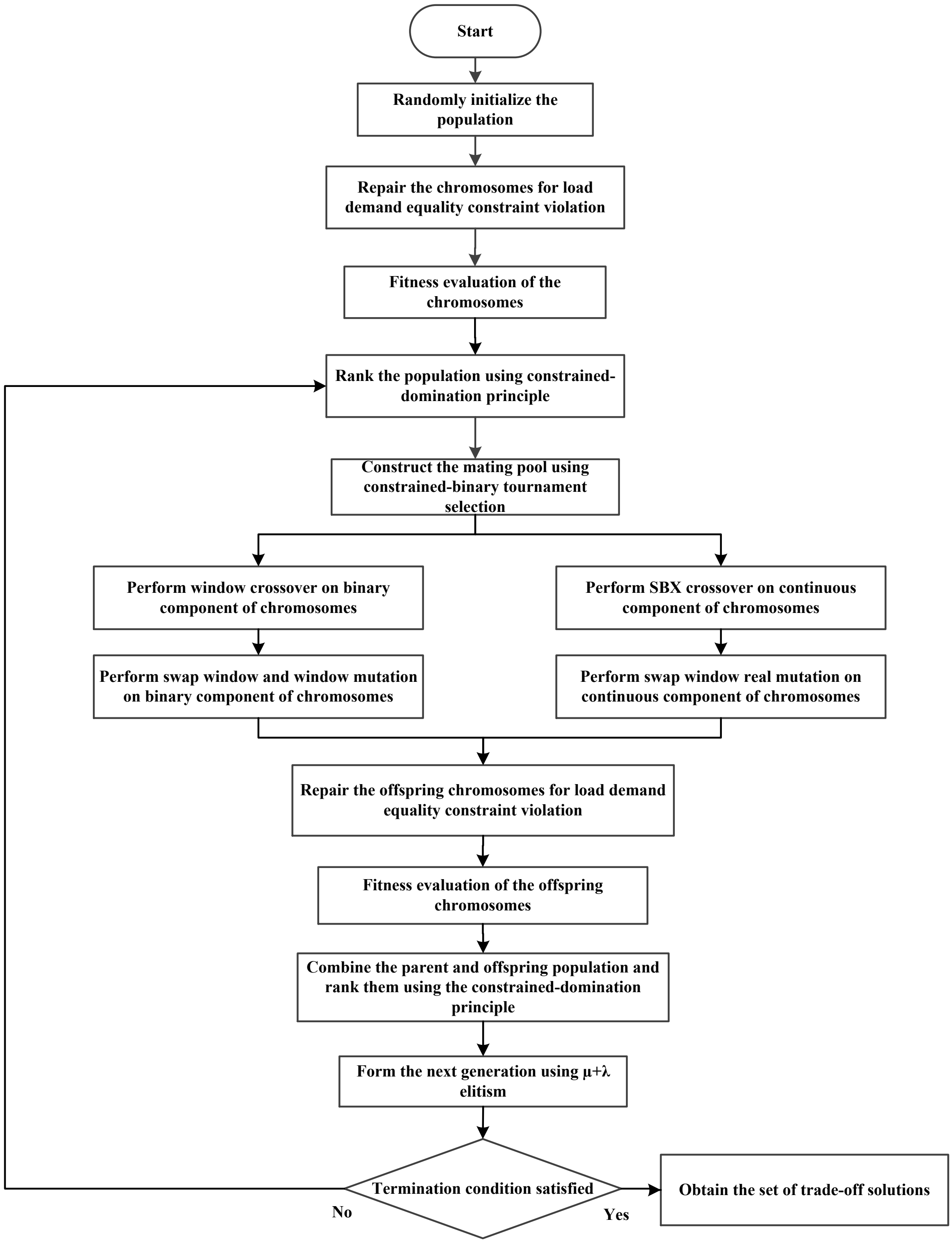


Fig. 4.3 Flowchart of the NSGA-II-SBX for MOEE-UC problem.

7. Finally, in the seventh case study, an enhanced MOEA/D-DE based on ensemble of MOEA/D-DE with uniform and non-uniform weight vector distribution strategy is proposed.

The proposed MOEAs are developed on C++ platform and executed on PC with Intel 3.10 GHz processor. The MOEAs are tested on the MOEE-UC problem for power systems with 10, 60 and 100 units in a 24 hour scheduling horizon [139]. The spinning reserve requirements are assumed to be 10% of the load demand [139]. For each experiment, 20 independent simulation trials are conducted to verify the robustness of the proposed algorithm.

4.8.1 Performance Metric

To investigate the performance of the proposed algorithms, inverted generational distance (IGD) [140] is used as the performance metric. The reasons behind choosing IGD metric are that it provides a measure of both proximity and diversity of the obtained non-dominated solutions in the objective space with respect to the Pareto-optimal front [140]. Further, recently it has been observed that IGD is the most widely used indicator to measure the performance of MOEAs in the evolutionary community.

For the analytical benchmark functions possessing pre-defined Pareto-optimal front, let P^* be the set of uniformly distributed Pareto-optimal solutions in the objective space and P be the obtained approximation set of non-dominated solutions in the objective space from an algorithm. The IGD of the approximation set P with respect to the ideal set P^* is defined as follows:

$$IGD(P, P^*) = \frac{\sum\limits_{\mathbf{v} \in P^*} d(\mathbf{v}, \mathbf{p})}{\mid P^* \mid}, \tag{4.14}$$

where $d(\mathbf{v}, \mathbf{p}) = \min_{\mathbf{v} \in P^*} ||\mathbf{v} - \mathbf{p}||$ with $\mathbf{p} \in P$, $\mid P^* \mid$ being the cardinality of P^*.

However, for real-world problems like MOEE-UC, the exact optimal front being unknown, the Pareto-optimal front is approximated by a reference front which is constructed by selecting the non-dominated solutions from all the simulation runs under the experiment [47]. It is noted that a smaller IGD reflects better proximity and diversity.

4.8.2 Parameter Tuning

Parameter tuning can play a very important role in deciding the performance of EAs [141]. In this Section, the parameters of the proposed MOEAs - NSGA-II-SBX, MOEA/D-SBX and MOEA/D-DE are tuned. The common parameters upon which the performance of the proposed MOEAs depend are - population size, terminating generation number, window crossover probability, binary mutation probability and swap window real mutation probability. Further, some parameters are specific to NSGA-II-SBX and MOEA/D-SBX like SBX distribution index (η_c) and SBX variable crossover probability and some parameters are specific to MOEA/D-DE like scaling factor F and crossover probability CR. Additionally, a parameter common to both MOEA/D-SBX and MOEA/D-DE is the neighborhood size T.

The common parameters like population size and generation number were decided through pilot experiments and are summarized for different test systems in Table 4.1. The rest of the parameters were tuned by employing IGD metric comparison as discussed below. The 60 unit test system was selected as the representative test system on which parametric tuning was conducted.

Parameter tuning with respect to NSGA-II-SBX

Table 4.1 Common parameter settings of MOEAs corresponding to different test systems

Test system	**10-unit system**	**60-unit system**	**100-unit system**
Population size	200	300	400
Generation number	10000	50000	50000

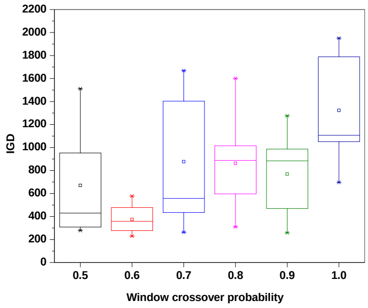
At first, the sensitivity of NSGA-II-SBX to window crossover probability was determined by fixing other parameters as: SBX distribution index at 2, SBX variable crossover probability at 1.0, binary mutation probability at 0.25, swap window real mutation probability at 0.25, and executing the algorithm 10 times each for different values of window crossover probability ranging from 0.5 to 1.0 at interval of 0.1. Thereafter, IGD metric was plotted using box plots as shown in Fig. 4.4a. It is observed from Fig. 4.4a that the performance of NSGA-II-SBX is best with respect to IGD metric for window crossover probability fixed at 0.6.

Thereafter, fixing window crossover probability at 0.6, and other parameters same as mentioned above, the sensitivity of NSGA-II-SBX to SBX distribution index was determined by executing the algorithm 10 times each for η_c fixed at 2, 5, 10 and 20. Fig. 4.4b shows the IGD metric comparison for different values of η_c. It is observed from Fig. 4.4b that the performance of NSGA-II-SBX is best for η_c fixed at 2.

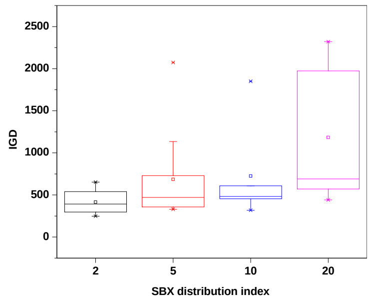
Similarly, one by one the sensitivity of NSGA-II-SBX to the parameters - SBX variable crossover probability, binary mutation probability and swap window real mutation probability was determined by plotting the IGD metric variation as shown in Fig. 4.4c, 4.4d and 4.4e, respectively. Finally, the optimal parameters corresponding to NSGA-II-SBX determined through the above procedure are summarized in Table 4.2.

Parameter tuning with respect to MOEA/D-SBX

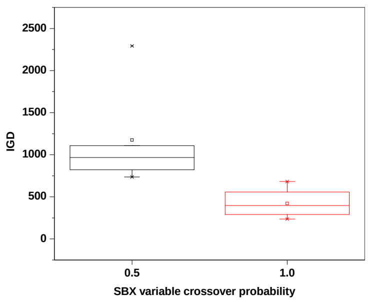
The optimal parameters corresponding to MOEA/D-SBX were also determined in a similar manner. Fig. 4.5a, 4.5b, 4.5c, 4.5d and 4.5e illustrate



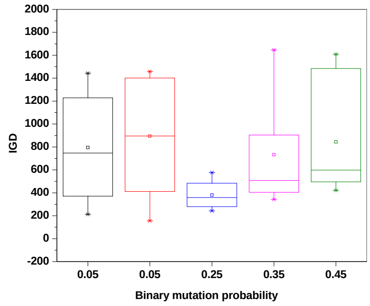
(a) Sensitivity to window crossover probability



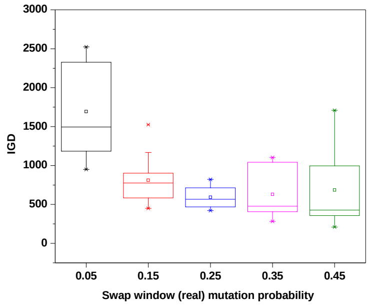
(b) Sensitivity to SBX distribution index



(c) Sensitivity to SBX variable crossover probability



(d) Sensitivity to binary mutation probability



(e) Sensitivity to swap window real mutation probability

Fig. 4.4 Parameter sensitivity results (IGD metric) with respect to NSGA-II-SBX on the 60 unit system.

the IGD metric variation with respect to parameter - window crossover probability, SBX distribution index, neighborhood size, binary mutation probability and swap window real mutation probability, respectively. The optimal parameters corresponding to MOEA/D-SBX are summarized in

Table 4.2 Optimal parameter settings corresponding to NSGA-II-SBX

Window crossover probability	0.6
SBX distribution index (eta_c)	2
SBX variable crossover probability	1.0
Binary mutation probability	0.25
Swap window real mutation probability	0.25

Table 4.3.

Table 4.3 Optimal parameter settings corresponding to MOEA/D-SBX

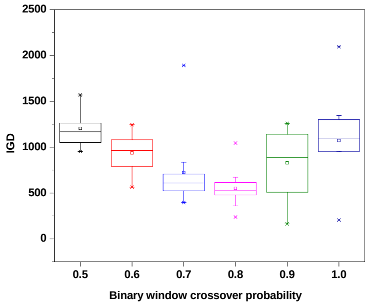
Window crossover probability	0.8
SBX distribution index (eta_c)	5
SBX variable crossover probability	1.0
Neighborhood size (T)	20%
Binary mutation probability	0.25
Swap window real mutation probability	0.35

Parameter tuning with respect to MOEA/D-DE

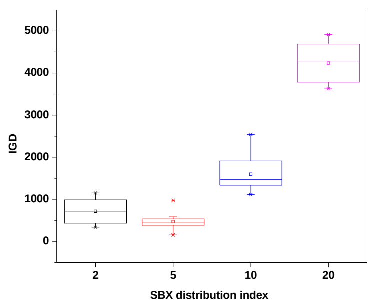
The optimal parameters corresponding to MOEA/D-DE were also determined in similar manner. Fig. 4.6a, 4.6b, 4.6c, 4.6d and 4.6e illustrate the IGD metric variation with respect to parameter - window crossover probability, scaling factor, neighborhood size, binary mutation probability and swap window real mutation probability, respectively. It is noted that the parameter CR is fixed at 1.0 as in the original study on MOEA/D-DE [86]. Further, unlike the original study on MOEA/D-DE [86], polynomial mutation was not adopted. This is because the polynomial mutation operator was not found to enhance the performance of the proposed MOEAs. The optimal parameters corresponding to MOEA/D-DE are summarized in Table 4.4.

Table 4.4 Optimal parameter settings corresponding to MOEA/D-DE

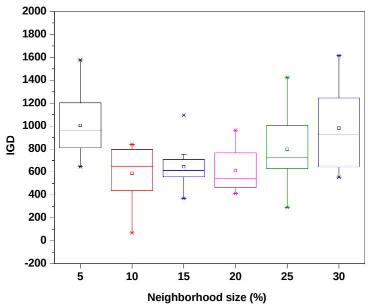
Window crossover probability	0.5
Scaling factor (F)	0.7
Binomial crossover probability (CR)	1.0
Neighborhood size (T)	15%
Binary mutation probability	0.35
Swap window real mutation probability	0.35



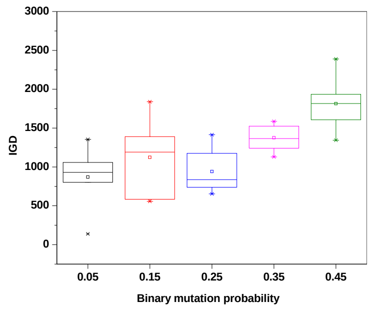
(a) Sensitivity to window crossover probability



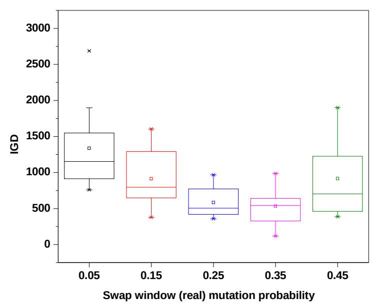
(b) Sensitivity to SBX distribution index



(c) Sensitivity to neighborhood size (T)

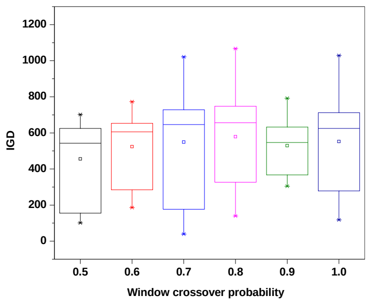


(d) Sensitivity to binary mutation probability

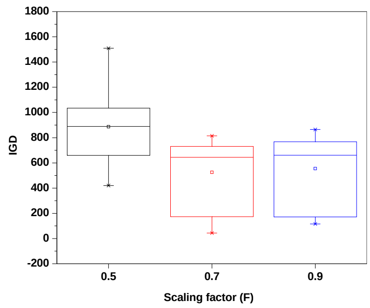


(e) Sensitivity to swap window real mutation probability

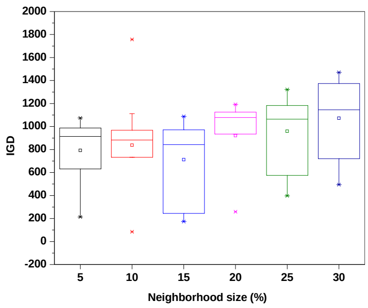
Fig. 4.5 Parameter sensitivity results (IGD metric) with respect to MOEA/D-SBX on the 60 unit system.



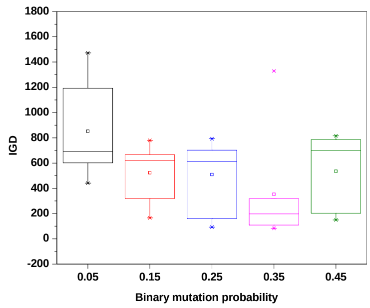
(a) Sensitivity to window crossover probability



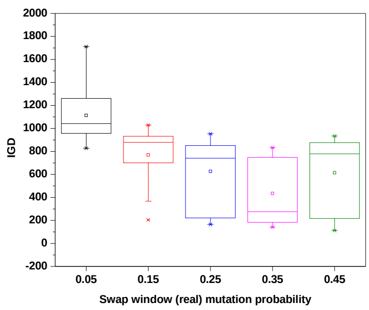
(b) Sensitivity to scaling factor F



(c) Sensitivity to neighborhood size (T)



(d) Sensitivity to binary mutation probability



(e) Sensitivity to swap window real mutation probability

Fig. 4.6 Parameter sensitivity results (IGD metric) with respect to MOEA/D-DE on the 60 unit system.

4.8.3 Case Study 1- Study on Efficacy of Window Crossover Operator

In this case study, the efficacy of window crossover operator was investigated by comparing with three other crossover operators - row, column and line. These crossover operators are defined as follows:

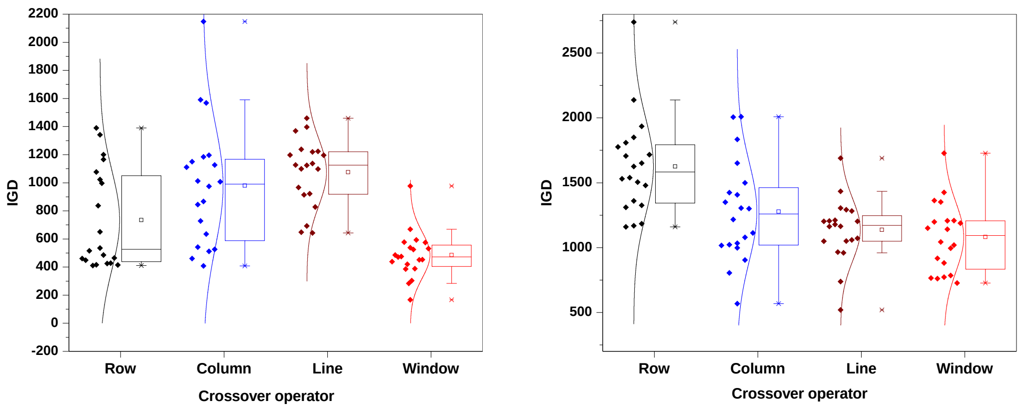
- *Row crossover* - It works by randomly selecting two parents and then randomly selecting two rows. The entries within the two rows (i.e., across all the columns) are exchanged between the UCM of two parents to generate the UCM of two offspring.

- *Column crossover* - It works by randomly selecting two parents and then randomly selecting two columns. The entries within the two columns (i.e., across all the rows) are exchanged between the UCM of two parents to generate the UCM of two offspring.

- *Line crossover* - It works by randomly selecting two parents and then randomly selecting a column. The entries on the right hand side of the column (i.e., across all the rows) are exchanged between the UCM of two parents to generate the UCM of two offspring.

It is noted that the row and column crossover operator are similar to window crossover operator but are different in the sense that unlike window crossover operator, row crossover operator spans across all the columns while column crossover operator spans across all the rows.

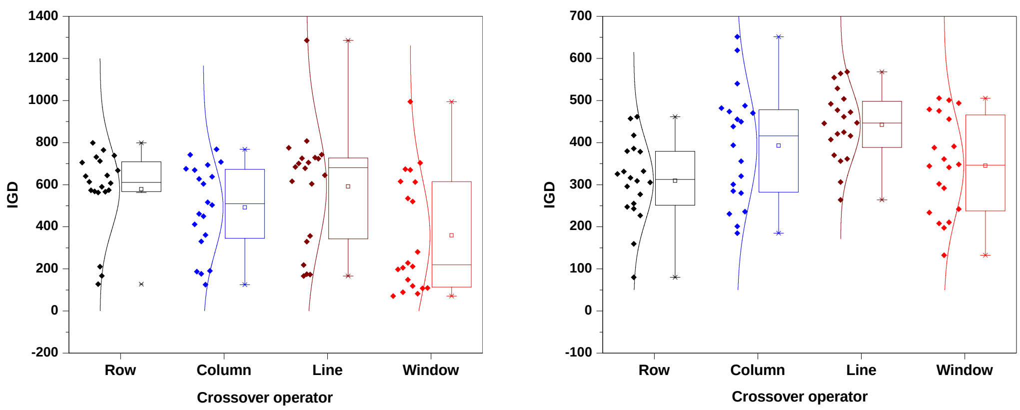
Fig. 4.7a and 4.7b illustrate the comparison of IGD metric results for different genetic crossover operators with respect to NSGA-II-SBX on 60 and 100 unit system, respectively while Fig. 4.8a and 4.8b illustrate the same with respect to MOEA/D-DE on 60 and 100 unit system, respectively.



(a) 60 unit system (b) 100 unit system

Fig. 4.7 IGD metric results for different genetic crossover operators with respect to NSGA-II-SBX.

It is observed from the figures that considering the IGD metric results with respect to both NSGA-II-SBX and MOEA/D-DE, window crossover is the most consistent crossover operator and in most of the cases window crossover operator significantly outperforms the other crossover operators. Thus, this case study justifies the reason behind incorporating window crossover operator in the proposed MOEAs.



(a) 60 unit system (b) 100 unit system

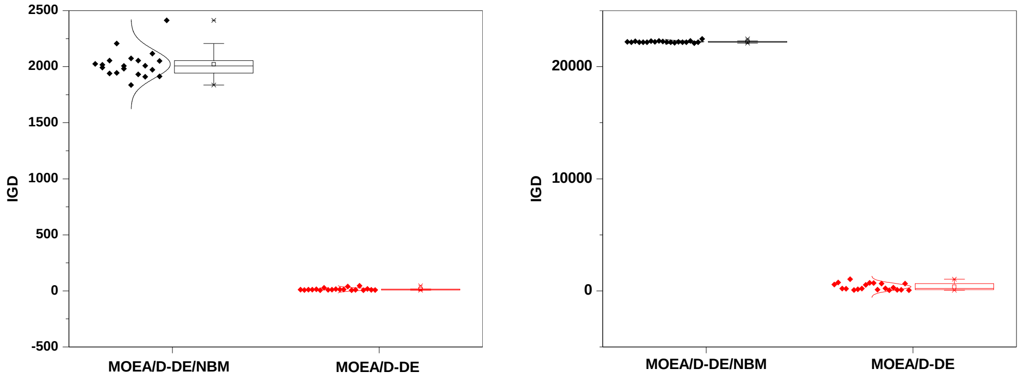
Fig. 4.8 IGD metric results for different genetic crossover operators with respect to MOEA/D-DE.

4.8.4 Case Study 2- Study on Efficacy of Binary Mutation Operators

In this case study, the efficacy of binary mutation operators (i.e., swap window mutation and window mutation) was investigated by executing NSGA-II-SBX and MOEA/D-DE with and without binary mutation operators. It is noted that instead of analyzing the effects of two binary mutation operators separately, their combined effect was investigated. Fig. 4.9a and 4.9b illustrate the comparison of IGD metric results for MOEA/D-DE (i.e., with binary mutation) and MOEA/D-DE/NBM (i.e., MOEA/D-DE without binary mutation) on 10 and 60 unit system, respectively. It is observed from the figures that with binary mutation, MOEA/D-DE performed remarkably better than without binary mutation. Further, in presence of 100 unit system, MOEA/D-DE/NBM could not find any feasible solution while NSGA-II-SBX/NBM (i.e., NSGA-II/SBX without binary mutation) could not find any feasible solution in each of the test systems. Thus, this case study validates the efficacy of the binary mutation operators incorporated in the proposed MOEAs.

4.8.5 Case Study 3- Study on Efficacy of Swap Window Real Mutation Operator

In this case study, the efficacy of the proposed swap window real mutation operator was analyzed by executing NSGA-II-SBX and MOEA/D-DE with and without the mutation operator. Fig. 4.10a and 4.10b illustrate the comparison of IGD metric results for NSGA-II-SBX (i.e., with swap window real mutation) and NSGA-II-SBX/NSWRM (i.e., without swap window real mutation) on 60 and 100 unit system, respectively while Fig.

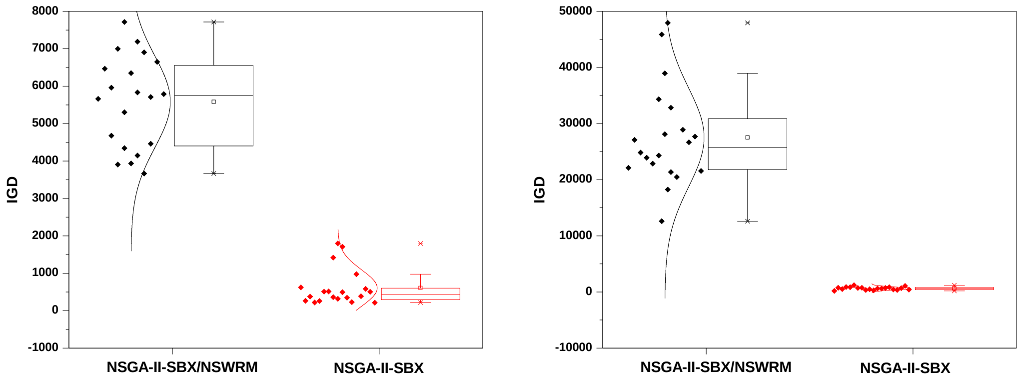


(a) 10 unit system

(b) 60 unit system

Fig. 4.9 IGD metric results for with and without binary mutation with respect to MOEA/D-DE.

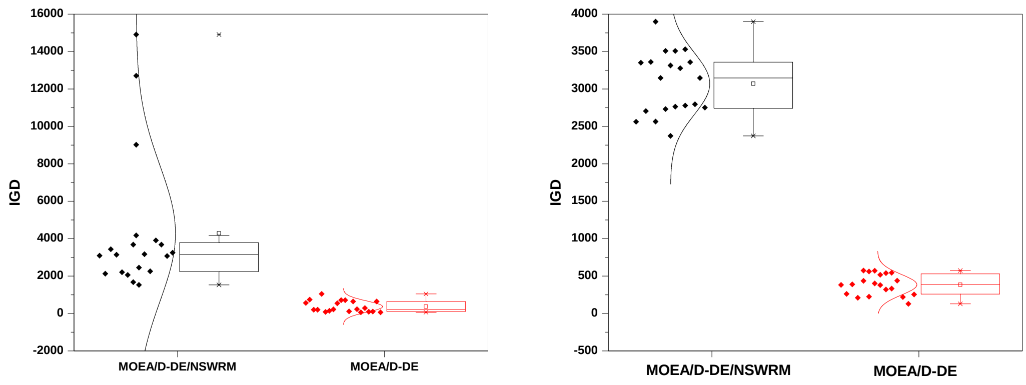
4.11a and 4.11b illustrate the same for MOEA/D-DE (i.e., with swap window real mutation) and MOEA/D-DE/NSWRM (i.e., MOEA/D-DE without swap window real mutation) on 60 and 100 unit system, respectively. It is observed from the figures that with respect to both NSGA-II-SBX and MOEA/D-DE, inclusion of the proposed swap window real mutation operator had a remarkable effect on the quality of results.



(a) 60 unit system

(b) 100 unit system

Fig. 4.10 IGD metric results for with and without swap window real mutation with respect to NSGA-II-SBX.



(a) 60 unit system

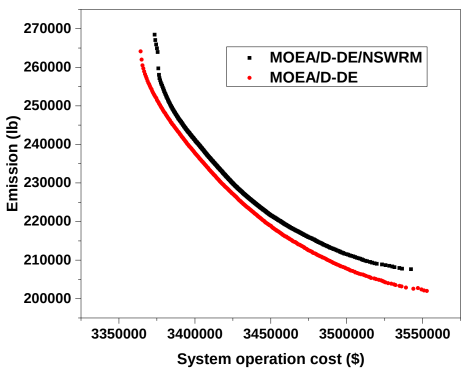
(b) 100 unit system

Fig. 4.11 IGD metric results for with and without swap window real mutation with respect to MOEA/D-DE.

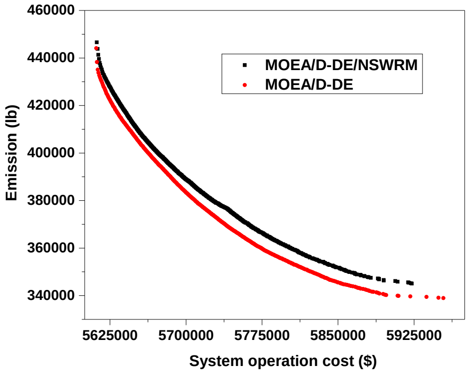
To further analyze how MOEA/D-DE and MOEA/D-DE/NSWRM compare in the objective space, the distribution of the final non-dominated solutions with the lowest IGD values found by MOEA/D-DE and MOEA/D-DE/NSWRM on 60 and 100 unit system are plotted in Fig. 4.12a and 4.12b, respectively. It is evident from the figures that with the incorporation of the proposed swap window real mutation operator, MOEA/D-DE is able to obtain significantly better convergence. Thus, this case study validates the effectiveness of the proposed swap window real mutation operator incorporated in the different MOEAs.

4.8.6 Case Study 4- Comparative Study of the Proposed MOEAs

In this case study, the performance of the proposed MOEAs i.e., NSGA-II-SBX, MOEA/D-SBX and MOEA/D-DE is compared on three different test systems comprising of 10, 60 and 100 units. Fig. 4.13a, 4.13b and 4.13c illustrate the IGD metric comparison for NSGA-II-SBX, MOEA/D-SBX



(a) 60 unit system



(b) 100 unit system

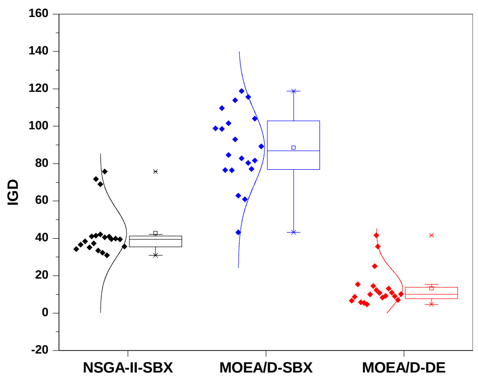
Fig. 4.12 The distribution of the final non-dominated solutions found (with the lowest IGD values) by MOEA/D-DE and MOEA/D-DE/NSWRM.

and MOEA/D-DE on 10, 60 and 100 unit system, respectively. Following are the observations from these figures:

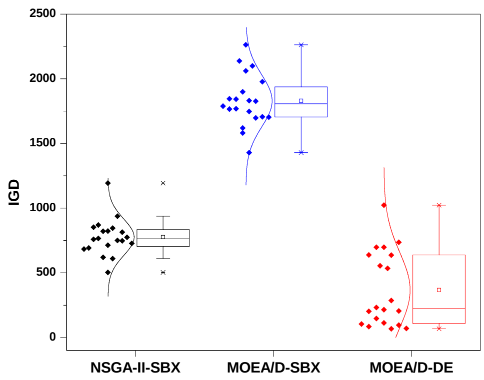
- The best, mean and median IGD of MOEA/D-DE is significantly lower than that of NSGA-II-SBX and MOEA/D-SBX on all the test systems and thus MOEA/D-DE is significantly superior to NSGA-II-SBX and MOEA/D-SBX in terms of IGD metric.

- Since, the performance of MOEA/D-SBX is worse than that of NSGA-II-SBX, it indicates that incorporation of decomposition based multi-objective optimization framework alone may not be adequate. Moreover, as MOEA/D-DE remarkably outperforms MOEA/D-SBX, it shows that the proposed hybridization strategy between GA and DE as in MOEA/D-DE plays a significant role in improving the performance of MOEA/D on the MOEE-UC problem.

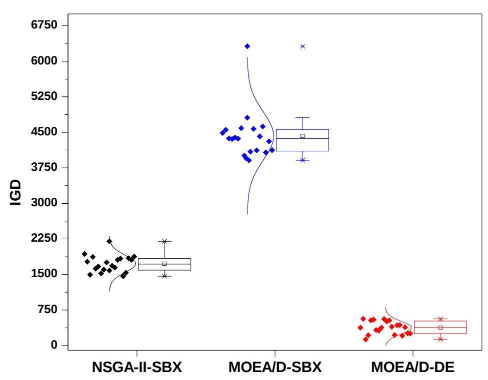
Fig. 4.14a, 4.14b and 4.14c show the distribution of the final non-dominated solutions found by NSGA-II-SBX, MOEA/D-SBX and MOEA/D-DE with the lowest IGD values on 10, 60 and 100 unit system, respectively.



(a) 10 unit system



(b) 60 unit system

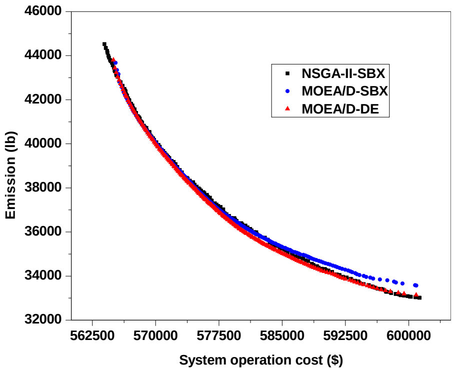


(c) 100 unit system

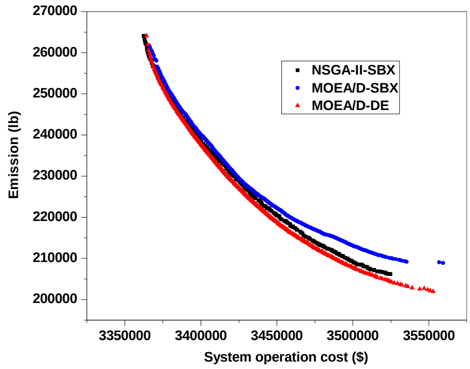
Fig. 4.13 IGD metric results with respect to performance of proposed MOEAs.

The superior performance of MOEA/D-DE is clearly visible on 60 and 100 unit system in Fig. 4.14b and 4.14c, respectively as MOEA/D-DE outperforms MOEA/D-SBX with respect to convergence and outperforms NSGA-II-SBX both with respect to convergence and diversity.

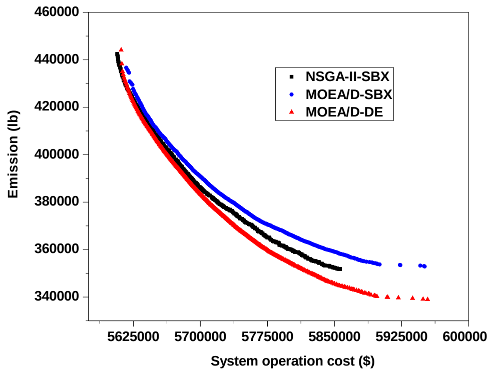
However, to further investigate if MOEA/D-DE is statistically different from NSGA-II-SBX and MOEA/D-SBX, Wilcoxon signed rank test [122] is applied with respect to the IGD metric values. Table 4.5 summarize the results obtained through Wilcoxon signed rank test. The table summarizes the statistics obtained from - a) pairwise comparison between MOEA/D-



(a) 10 unit system



(b) 60 unit system



(c) 100 unit system

Fig. 4.14 The distribution of the final non-dominated solutions (with the lowest IGD values) found by NSGA-II-SBX, MOEA/D-SBX and MOEA/D-DE.

DE and NSGA-II-SBX, and b) pairwise comparison between MOEA/D-DE and MOEA/D-SBX on the three test systems.

It is observed from the *p*-values in the Table 4.5 that MOEA/D-DE is statistically superior to both NSGA-II-SBX and MOEA/D-SBX at 0.01 level of significance. Thus, Wilcoxon test results confirm that MOEA/D-DE is significantly superior to both NSGA-II-SBX and MOEA/D-SBX. The comparative analysis of the proposed MOEAs presented in this case study shows that MOEA/D-DE is the best performing algorithm among the proposed MOEAs on this problem. This highlights the efficacy of the proposed hybridization strategy between GA and DE for solving the MOEE-

Table 4.5 Results obtained through Wilcoxon signed rank test between MOEA/D-DE and other two MOEAs with respect to quality of IGD metric on different test systems

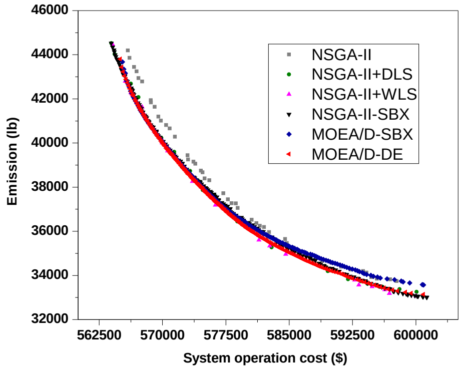
MOEA/D-DE	**10-unit system**			**60-unit system**			**100-unit system**		
v.s.	**R+**	**R-**	***p-value***	**R+**	**R-**	***p-value***	**R+**	**R-**	***p-value***
NSGA-II-SBX	209	1	**3.81E-6**	196	14	**2.09E-4**	210	0	**1.91E-6**
MOEA/D-SBX	210	0	**1.91E-6**	210	0	**1.91E-6**	210	0	**1.91E-6**

UC problem.

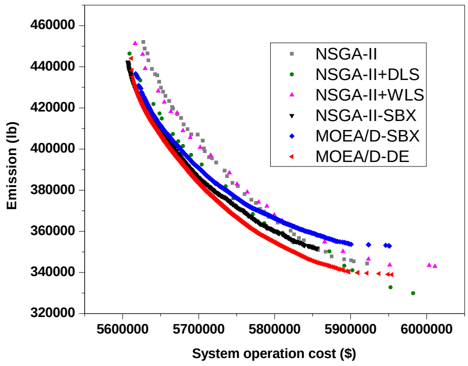
4.8.7 Case Study 5- Benchmarking of the Proposed MOEAs

In this case study, the ability of the proposed MOEAs is investigated by benchmarking it against the algorithms presented in the literature. The competitor algorithms considered are NSGA-II [139] and two variants of multi-objective memetic algorithms suggested in [139], termed NSGA-II+WLS and NSGA-II+DLS, which were designed by incorporating two local search techniques called wide local search (WLS) and deep local search (DLS) within NSGA-II. The results of NSGA-II, NSGA-II+WLS and NSGA-II+DLS are directly obtained from the authors' of the original publication [139]. Fig. 4.15a and 4.15b show the distribution of the final non-dominated solutions found by NSGA-II-SBX, MOEA/D-SBX and MOEA/D-DE (with the lowest IGD values) and the benchmark algorithms NSGA-II, NSGA-II+WLS and NSGA-II+DLS (reported in [139]) on 10 and 100 unit system, respectively. It is noted that only 10 and 100 unit systems are considered here for benchmarking because in the original publication [139], the results were presented for the benchmark algorithms on only 10 and 100 unit system.

However, as MOEA/D-DE was found to perform the best among the proposed MOEAs in the previous case study, the focus of rest of the Chap-



(a) 10 unit system



(b) 100 unit system

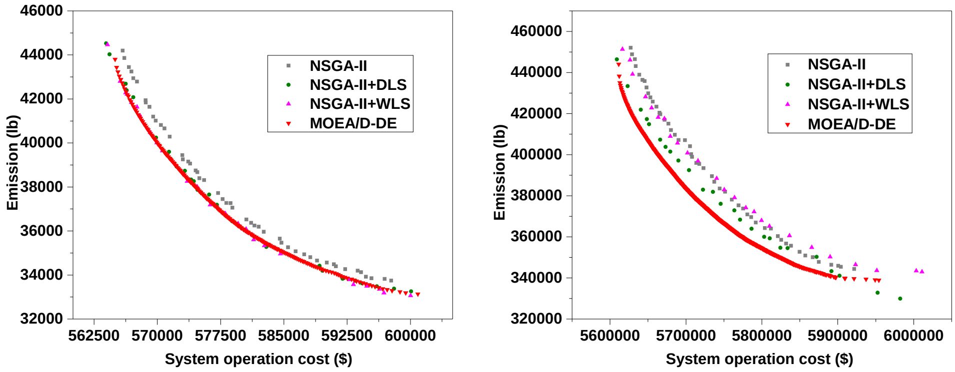
Fig. 4.15 The distribution of the final non-dominated solutions found (with the lowest IGD values) by NSGA-II-SBX, MOEA/D-SBX and MOEA/D-DE and the benchmark algorithms NGSA-II, NSGA-II+WLS and NSGA-II+DLS.

ter is on comparing MOEA/D-DE with the benchmark algorithms i.e., NSGA-II, NSGA-II+WLS and NSGA-II+DLS. For clarity, the distribution of the final non-dominated solutions found by MOEA/D-DE (with the lowest IGD values) and the benchmark algorithms NSGA-II, NSGA-II+WLS and NSGA-II+DLS (reported in [139]) on 10 and 100 unit system is presented in Fig. 4.16a and 4.16b, respectively.

Before analyzing the performance of the proposed algorithm MOEA/D-DE with respect to the benchmark algorithms, the goals of an ideal MOEA (as discussed in Chapter 2) are re-visited. The goals are to obtain - 1) good convergence, 2) uniform distribution and 3) good spread in the objective space [29].

Following are the observations from Fig. 4.16a and 4.16b with respect to convergence, distribution and spread characteristics of the contender MOEAs.

- **Convergence** - Fig. 4.16a shows that on the 10 unit test system, MOEA/D-DE outperforms NSGA-II and performs comparable to NSGA-II+WLS and NSGA-II+DLS in terms of convergence. On



(a) 10 unit system

(b) 100 unit system

Fig. 4.16 The distribution of the final non-dominated solutions found (with the lowest IGD values) by MOEA/D-DE and the benchmark algorithms NGSA-II, NSGA-II+WLS and NSGA-II+DLS.

the 100 unit system, Fig. 4.16b shows that MOEA/D-DE significantly outperforms all the three benchmark algorithms i.e., NSGA-II, NSGA-II+WLS and NSGA-II+DLS in terms of convergence.

- **Distribution** - It is evident from Fig. 4.16a and 4.16b that MOEA/D-DE provides a uniformly distributed set of trade-off solutions while the solutions provided by NSGA-II, NSGA-II+WLS and NSGA-II+DLS are distinctly scattered in the objective space and not well distributed.

- **Spread** - On both 10 and 100 unit systems, MOEA/D-DE fails to capture the solution with the minimum system operation cost (i.e., objective F_1) while on the 100 unit system, MOEA/D-DE also fails to capture the solutions with as well as close to minimum emission (i.e., objective F_2).

The above characteristics of the contender MOEAs shows that the proposed algorithm MOEA/D-DE outperforms the benchmark algorithms with respect to convergence and distribution aspects. The only drawback of MOEA/D-DE in contrast to the benchmark algorithms is the inability to obtain a better spread (or better convergence at the extremes) in the objective space. Although, the solutions with minimum emission are practically least desirable to system operators as these solutions (because of the conflicting nature of the objectives) correspond to maximum system operation cost, yet the proposed algorithm MOEA/D-DE cannot be considered superior to the benchmark algorithms until it obtains the solution with minimum system operation cost as well. Notwithstanding, the benchmarking analysis presented in this case study clearly unveils the potential of the proposed algorithm MOEA/D-DE to solve the MOEE-UC problem, with some scope of enhancement to obtain a better spread as well in the objective space.

4.8.8 Case Study 6- Proposed Non-uniform Weight Vector Distribution Strategy and its Effectiveness

It is clear from the previous case study that the proposed MOEA/D-DE requires some modification to catch the tails of the Pareto-optimal front. Further, it is intuitively identified that MOEA/D-DE requires a guided exploration towards the extremities in order to obtain a better spread (or better convergence at the extremes) in the objective space. Different strategies can be incorporated to enhance the performance of a MOEA. One of the interesting strategy can be incorporation of local search within the framework of MOEA. There are several studies in the literature in which local search operators have been proposed to improve the efficacy of MOEAs [47–49], etc. Such MOEAs are called multi-objective memetic algorithms. However, there are many issues that may affect the performance of memetic algorithms such as the choice of scalarization function in the local search operator and frequency of local search operator [142].

Another interesting strategy can be making some modification in the framework of MOEA/D. Several modifications of the original MOEA/D have been presented in the literature recently [87–97] (as discussed in Chapter 2). However, we decided to present a new modification which can exactly target the existing limitation observed in the performance of MOEA/D-DE on the MOEE-UC problem.

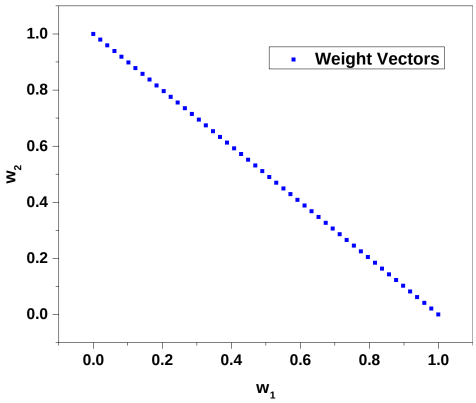
In the original study of MOEA/D [85] and MOEA/D-DE [86], a uniform weight-vector distribution (UWD) strategy is suggested and the weight vectors employed are uniformly distributed in the closed interval $[0, 1]$ to provide uniform weight to all the search directions. However, as the proposed algorithm with the UWD strategy in spite of obtaining well converged and uniformly distributed non-dominated solutions fails to outperform the benchmark algorithms in terms of spread; a non-uniform weight-vector distribution (NUWD) strategy is proposed in this case study.

The target of the proposed NUWD strategy is to help MOEA/D-DE achieve more spread and/or convergence in both directions while maintaining the performance of MOEA/D-DE in terms of convergence and distribution throughout the trade-off front. Thus, in the proposed NUWD strategy, search directions are more concentrated towards the extremes with slight compromise in the middle i.e., more sub-problems are allocated towards the extremes and relatively fewer sub-problems in the middle. A function selected to generate non-uniformly distributed weight vectors is a scaled and shifted cosine function, and is defined as:

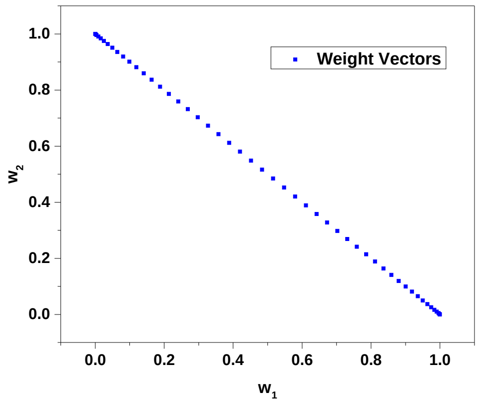
$$\lambda_i^{k'} = g(\lambda_i^k) = 0.5(1 - cos\pi\lambda_i^k) \quad i = 1, 2, \ldots, NP;\ k = 1, 2. \tag{4.15}$$

where $\lambda_i^{k'}$ replaces λ_i^k as input in the algorithm MOEA/D-DE.

Fig. 4.17a and 4.17b show the uniform weight-vector distribution employed in the original MOEA/D-DE [86] and the non-uniformly distributed weight vectors generated using the proposed strategy employing the cosine function mentioned above.



(a)

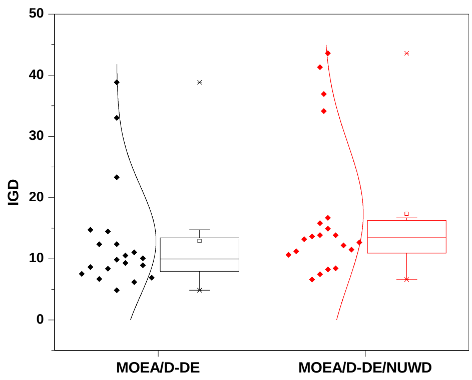


(b)

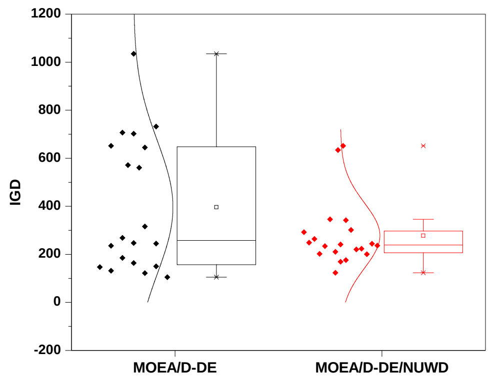
Fig. 4.17 (a) Uniform weight vector distribution in the original MOEA/D-DE, (b) Proposed non-uniform weight vector distribution.

Next, the NUWD strategy is incorporated within MOEA/D-DE and the performance of the resulting algorithm, MOEA/D-DE/NUWD, is investigated by comparing it against MOEA/D-DE (i.e., with the UWD strategy). Fig. 4.18a, 4.18b and 4.18c show the comparison of MOEA/D-DE and MOEA/D-DE/NUWD on the basis of IGD metric for 10, 60 and 100 unit system, respectively. It is observed from the figures that although on 10 unit system, MOEA/D-DE and MOEA/D-DE/NUWD perform comparably but on 60 and 100 unit system, MOEA/D-DE/NUWD performs much better than MOEA/D-DE. The NUWD strategy helps in improving the average and worst IGD value in the case of 60 unit system while improving the best, mean, median and worst IGD value in the case of 100 unit system.

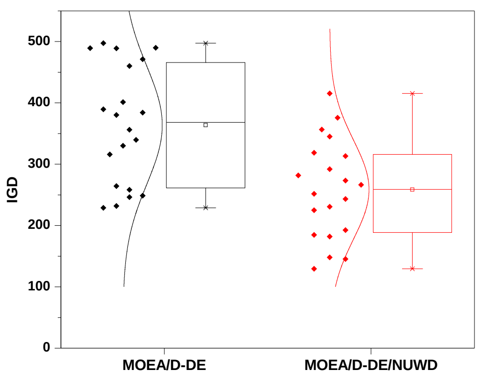
To further analyze how MOEA/D-DE and MOEA/D-DE/NUWD com-



(a) 10 unit system



(b) 60 unit system



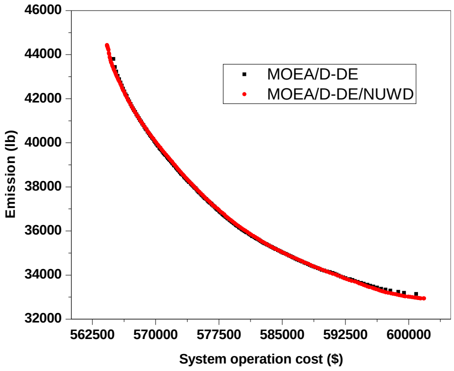
(c) 100 unit system

Fig. 4.18 IGD metric results for MOEA/D-DE with uniform and (proposed) non-uniform weight vector distribution.

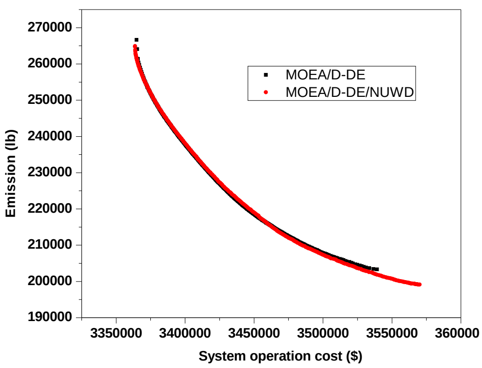
pare in the objective space, the distribution of the final non-dominated solutions with the lowest IGD values found by MOEA/D-DE and MOEA/D-DE/NUWD on 10, 60 and 100 unit system are plotted in Fig. 4.19a, 4.19b and 4.19c, respectively.

It is visually evident from the figures that the proposed NUWD strategy improves the performance of MOEA/D-DE/NUWD (in comparison to MOEA/D-DE) on different test systems in the following ways:

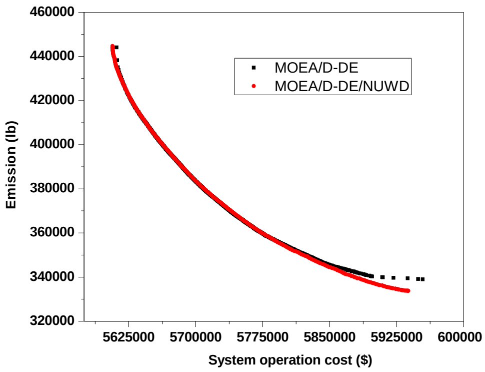
- Attaining better spread in the extreme region of minimum system



(a) 10 unit system



(b) 60 unit system



(c) 100 unit system

Fig. 4.19 The distribution of the final non-dominated solutions found (with the lowest IGD values) by MOEA/D-DE with uniform and proposed non-uniform weight vector distribution.

operation cost in the case of 10 unit system (see Fig. 4.19a)

- Attaining better spread in the extreme region of minimum emission in the case of 60 unit system (see Fig. 4.19b)
- Attaining better convergence in the extreme regions of minimum system operation cost and minimum emission in the case of 100 unit system (see Fig. 4.19c).

However, there is a limitation of MOEA/D-DE/NUWD which is difficult to be observed if the IGD metric box plots and the scatter plots are viewed separately. It lies in the fact that the improvement of MOEA/D-DE/NUWD in terms of spread and convergence at the extreme regions come at the cost of slight compromise in convergence as compared to MOEA/D-DE in the remaining part of the trade-off front. This can be analyzed by viewing and comparing the corresponding IGD metric box plots and scatter plots of MOEA/D-DE and MOEA/D-DE/NUWD on different test systems. For example, on the 60 unit system, the scatter plot in Fig. 4.19b shows that MOEA/D-DE NUWD obtains much better spread in the extreme region of minimum emission. However, the comparative IGD metric plots in Fig. 4.18b shows that the best IGD value attained by MOEA/D-DE/NUWD is almost comparable to that of MOEA/D-DE. Similar observation is made in the case of 10 unit system as well. Inspite of better spread as in case of MOEA/D-DE/NUWD, if the best IGD value remains the same, it means that there is a slight compromise in the remaining part of the trade-off front.

Thus, the comparative analysis of MOEA/D-DE and MOEA/D-DE with NUWD strategy leads to the following inferences:

- The proposed non-uniform weight vector distribution strategy considerably improves the overall performance of MOEA/D-DE as evident from IGD metric comparison.

- Thus, non-uniform weight vector distribution strategy is a promising method to bias the search direction of MOEA/D towards a particular region of the trade-off front (in this case towards the extremes).

- MOEA/D-DE and MOEA/D-DE/NUWD seem to complement each other in the sense that MOEA/D-DE/NUWD provides better spread and convergence towards the extremes of the trade-off front while

MOEA/D-DE provides better convergence in the remaining part of the trade-off front.

- Hence, there is scope for further improvement in the performance of the proposed MOEA/D-DE algorithm on the MOEE-UC problem.

4.8.9 Case Study 7- Proposed Enhanced MOEA/D-DE and comparison with the benchmark algorithms

Having observed that there is scope for further improvement in the performance of MOEA/D-DE and that the nature of MOEA/D-DE and MOEA/D-DE/NUWD is complementary to each other, it was intuitive that combining the two MOEAs may lead to an enhanced performance. Thus, in this case study, an ensemble optimizer is proposed to overcome the limitations of MOEA/D-DE and MOEA/D-DE/NUWD. The proposed ensemble optimizer is based on combination of MOEA/D-DE and MOEA/D-DE/NUWD.

It is well known that with the advent of CPU with multiple cores, parallel computing has become quite affordable and convenient. Thus, in the proposed ensemble optimizer, MOEA/D-DE and MOEA/D-DE/NUWD are executed independently but simultaneously on two different cores (processors) without any communication (i.e., no migration) until both the algorithms reach the termination condition (i.e., maximum generation). Upon the completion of single run of both the algorithms, the final population of both are combined and non-dominated sorting [81] is implemented. The non-dominated solutions of the combined population are then sorted in the descending order with respect to the crowding distance [81] and the top NP (i.e., popsize) solutions are retained as the final trade-off solutions.

The proposed ensemble optimizer is based on a parallel multi-start model which is heterogeneous and independent i.e., non co-operative [143]. The parallel multi-start model is heterogeneous because it consists of two different algorithms (i.e., MOEA/D-DE and MOEA/D-DE/NUWD) and independent because there is no exchange of information during the execution of the component algorithms [143]. It is noted that the basic idea of the proposed ensemble optimizer is to efficiently combine the complementary strengths of MOEA/D-DE and MOEA/D-DE/NUWD by running them parallelly and combining in the end to amplify the overall performance [143].

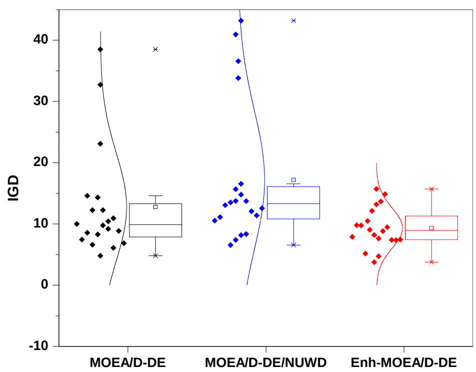
The algorithm based on the aforementioned ensemble strategy is termed Enhanced-MOEA/D-DE (or Enh-MOEA/D-DE) and is tested by running on PC with Intel dual core 3.10 GHz processor. At first, Enh-MOEA/D-DE is compared with its individual component algorithms i.e., MOEA/D-DE and MOEA/D-DE/NUWD in terms of IGD metric on 10, 60 and 100 unit test systems (see Fig. 4.20a, 4.20b and 4.20c). The observations from these comparative IGD box plots are as follows:

- On 10 unit system, Enh-MOEA/D-DE performs better than both MOEA/D-DE and MOEA/D-DE/NUWD.

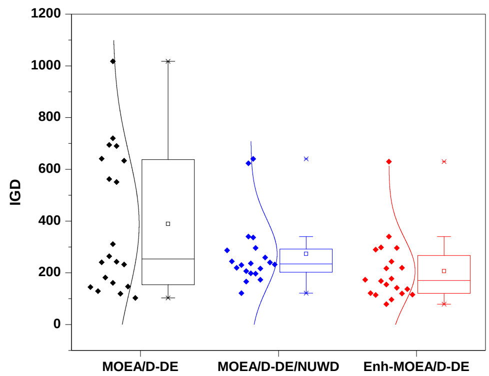
- On 60 unit system, Enh-MOEA/D-DE performs much better than MOEA/D-DE and comparable to MOEA/D-DE/NUWD.

- On 100 unit system, Enh-MOEA/D-DE significantly outpeforms both MOEA/D-DE and MOEA/D-DE/NUWD.

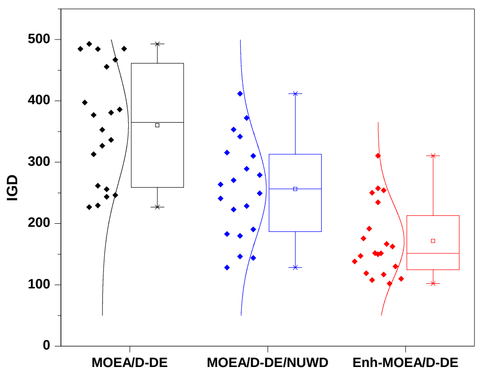
To further investigate if Enh-MOEA/D-DE is statistically different from MOEA/D-DE and MOEA/D-DE/NUWD, Wilcoxon signed rank test [122]



(a) 10 unit system



(b) 60 unit system



(c) 100 unit system

Fig. 4.20 IGD metric results for MOEA/D-DE, MOEA/D-DE/NUWD and Enh-MOEA/D-DE.

is applied with respect to the IGD metric values. Table 4.6 summarize the results obtained through Wilcoxon signed rank test. The table summarizes the statistics obtained from - a) pairwise comparison between Enh-MOEA/D-DE and MOEA/D-DE, and b) pairwise comparison between Enh-MOEA/D-DE and MOEA/D-DE/NUWD on the three test systems. Following are the observations from the p-values in the Table 4.6:

- Enh-MOEA/D-DE shows significant improvement over MOEA/D-DE on 10, 60 and 60 unit system at the level of significance 0.1, 0.01

Table 4.6 Results obtained through Wilcoxon signed rank test between Enh-MOEA/D-DE and MOEA/D-DE and between Enh-MOEA/D-DE and MOEA/D-DE/NUWD with respect to quality of IGD metric on different test systems

Enh-MOEA/D-DE	**10-unit system**			**60-unit system**			**100-unit system**		
v.s.	**R+**	**R-**	***p-value***	**R+**	**R-**	***p-value***	**R+**	**R-**	***p-value***
MOEA/D-DE	153	57	**0.075**	195	15	**2.61E-4**	210	0	**1.91E-6**
MOEA/D-DE/NUWD	189	1	**7.63E-6**	138	15	**0.002**	210	0	**1.91E-6**

and 0.01, respectively.

- Enh-MOEA/D-DE shows significant improvement over MOEA/D-DE/NUWD on all the three test systems at the level of significance 0.01.

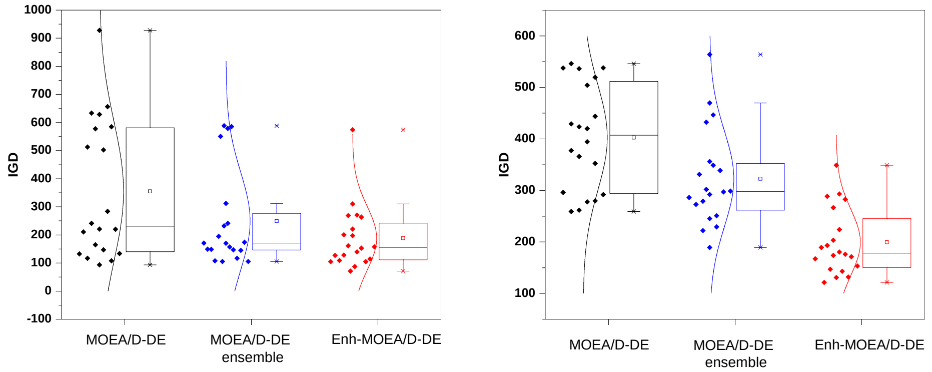
The IGD metric and the statistical comparison indicate that Enhanced-MOEA/D-DE significantly outperforms its component algorithms i.e., MOEA/D-DE and MOEA/D-DE/NUWD. This shows that integration of MOEA/D-DE and MOEA/D-DE/NUWD efficiently combines their strengths and remarkably enhances the performance of MOEA/D-DE. However, there is a limitation in the comparison of Enh-MOEA/D-DE with MOEA/D-DE and MOEA/D-DE/NUWD. The limitation is that since Enh-MOEA/D-DE is an ensemble of MOEA/D-DE/NUWD and MOEA/D-DE, the computational resources (i.e., function evaluations) consumed by Enh-MOEA/D-DE is twice that of MOEA/D-DE and MOEA/D-DE/NUWD. Thus, it was expected that Enh-MOEA/D-DE will show improvement over its constituent algorithms. However, the experiment was still conducted to see how much improvement is actually obtained.

To further investigate the strength of Enh-MOEA/D-DE, another experiment is conducted. In this experiment, an ensemble optimizer based on combination of MOEA/D-DE with itself is executed i.e., MOEA/D-DE is run on two processors and upon the completion of single run of both the algorithms, the final population of both are combined and non-dominated sorting [81] is implemented. The non-dominated solutions of the combined population are then sorted according to descending order with respect to crowding distance [81] and the top NP (i.e., popsize) solutions are retained as the final trade-off solutions. This algorithm is termed MOEA/D-DE ensemble and it resembles Enh-MOEA/D-DE except that the latter is an ensemble of MOEA/D-DE and MOEA/D-DE/NUWD while the former is an ensemble of MOEA/D-DE and MOEA/D-DE.

In this experiment, MOEA/D-DE, MOEA/D-DE ensemble and Enh-MOEA/D-DE are executed on 60 and 100 unit test systems. Fig. 4.21a and 4.21b show the IGD metric comparison of the algorithms on 60 and 100 unit test systems, respectively. It is observed from the figures that although on 60 unit system, the performance of Enh-MOEA/D-DE and MOEA/D-DE ensemble is comparable but on 100 unit system, Enh-MOEA/D-DE significantly outperforms MOEA/D-DE ensemble algorithm as well.

To further analyze if Enh-MOEA/D-DE is statistically superior to MOEA/D-DE and MOEA/D-DE ensemble, Wilcoxon signed rank test [122] is applied with respect to the IGD metric values. Table 4.7 summarize the results obtained through Wilcoxon signed rank test. The table summarizes the statistics obtained from pairwise comparison between Enh-MOEA/D-DE and MOEA/D-DE ensemble on the two test systems. The p-values in the Table 4.7 indicates that Enh-MOEA/D-DE shows significant improvement over MOEA/D-DE ensemble on the 100 unit system at the level of significance 0.01.

Next, the efficacy of Enh-MOEA/D-DE is verified by comparing it against the benchmark algorithms. Fig. 4.22a and 4.22b show the distribution of the final non-dominated solutions found by Enh-MOEA/D-DE (with the lowest IGD values) and the benchmark algorithms NSGA-II,



(a) 60 unit system

(b) 100 unit system

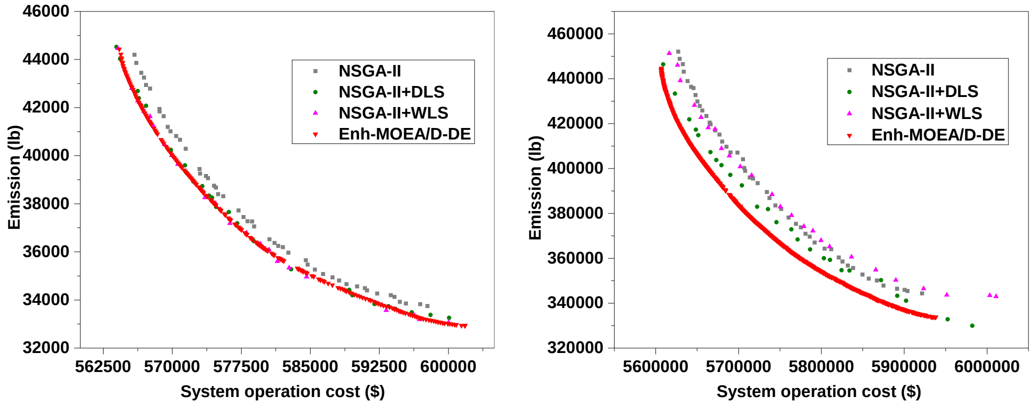
Fig. 4.21 IGD metric results for MOEA/D-DE, MOEA/D-DE ensemble and MOEA/D-DE + MOEA/D-DE/NUWD ensemble i.e., Enh-MOEA/D-DE.

Table 4.7 Results obtained through Wilcoxon signed rank test between Enh-MOEA/D-DE and MOEA/D-DE ensemble with respect to quality of IGD metric on different test systems

Enh-MOEA/D-DE	**60-unit system**			**100-unit system**		
v.s.	**R+**	**R-**	***p-value***	**R+**	**R-**	***p-value***
MOEA/D-DE ensemble	147	63	0.12	207	3	**9.53E-6**

NSGA-II+WLS and NSGA-II+DLS on the 10 and 100 unit system, respectively.

Fig. 4.22a shows that on the 10 unit test system, with respect to convergence, Enh-MOEA/D-DE outperforms NSGA-II while with respect to distribution, Enh-MOEA/D-DE is significantly superior to all the three benchmark algorithms. Further, on the 100 unit system, Fig. 4.22b shows that Enh-MOEA/D-DE remarkably outperforms all the three benchmark algorithms in terms of both convergence and uniform distribution. The improvement of Enh-MOEA/D-DE in comparison to MOEA/D-DE (visually) reflects in the aspect that Enh-MOEA/D-DE is able to capture the solution with minimum system operation cost as well (which MOEA/D-DE failed to do). Although, in terms of spread, NSGA-II+DLS provides few



(a) 10 unit system

(b) 100 unit system

Fig. 4.22 The distribution of the final non-dominated solutions found by Enh-MOEA/D-DE and the benchmark algorithms NGSA-II, NSGA-II+WLS and NSGA-II+DLS.

solutions that Enh-MOEA/D-DE does not. However, the solutions with minimum emission are practically least desirable to system operators as these solutions because of the conflicting nature of the objectives correspond to maximum system operation cost (as mentioned earlier as well).

Next, Enh-MOEA/D-DE is compared in terms of the minimum system operation cost (i.e., single-objective comparison) with the algorithms proposed in the literature for solving the MOEE-UC problem, namely, NSGA-II [139], NSGA-II+WLS [139], and NSGA-II+DLS [139]; the hGADE variants proposed for the UC problem in Chapter 3, namely, hGADE/r1, hGADE/cur1, and hGADE/JADE; and traditional optimization method proposed in the literature for the UC problem, namely, LR [8]. The comparison of the contender algorithms in terms of minimum system operation cost is summarized in Table 4.8. It is observed from Table 4.8 that on the 10 and 100 unit test systems, the proposed Enh-MOEA/D-DE significantly outperforms NSGA-II [139] and NSGA-II+WLS [139] and performs comparably to NSGA-II+DLS [139]. In comparison to the algorithms for solving the single-objective UC problem (i.e., LR, hGADE/r1, hGADE/cur1, and hGADE/JADE), Table 4.8 shows that the proposed Enh-MOEA/D-DE achieves near-optimal solution on all the test systems.

Table 4.8 Single-objective comparison of Enh-MOEA/D-DE with the benchmark algorithms

Algorithm	Best Cost ($)		
	10 unit	60 unit	100 unit
LR [8]	563,977	363,491	5,605,678
hGADE/r1	563,938	363,470	5,604,787
hGADE/cur1	563,959	362,908	5,605,075
hGADE/JADE	563,959	362,880	5,605,632
NSGA-II [139]	565,898	N.A.	5,625,616
NSGA-II+WLS [139]	564,114	N.A.	5,618,657
NSGA-II+DLS [139]	563,938	N.A.	5,605,918
Enh-MOEA/D-DE	563,959	362,971	5,605,425

This case study demonstrates that the proposed ensemble optimizer Enh-MOEA/D-DE shows significant improvement over MOEA/D-DE and significantly outperforms the benchmark algorithms in solving the MOEE-UC problem. Further, in terms of single-objective comparison, the proposed Enh-MOEA/D-DE achieves near-optimal solution on all the test systems. Thus, this case study validates the efficacy of the proposed Enh-MOEA/D-DE in solving the MOEE-UC problem.

4.9 Summary

In this Chapter, NSGA-II-SBX, MOEA/D-SBX and MOEA/D-DE were efficiently customized and applied to solve the UC problem considering system operation cost and emission as the multiple conflicting objectives. The three proposed MOEAs were exhaustively compared among themselves on a range of test systems and MOEA/D-DE was found to significantly outperform both NSGA-II-SBX and MOEA/D-SBX in terms of IGD metric comparison.

The benchmarking of MOEA/D-DE with the results presented in the literature on the MOEE-UC problem revealed that MOEA/D-DE outperforms the benchmark algorithms in terms of convergence and distribution throughout the trade-off front except at the extremes. Thus, a non-uniform weight vector distribution strategy (NUWD) was proposed to improve the performance of MOEA/D-DE towards the extremes. The comparison of MOEA/D-DE and MOEA/D-DE with NUWD strategy i.e., MOEA/D-DE/NUWD revealed that although the latter has superior performance at the extremes, it comes at the cost of slight compromise in convergence as compared to the former in the remaining part of the trade-off front.

Since, MOEA/D-DE and MOEA/D-DE/NUWD were found to complement each other, an ensemble optimizer, termed Enh-MOEA/D-DE, based on combination of MOEA/D-DE and MOEA/D-DE/NUWD was proposed. The proposed Enh-MOEA/D-DE was found to efficiently integrate the strengths of MOEA/D-DE and MOEA/D-DE/NUWD and outperform the individual component algorithms. Also, Enh-MOEA/D-DE was observed to be significantly superior to the benchmark algorithms in terms of convergence as well as uniform distribution.

In this Chapter, the UC problem was solved in a deterministic environment considering economic and emission objectives. In the next Chapter, reliability is added as an additional objective and the UC problem is solved as a three-objective optimization problem considering the uncertainties occurring due to load forecast error and generator outage.

Chapter 5

Multi-objective Day-Ahead Thermal Generation Scheduling in Uncertain Environment

5.1 Introduction

The bi-objective unit commitment problem formulation presented in Chapter 4 for system operation cost and emission as the conflicting objectives considers the environment to be deterministic. However, the generation scheduling is subject to uncertainty due to deviations from load forecasts and outage of components such as generator, etc. Thus, the system operators would prefer to obtain trade-off optimal solutions by incorporating various uncertainties and considering reliability as an additional objective along with system operation cost and emission for better decision making. Thus, this Chapter extends the bi-objective unit commitment problem formulation presented in Chapter 4 to consider maximizing reliability as an additional objective along with minimizing system operation cost and emission. In this Chapter, the uncertainties taken into account are the uncertainties occurring due to thermal generator outage and load forecast error. These uncertainties are captured using expected energy not served (EENS) reliability index while EENS cost is used to reflect the reliability objective. The UC problem considering system operation cost, emission and reliability as the multiple objectives is a a nonlinear, mixed-integer, combinatorial, high-dimensional, highly constrained, three-objective optimization problem. The MOEAs developed for the bi-objective UC problem in deterministic environment in Chapter 4 i.e., NSGA-II-SBX, MOEA/D-SBX and MOEA/D-DE are efficiently extended in this Chapter to solve the three-objective UC problem in uncertain environment.

The rest of the Chapter is organized as follows. Section 5.2 discusses the related work on managing uncertainty in the unit commitment problem. Section 5.3 presents the proposed work and the motivation. The three-objective UC problem formulation is presented in Section 5.4. The procedure for evaluating the reliability objective is presented in Section 5.5. The proposed algorithms are discussed in Section 5.6. The experimental study is presented in Section 5.7 and the Chapter is summarized in Section 5.8.

5.2 Related Work

To address uncertainty in the unit commitment problem, units with total capacity exceeding the forecasted load need to scheduled. In the literature, there are two popular approaches for taking uncertainty into account in the unit commitment problem: reserve requirements and stochastic programming [25]. In both the approaches, reserve is committed either implicitly or explicitly [25]. These approaches and the algorithms based on these approaches proposed in the literature are discussed as follows.

Managing uncertainty through reserve requirements

Spinning reserve allows system operator (SO) to respond to unforeseen imbalances between load and generation caused by generation outages and load forecasting errors. Traditionally, the SR requirement has been based on deterministic criteria that the minimum amount of SR should be either equal to the largest capacity of the online generator, or greater than or equal to a fixed percentage (10%) of the load demand [1]. However, such deterministic criteria do not properly take into account the generation outages or the errors in load forecast i.e., the stochastic nature of the power system. For example, if the SR is set equal to the largest capacity of the online generator than as per this criterion no load shedding will take place if the outage of a single unit occurs. However, the SR could be over-committed if the largest online generator is highly reliable and/or if the customers do not attach high value to continuous supply of energy. Moreover, this criterion does not ensure that the entire load demand would be met in the case two or more generators undergo outage simultaneously. Thus, although deterministic criterion of setting spinning reserve are simple, the solutions obtained can be sub-optimal and unreliable.

Probabilistic reserve assessment methods, on the other hand can provide a more realistic evaluation of the SR requirement in the UC problem. Anstine *et al.* [144] were the first to propose probabilistic reserve assessment criterion to account for the uncertainty related to outage of generating units. They proposed a technique according to which the SR requirement is adjusted at each scheduling period such that a uniform level of risk probability is maintained throughout the scheduling horizon. The disadvantage of this approach is that the risk is an abstract quantity and it does not clearly reflect how much SR should be scheduled. Moreover, maintaining a uniform risk level throughout the scheduling horizon may provide sub-optimal solution as it may require commitment of expensive generating units when such expensive reserve may not be economically justified.

Gooi *et al.* [145] were the first to optimize SR within the UC problem. They proposed a LR based UC method according to which the generation can be scheduled to meet a pre-defined acceptable value of risk index. However, as the risk is an abstract quantity and lacks intuitive interpretation, they used external cost/benefit analysis to suggest that the risk index should be the value at which the sum of operating cost and expected cost of energy not served is minimum.

Chattopadhyay and Baldick [146] proposed a probabilistic reserve assessment method according to which the SR at each scheduling period should be such that the loss of load probability (LOLP) is below a pre-defined LOLP limit. Although, this method is easy to implement, the drawback of this method is that selecting an appropriate LOLP limit is a difficult task and setting LOLP limit arbitrarily may lead to sub-optimal solution.

Bouffard and Galiana [147] proposed a probabilistic reserve assessment method in which the UC problem formulation considers two reliability metrics, expected energy not served (EENS) and loss of load probability (LOLP). They suggested that the SR requirement should be such that both EENS and LOLP at each scheduling period are less than a pre-defined limit. The disadvantage of this approach is the same as the approach proposed in [146] because selecting an appropriate EENS limit and LOLP limit are difficult tasks and setting the limits arbitrarily may lead to sub-optimal solution.

Simopoulos *et al.* [148] included the uncertainties due to unit outage and load forecast error in the UC problem by implementing reliability constraints on LOLP and EENS reliability indices and presented a SA based algorithm to solve the reliability constrained UC problem. Wang and Singh [149] included the uncertainty due to unit outage in the UC problem by revising the spinning reserve constraint and presented a hybrid of binary PSO and real PSO to solve the UC problem. However, this approach neglected the uncertainty due to load forecast error.

A two-level evolutionary approach considering system operation cost for forecasted demand as the first objective and the risk of not fulfilling possible demand variations as the second objective is presented by Georgopoulou and Giannakoglou [150]. However, this approach considered only the uncertainty due to load forecast error and neglected the uncertainty due to unit outages.

Ortega-Vazquez and Kirschen [151] suggested a method in which at first the SR requirement for each scheduling period is determined in an auxiliary computation prior to solving the UC problem. They argued that the SR requirement for each scheduling period should be such that the sum of operating costs and EENS cost is minimum for that period. Thus, they employed external cost/benefit analysis to determine the optimum level of reserve at each period and thereafter, the reserve constrained unit commitment problem was solved.

Chandrasekaran and Simon [152] proposed a fuzzy assisted hybrid of binary and real-coded cuckoo search algorithm (CSA) and solved the UC problem considering system operation cost, emission and reliability. Expected energy not served is used to reflect the reliability objective. However, the drawback of this approach is that only the best compromise solution corresponding to the three-objective optimization problem is presented and not the entire trade-off solutions.

Trivedi *et al.* [23] presented a NSGA-II based generation scheduling algorithm and different optimization models to include uncertainties due to unit outage and load forecast error such as - a) bi-objective optimization model considering system operation cost and EENS reliability index as the multiple objectives with constraint on LOLP reliability index, b) bi-objective optimization model considering system operation cost and emission as the multiple objectives with constraints on LOLP and EENS reliability indices and c) three-objective optimization model considering system operation cost, emission and EENS reliability index as the multiple objectives with constraints on LOLP and EENS reliability indices. The drawback of this approach is (the same as discussed above) that selecting an appropriate LOLP and EENS limit are difficult tasks and setting the limits arbitrarily may lead to sub-optimal solutions.

Managing uncertainty through stochastic optimization

One of the stochastic optimization methods which has been used in the literature to incorporate the uncertainties and solve the UC problem is stochastic programming [153]. Stochastic programming is a framework for modeling optimization problems that involve uncertainty. This approach uses a multi-stage decision framework which recognizes the ability to adapt some decisions to the conditions in real time which generally differ from those forecasted. The simplest such framework has two stages that mimic the decision process and is generally the one which is adopted for solving the UC problem.

Solving UC problem through stochastic programming requires scenario analysis to model the uncertainties [154]. Scenarios are different possible situations or realization that could happen because of the existing uncertainties. Ideally a huge number of scenarios are required to completely incorporate the uncertainties. However, solving the stochastic UC problem with the huge set of scenarios is computationally too expensive. Thus, scenario reduction techniques are employed to limit the number of scenarios. Then the goal of the stochastic programming approach for UC problem is to minimize the total expected system operation cost over the representative scenarios [154].

The applicability of stochastic programming approach to solve the UC problem is demonstrated in [27, 155]. However, there are some challenges in adopting the stochastic programming approach: a) construction of scenarios, b) reduction of scenarios, and c) measuring the quality of solution with respect to the true optimum [154].

Another stochastic optimization method which has been implemented to solve the UC problem in uncertain environment is robust optimization [156]. In contrast to stochastic programming models which requires the information of underlying probability distribution of the uncertainty, robust unit commitment (RUC) models require only moderate information about the underlying uncertainty, such as the mean and the range of the uncertain data [157]. Further, in contrast to stochastic programming models in which the total expected cost is minimized, robust unit commitment models minimize the worst-case cost regarding all possible outcomes of uncertain parameters. The applicability of robust optimization based approach to solve the UC problem is demonstrated in [157–160]. The drawback of the robust unit commitment models is that the solutions obtained can be very conservative.

5.3 Proposed Work and the Motivation

The literature survey shows that the unit commitment problem has been rarely solved in an uncertain environment considering system operation cost, emission and reliability as the multiple conflicting objectives. Although, the study presented in [152] considered the aforementioned objectives, the limitation is that only the best compromise solution was presented and not the entire trade-off solutions. Further, the study presented in [23] also considered the three-objective optimization problem, the limitation is that reliability (EENS and LOLP) constraints were also incorporated in the optimization model. As discussed above, selecting an appropriate EENS limit and LOLP limit are difficult tasks and setting the limits arbitrarily may lead to sub-optimal solutions. Moreover, the NSGA-II based approach presented in [23] was not validated by comparing against other efficient MOEAs.

In this Chapter, a three-objective optimization model considering maximizing reliability as an additional objective along with minimizing system operation cost and emission is presented. The uncertainties occurring due to both unit outage and load forecast error are taken into account. These uncertainties are captured using expected energy not served (EENS) reliability index while EENS cost is used to reflect the reliability objective. The three-objective optimization problem of considering economic, emission and reliability objectives in the UC problem formulation is called MOEER-UC problem in this Chapter.

In Chapter 4, the MOEAs - NSGA-II-SBX, MOEA/D-SBX and MOEA-/D-DE were efficiently customized and proposed for solving the MOEE-UC problem. Among the proposed MOEAs, MOEA/D-DE was found to be the best algorithm for the bi-objective UC problem in deterministic environment. Thus, in this Chapter, MOEA/D-DE algorithm is applied to the MOEER-UC problem. However, for comprehensive comparison, MOEA/D-SBX and NSGA-II-SBX are also selected and applied to the MOEER-UC problem.

Moreover, in this Chapter as well, a new non-uniform weight vector distribution strategy is proposed to enhance the performance of MOEA/D-DE on the MOEER-UC problem. Furthermore, MOEA/D-DE with ϵ-dominance based external archive is presented to obtain a well-distributed set of trade-off solutions.

5.4 Problem Formulation

In this Section, the MOEER-UC problem formulation is presented.

5.4.1 Objective Functions

1. System Operation Cost: The first objective function is to minimize the system operation cost (SOC), where SOC includes the fuel cost and the transition cost of all the generating units over the entire scheduling horizon [139]. The fuel cost ${f_i}^t$ of unit i is expressed as the quadratic function of its power output ${P_i}^t$ during hour t.

$$ {f_i}^t = a_i {P_i^t}^2 + b_i {P_i}^t + c_i \tag{5.1} $$

where a_i, b_i, c_i are the fuel cost coefficients of unit i.

The transition cost is the sum of the start-up costs and the shut-down costs. In this Chapter, the shut-down costs have not been taken into consideration in accordance with the literature [139] while the start-up cost is modeled as follows:

$$ SU_i^t = \begin{cases} HSC_i, & \text{if } \ MDT_i \leq T_{OFF,i}^t \leq MDT_i + T_{cold,i} \\ CSC_i, & \text{if } \ T_{OFF,i}^t > MDT_i + T_{cold,i} \end{cases} \tag{5.2} $$

where SU_i^t is the start-up cost of unit i at hour t, HSC_i and CSC_i represents the hot start cost and cold start cost of unit i, respectively, MDT_i

represents the minimum down time of unit i, $T_{OFF,i}^{t}$ is the continuous off time of unit i up to hour t and $T_{cold,i}$ is the cold start cost of unit i.

Subsequently, the first objective function (F_1) is given by minimization of the following cost function [139].

$$F_1 = \sum_{t=1}^{T_{\max}} \sum_{i=1}^{N} \left(f_i^t . u_i^t + SU_i^t \left(1 - u_i^{t-1}\right) u_i^t \right) \tag{5.3}$$

where u_i^t represents the unit commitment status of unit i at hour t ($1 = ON$, $0 = OFF$), T_{max} is the number of hours in the scheduling horizon and N is the number of thermal generating units in the system.

2. Emission: The second objective function (F_2) is the reduction of emission of air-pollutants into the atmosphere [139].

$$F_2 = \sum_{t=1}^{T_{\max}} \sum_{i=1}^{N} \left(E_i^t . u_i^t \right) \tag{5.4}$$

where ${E_i}^t$ (lb) represents the quantity of pollutants produced by unit i at time t and is defined as

$${E_i}^t = a_{1i} {P_i^t}^2 + b_{1i} P_i^t + c_{1i} \tag{5.5}$$

and a_{1i}, b_{1i}, c_{1i} are the emission coefficients of unit i.

3. Expected Energy Not Served (EENS) Cost

The third objective function (F_3) is to maximize the reliability of the system. The function used to represent the reliability of the system is the expected energy not served (EENS) cost [151] which is defined as the product of the expected energy not served (EENS) and a value of lost load (VOLL) determined using survey [161]. It is noted that VOLL represents the average value (in \$/MWh) that consumers place on the accidental loss of 1 MWh of electricity [151]. Since, predicting the generation outages and deviation of load demand from the forecasted demand during the actual implementation of a particular generation schedule is impossible, only an EENS cost (also called outage cost) can be computed. The EENS cost is given by

$$F_3 = VOLL \times EENS_{tot} \tag{5.6}$$

where $EENS_{tot}$ is total expected unserved energy for the entire scheduling horizon.

It is noted that the lower the EENS cost, the higher is the reliability of the system and vice-versa.

5.4.2 Constraints

1. System power balance: the total power generation at hour t must be equal to the load demand L^t for that hour.

$$\sum_{i=1}^{N} (P_i^t.u_i^t) = L^t, \quad t = 1, 2, ...T_{max} \tag{5.7}$$

2. Unit minimum up/down time: if a unit i is turned on/off, it must remain on/off for at least its minimum up/down time (MUT_i/MDT_i) duration.

$$\begin{aligned} T_{ON,i}^t &\geq MUT_i \\ T_{OFF,i}^t &\geq MDT_i \end{aligned} \tag{5.8}$$

where $T_{ON,i}^t$ and $T_{OFF,i}^t$ represent the continuous on and off time of unit i up to hour t, respectively.

3. Unit generation limits: for stable operation, the power output of each generator is restricted within its limits:

$$P_{min,i} \leq P_i^t \leq P_{max,i} \tag{5.9}$$

where $P_{min,i}$ and $P_{max,i}$ represent the rated lower and upper limit generation of unit i, respectively.

4. Maximum system operation cost:

   This constraint is incorporated as:

$$F_1 \leq SOC_{max} \tag{5.10}$$

   where F_1 represents the objective function system operation cost and SOC_{max} is the user-defined upper limit for solution's SOC.

5. Maximum Emission:

   This constraint is incorporated as:

$$F_2 \leq Emis_{max} \tag{5.11}$$

   where F_2 represents the objective function emission and $Emis_{max}$ is the user-defined upper limit for solution's emission.

6. Maximum EENS cost:

   This constraint is incorporated as:

$$F_3 \leq EENSC_{max} \tag{5.12}$$

   where F_3 represents the objective function EENS cost and $EENSC_{max}$ is the user-defined upper limit for solution's EENS cost.

In this Chapter, the MOEER-UC problem has been formulated by extending the MOEE-UC problem formulation considered in Chapter 4. Thus, the modifications made in the problem formulation considered in this Chapter as compared to the problem formulation considered in Chapter 4 are clearly highlighted below.

- Minimizing EENS cost (i.e., maximizing reliability) is added as an additional objective function to the existing objective functions - minimizing system operation cost and emission.

- The constraint related to system spinning reserve requirements is removed.

- The additional constraints related to maximum system operation cost, maximum emission and maximum EENS cost as given by (5.10), (5.11) and (5.12) are incorporated.

The reason behind removal of the spinning reserve (SR) requirement constraint is that in the presented approach, SR is implicitly scheduled according to the system operation cost and the level of reliability (i.e., EENS cost). In other words, the system operator does not need to determine a-priori the SR to be scheduled at each hour. The presented approach provides system operator with several trade-off optimal solutions and the system operator can select a particular solution according to the system operation cost and EENS cost.

The reasons behind incorporation of the constraints related to maximum system operation cost, maximum emission and maximum EENS cost as given by (5.10), (5.11) and (5.12), respectively are discussed in the case study 2 in Section 5.7.

5.5 Procedure for Calculation of EENS cost (i.e., Reliability objective)

In this Section, the procedure for calculation of the EENS cost in presence of uncertainty due to thermal unit outage and load forecast error is presented. The EENS cost calculation requires evaluating $EENS_{tot}$ (i.e., total expected unserved energy for the entire scheduling horizon) as EENS cost is obtained by just multiplying $EENS_{tot}$ with VOLL (see (5.6)). The method for incorporating the uncertainties and evaluating $EENS_{tot}$ is as follows:

5.5.1 Incorporating Uncertainty due to Thermal Unit Outage

Each thermal unit is considered as a two-state model, according to which a unit is either available or unavailable for generation. According to this model, the unavailability of the unit i during a short time interval LT (known as the system lead time) is given by

$$U_i(LT) = 1 - e^{\lambda_i LT} \tag{5.13}$$

where λ_i is the failure rate of unit i [162]. The probability $U_i(LT)$ given by (5.13) is known as the outage replacement rate (ORR) of the unit, i.e., the probability of losing capacity and not being able to replace it.

To calculate the $EENS_{tot}$ index for every chromosome, the conventional "loss of load" method is used, except that ORR is used instead of FOR (Forced Outage Rate) [162]. This method is based on the creation of the capacity outage probability table (COPT) according to the given load curve [163]. A COPT is formed for every hour using the ORR of all the committed units. The creation of COPT is based on the unit addition algorithm [163]. A COPT may be visualized as a table with n rows ($j = 1, 2, \ldots n$) and 3 columns. The first column represents n different generation levels that may be outaged. The second and third column represents the probability PR_j and the total capacity CR_j that remains in service corresponding to each outage level, respectively. The reliability index $EENS_t$ (i.e., expected energy not served for each hour t) is calculated as follows:

$$EENS_t = \sum_{j=1}^{n} PR_j.LOSS_j.(L^t - CR_j), \quad t \in [1, T_{max}] \tag{5.14}$$

where $LOSS_j$ is given by

$$LOSS_j = \begin{cases} 1, & if\ CR_j < L^t \\ 0, & otherwise \end{cases} \tag{5.15}$$

The EENS index of the entire scheduling horizon, $EENS_{tot}$ is given by

$$EENS_{tot} = \sum_{t=1}^{T_{max}} EENS_t \tag{5.16}$$

5.5.2 Incorporating Uncertainty due to Thermal Unit Outage and Load Forecast Error

The procedure presented in the above Section represents the method to evaluate $EENS_{tot}$ if the uncertainty due to only thermal unit outage is considered and the uncertainty due to load forecast error is neglected. However, load forecast is generally associated with uncertainty and hence should be considered in the UC problem. Thus, the procedure below represents the method to evaluate $EENS_{tot}$ in presence of both the uncertainties.

It is an accepted practice to assume that the forecast load consists of actual load plus a normally distributed error [162, 163]. The standard deviation (σ^t_{load}) of the load forecast error is equal to a percentage SL of the expected demand and depends upon the accuracy of the forecasting tool.

$$\sigma^t_{load} = SL \times L^t \tag{5.17}$$

The distribution representing the forecast demand can be divided into a discrete number of class intervals with the distribution mean being the net forecast demand and standard deviation given by (5.17). The load representing the class interval mid-point is assigned the designated probability for that class interval. It is recommended in [163] that a seven-step approximation (0, $\pm1\sigma$, $\pm2\sigma$, $\pm3\sigma$) to the normal distribution (known as seven-step model) is adequate to represent the uncertainty in demand forecast. Thus, with the assumption of demand forecast uncertainty to be normally distributed and represented by the seven-step model, the EENS index calculation for each hour t is given by

$$EENS_t = \sum_{m=1}^{7} (EENS_t(m)PL(m)), \quad t \in [1, T_{max}] \tag{5.18}$$

where $PL(m)$ and $EENS_t(m)$ indicate the probability and EENS value for hour t associated with the discrete class interval m in the seven-step model, respectively. It is noted that $EENS_t(m)$ is calculated using (5.14) and (5.15) by simply replacing L^t in the two equations by $L^t(m)$ where $L^t(m)$ represents the load for hour t associated with the discrete class interval m in the seven-step model. Once $EENS_t$ for every hour t is evaluated using (5.18), the EENS index of the entire scheduling horizon, $EENS_{tot}$ (as discussed above as well) is given by

$$EENS_{tot} = \sum_{t=1}^{T_{max}} EENS_t \tag{5.19}$$

5.5.3 Techniques Applied to Reduce the Computational Time in Evaluation of EENS cost

The evaluation of $EENS_t$ index (i.e., expected energy not served at every hour) of a chromosome is a very computationally intensive task [162]. This makes a MOEA computationally inefficient as the $EENS_t$ has to be calculated for every chromosome and at every hour. Considering the population size to be 300 and the terminating generation number to be 50,000 as in the case of a 60 unit system, requires $EENS_t$ index to be calculated 300 x 24 x 50,000 i.e., 360,000,000 times which is an enormous figure. Thus, the following techniques are applied to reduce the computational time in evaluation of the $EENS_t$ index:

- The computational time in creating COPT is reduced by omitting the outage levels for which the cumulative probabilities are less than a predefined limit, e.g., 10^{-7} [162].
- Additionally, to avoid the need for repeated creation of COPT, a memory archive is created to store the commitment patterns for each time period, and their corresponding $EENS_t$ index values. In subsequent generations, whenever a commitment pattern is repeated, the corresponding $EENS_t$ value is copied and assigned to the repeated pattern. This technique significantly reduces the computational time.

5.6 Proposed Algorithms

In this Chapter, the MOEAs proposed for the MOEER-UC problem are NSGA-II-SBX, MOEA/D-SBX and MOEA/D-DE. These MOEAs are developed for the MOEER-UC problem by extending the MOEAs proposed in Chapter 4 (for the MOEE-UC problem in deterministic environment). Thus, the similarities as well as the differences in the MOEAs proposed in this Chapter (for the MOEER-UC problem) and the MOEAs proposed in the Chapter 4 (for the MOEE-UC problem) are clearly highlighted below:

- **Similarities** - The basic framework of the MOEAs i.e., chromosome representation, crossover operators, mutation operators, constraint handling, etc. remains the same.

- **Differences** - Only modifications for the algorithm MOEA/D-DE are suggested to improve the performance of MOEA/D-DE on the three-objective UC problem. However, these modifications are discussed only later in the case studies presented in Section 5.7.

Thus, in a way, the MOEAs proposed in Chapter 4 for the MOEE-UC problem are applied (and modified as well in the case of MOEA/D-DE) in this Chapter to a different problem which is MOEER-UC problem. However, the MOEER-UC problem is more complex and challenging than the MOEE-UC problem because of the presence of an additional objective.

Since, the basic framework of the MOEAs remains the same as presented in Chapter 4; to avoid repetition, these are not presented again. Further, to avoid repetition, only the pseudo-code of MOEA/D-DE is presented as follows.

5.6.1 Steps of the Proposed Algorithm MOEA/D-DE

Input

- NP: the number of subproblems considered in MOEA/D-DE i.e., the population size;
- $\lambda_1, \lambda_2, ..., \lambda_{NP}$: a set of NP weight vectors;
- T: the neighborhood size;
- δ: the probability that parent solutions are selected from the neighborhood;
- n_r: the maximal number of solutions that can be replaced by each child solution.
- z: the initial reference point $(z_1, z_2, z_3) = (10^{30}, 10^{30}, 10^{30})$. The reference point initially has very large dimensions and is updated during the evolution of population.

At each generation, MOEA/D-DE maintains the following:

- A population of NP solutions $x_1, x_2, \ldots, x_{NP}$, where x_i is the current solution to the i_{th} subproblem.
- $F(x_1), F(x_2), \ldots, F(x_{NP})$, where $F(x_i) = \{F_1(x_i), F_2(x_i), F_3(x_i)\} \forall i = 1, 2, \ldots, NP$.
- $CV(x_i)$ = total constraint violation of x_i $\forall i = 1, 2, \ldots, NP$.
- $z = (z_1, z_2, z_3)$, where z_1, z_2 and z_3 are the best values found so far for objective F_1, F_2 and F_3, respectively.

The steps executed are as follows.

- **Step 1: Initialization**
  - **Step 1.1** Compute the Euclidean distances between any two weight vectors and then calculate T closest weight vector to each λ_i. For all $i = 1, 2, \ldots, NP$, set $B(i) = \{i_1, i_2, \ldots, i_T\}$, where λ_j, $\forall j \in B(i)$ are T closest vectors to λ_i.
  - **Step 1.2** Generate initial population randomly.
  - **Step 1.3** For all $i = 1, 2, \ldots, NP$, repair x_i for load demand equality constraint violation.
  - **Step 1.4** Calculate $CV(x_i)$ and $F(x_i)$ i.e., $\{F_1(x_i), F_2(x_i), F_3(x_i)\}$.
  - **Step 1.5** Update $z = (z_1, z_2, z_3)$ according to the condition: $z_j = \min_{1 \leq i \leq NP} F_j(x_i)$ if x_i is feasible.
- **Step 2: Update**
  For $i = 1, 2, \ldots, NP$, **do**
  - **Step 2.1 Selection of Mating/Update Range:** Uniformly generate random number $rand$ from [0,1]. Then,

$$P = \begin{cases} B(i), & \text{if } rand < \delta \\ 1, 2, \ldots, NP, & \text{otherwise} \end{cases}$$

  - **Step 2.2 Reproduction:**
    1. Randomly select three indices r_1, r_2 and r_3 from P which are different from i.
    2. Decode x_k in UCM_k and RPM_k, where $k = i, r_1, r_2, r_3$.
    3. Generate a solution UCM_{child} using GA recombination operators on UCM_k, where $k = i, r_1$.
    4. Generate a solution RPM_{child} using DE recombination operators on RPM_k, where $k = r_1, r_2, r_3$.
    5. Encode UCM_{child} and RPM_{child} in x_{child}.

- **Step 2.3 Repair:** Repair x_{child} for boundary constraint violation and load demand equality constraint violation.
- **Step 2.4:** Calculate $CV(x_{child})$ and $F(x_{child})$ i.e., $\{F_1(x_{child}), F_2(x_{child}), F_3(x_{child})\}$.
- **Step 2.5 Update of z:** For $j = 1, 2$ do
  1. If x_{child} is feasible and $z_j > F_j(x_{child})$ then set $z_j = F_j(x_{child})$
- **Step 2.6 Replacement/Update of Solutions:** Set $c = 0$ and then do
  1. Set $flag = 0$.
  2. If $c = n_r$ or P is empty, $i = i + 1$ and go to **Step 2.1**, else randomly pick an index j from P.
  3. Determine if x_{child} replaces x_j or not according to the replacement rules.
  4. If x_{child} replaces x_j then $flag = 1$ and $c = c + 1$.
  5. If $flag = 1$, remove j from P and go to **Step 2.6.1**.

- **Step 3: Stopping Criteria**
  If termination criterion is satisfied, then stop else go to **Step 2**.

Output

- Approximation to Pareto-optimal solutions: $\{x_1, x_2, \ldots, x_{NP}\}$.
- Approximation to Pareto-optimal front: $\{F(x_1), F(x_2), ..., F(x_{NP})\}$.

5.7 Experimental Study

In this Section, extensive case studies are undertaken to investigate the performance of the proposed algorithms - NSGA-II-SBX, MOEA/D-SBX and MOEA/D-DE on the MOEER-UC problem. The experimental evaluation is systematically divided into 5 case studies.

1. In the first case study, MOEA/D-DE is implemented considering system operation cost and EENS cost as the multiple objectives;
2. In the second case study, all the three proposed algorithms are implemented considering system operation cost, emission and EENS cost as the multiple objectives in the constrained objective space;

3. Thereafter, in the third case study, the proposed algorithms are implemented considering system operation cost, emission and EENS cost as the multiple objectives in the unconstrained objective space;
4. In the fourth case study, a novel non-uniform weight vector distribution strategy is proposed to improve the performance of MOEA/D-DE on the three-objective optimization problem; and
5. Finally, in the fifth case study, MOEA/D-DE with external archive is presented to enhance the overall performance of MOEA/D-DE on the MOEER-UC problem.

The proposed MOEAs are developed on C++ platform. The MOEAs are tested on the MOEER-UC problem for power systems with 10, 20 and 60 units in a 24 hour scheduling horizon [139]. The lead time of the system is fixed as 4 hours [162] while the standard deviation (σ^t_{load}) of the load forecast error is assumed to be 5% of the hourly load demand as suggested in [164]. Further, VOLL is assumed to be 5000 $\$/MWh$ [151]. For each experiment, 15 independent simulation trials are conducted to verify the potential of the proposed algorithms. The inverted generational distance (IGD) is used as the performance metric to investigate the performance of the proposed algorithms in this Chapter as well (like in Chapter 4). The common parameters of the three MOEAs like population size and generation number are summarized for different test systems in Table 5.1. The rest of the algorithmic parameters corresponding to NGSA-II-SBX, MOEA/D-SBX and MOEA/D-DE were kept the same as shown in Tables 4.2, 4.3 and 4.4, respectively in Chapter 4.

Table 5.1 Common parameter settings of MOEAs corresponding to different test systems

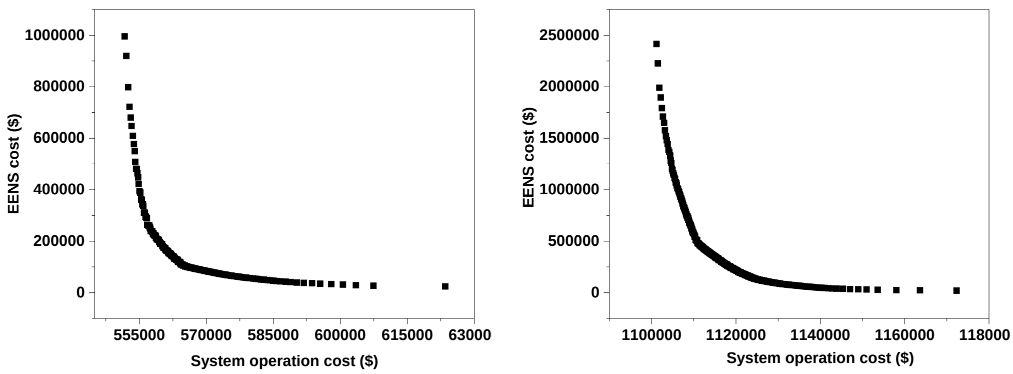
Test system	10-unit system	20-unit system	60-unit system
Population size	300	300	300
Generation number	10000	20000	50000

5.7.1 Case Study 1 - Study on considering system operation cost and EENS cost as the multiple objectives in unconstrained objective space

In this case study, MOEA/D-DE is implemented considering system operation cost and EENS cost as the multiple objectives. The emission objective and the constraint related to maximum emission i.e., (5.11) is neglected here as the primary aim of this case study is to demonstrate the relationship between the two objectives - system operation cost and EENS cost. Moreover, the objective space is unconstrained i.e., the constraints related to SOC_{max} and $EENSC_{max}$ given by (5.10) and (5.12), respectively are not considered. Fig. 5.1a and 5.1b show the trade-off front obtained using MOEA/D-DE corresponding to 10 and 20 unit system, respectively. These figures clearly show that system operation cost and EENS cost objectives are conflicting in nature.

It is also interesting to note that a knee region (sharp bend in the curve) exists in each of the trade-off front obtained in Fig. 5.1a and 5.1b. This region is important in multi-objective optimization problems because beyond this region the front is steep which implies that there is a sharp decrease in one objective with a slight increase in the other objective [165]. Such characteristic of the knee solutions make them unique to decision makers for practical applications. A knee region can be visually identified as a convex bulge in the Pareto-optimal front [165]. According to [166], knee on the P-O front corresponds to farthest solution from the line formed by joining the extreme solutions on the Pareto-optimal front. The neighboring solutions to the knee on the P-O front are called the knee solutions.

The reason for which system operation cost and EENS cost objectives are conflicting is the fact that system operation cost increases when more



(a) 10 unit system

(b) 20 unit system

Fig. 5.1 Non-dominated solutions obtained by MOEA/D-DE with system operation cost and EENS cost as the multiple objectives.

number of units are committed while the same leads to increase in reliability of the system i.e., decrease in EENS cost. However, this case study experimentally proves that system operation cost and EENS cost are conflicting objectives. The system operator may adopt the bi-objective optimization model considered in this case study if he/she is not concerned with the emission and is satisfied with the trade-off solutions obtained with respect to system operation cost and EENS cost. However, as the main aim of this Chapter is to present trade-off solutions to the system operator with respect to all the three objectives; the remaining case studies consider the three-objective optimization model discussed in the problem formulation.

5.7.2 Case Study 2 - Study on considering system operation cost, emission and EENS cost as multiple objectives in constrained objective space

In this case study, the proposed MOEAs are implemented considering system operation cost, emission and EENS cost as the multiple objectives in the constrained objective space. The reasons behind incorporation of the constraints related to maximum system operation cost, maximum emission and maximum EENS cost as given by (5.10), (5.11) and (5.12), respectively and solving the problem in the constrained objective space are as follows:

- The three-objective dimensional space for the problem under consideration is extremely large and thus solving the problem in the unconstrained objective space may require a very large population size and increase the computational cost manifold.
- Moreover, the system operators would not like to see a solution which is very poor in any of the three objectives. Thus, solving the problem in the constrained objective space can provide only good solutions to the system operators for easier decision making.
- Further, it was observed in case study 1 that a knee region exists in the trade-off front corresponding to system operation cost and EENS cost. In such optimization problems, the knee region is the most desired as the solutions outside this region do not offer a good trade-off to the decision maker.

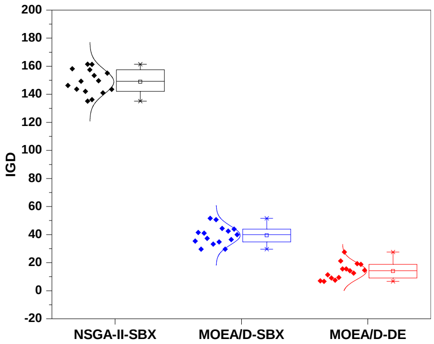
Overall, in the constrained objective space, the algorithms can provide better approximation of the Pareto-optimal surface and the system operator can obtain selected solutions for better decision making. However, care must be taken to set the values corresponding to SOC_{max}, $Emis_{max}$ and $EENSC_{max}$. This is because if the problem is solved in highly constrained objective space then either the algorithm may not be able to find any feasible solution or may perform poorly. In this study, the values corresponding to SOC_{max} and $EENSC_{max}$ were determined by first solving the bi-objective optimization problem in the unconstrained objective space as discussed in case study 1. The SOC_{max} and $EENSC_{max}$ values were chosen such that the knee region and certain region beyond the knee region is preserved. Moreover, the $Emis_{max}$ value was decided based on the results obtained in the previous Chapter in which system operation cost and emission were considered as the multiple objectives in deterministic environment. This led to informed decision making regarding the setting of SOC_{max}, $Emis_{max}$ and $EENSC_{max}$ values and these values set corresponding to different test systems are summarized in Table 5.2.

Table 5.2 Settings corresponding to constraints on objective functions for different test systems

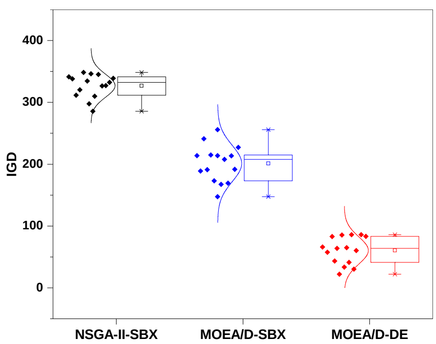
Test system	10-unit system	20-unit system	60-unit system
SOC_{max} (\$)	600,000	1,150,000	3,400,000
$Emis_{max}$ (\$)	45,000	80,000	230,000
$EENSC_{max}$ (\$)	300,000	500,000	1,200,000

Fig. 5.2a, 5.2b and 5.2c illustrate the IGD metric comparison between NSGA-II-SBX, MOEA/D-SBX and MOEA/D-DE on 10, 20 and 60 unit system, respectively. It is observed from these figures that the best, mean and median IGD of MOEA/D-DE is significantly lower than that of NSGA-II-SBX and MOEA/D-SBX on all the test systems and thus MOEA/D-DE is significantly superior to NSGA-II-SBX and MOEA/D-SBX in terms of IGD metric.

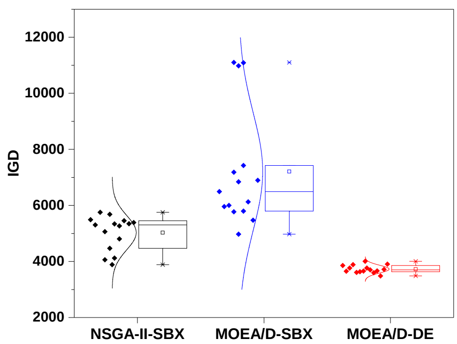
Fig. 5.3a, 5.3b and 5.3c show the distribution of the final non-dominated solutions found by MOEA/D-DE, MOEA/D-SBX and NSGA-II-SBX with the lowest IGD values on 10 unit system, respectively. Similarly, Fig. 5.4a, 5.4b and 5.4c show the distribution of the final non-dominated solutions found by MOEA/D-DE, MOEA/D-SBX and NSGA-II-SBX with the lowest IGD values on 20 unit system, respectively. The superior performance of MOEA/D-DE, particularly with respect to NSGA-II-SBX, is clearly visible in these figures as MOEA/D-DE is able to obtain much uniformly distributed solutions as compared to NSGA-II-SBX.



(a) 10 unit system



(b) 20 unit system



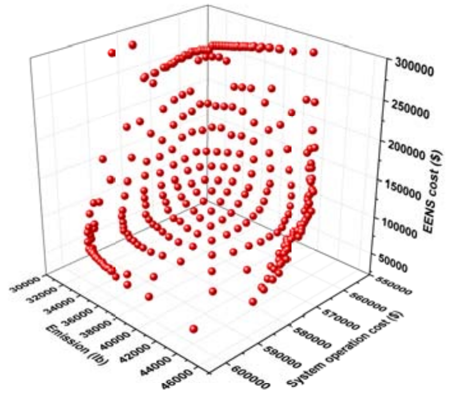
(c) 60 unit system

Fig. 5.2 IGD metric results with respect to performance of proposed MOEAs.

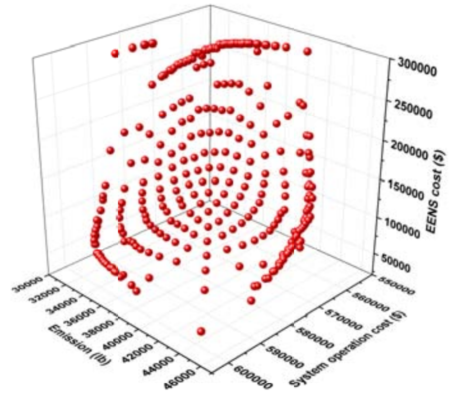
Since, MOEA/D-DE is found to significantly outperform NSGA-II-SBX and MOEA/D-SBX in solving the MOEER-UC problem, the rest of the Chapter is devoted to further investigating the performance of MOEA/D-DE.

5.7.3 Case Study 3 - Study on system operation cost, emission and EENS cost as the multiple objectives in unconstrained objective space

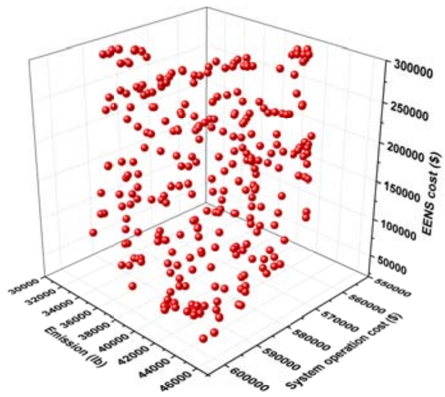
Although, MOEA/D-DE outperforms MOEA/D-SBX and NSGA-II-SBX on the three-objective optimization problem in the constrained objective



(a) MOEA/D-DE



(b) MOEA/D-SBX

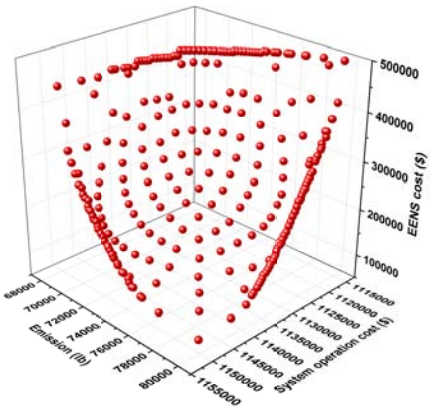


(c) NSGA-II-SBX

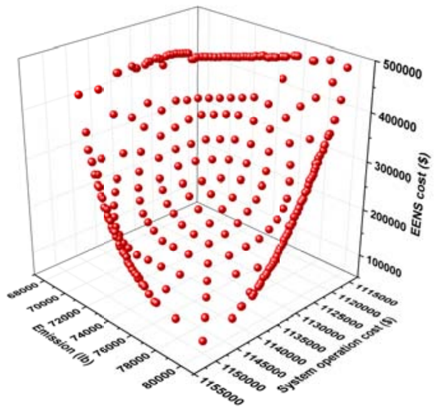
Fig. 5.3 3-D scatter plot of proposed MOEAs with best IGD metric on 10 unit system.

space (as observed in case study 2), there seems to be a limitation in the performance of MOEA/D-DE as many solutions are clustered at the boundary of the constrained objective space (see Fig. 5.3a and 5.4a). This is also observed in the results obtained using MOEA/D-SBX (see Fig. 5.3b and 5.4b). Thus, an investigation is undertaken in this case study to analyze if MOEA/D-DE is able to perform satisfactorily in the constrained objective space.

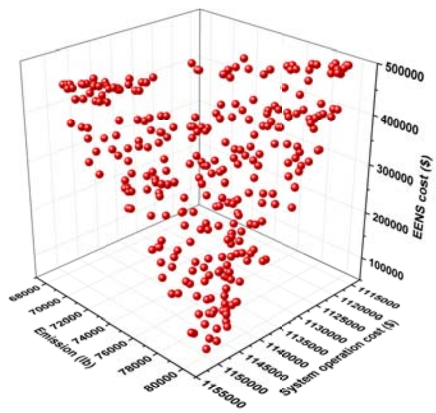
In this case study, MOEA/D-DE is implemented considering system operation cost, emission and EENS cost as the multiple objectives in the unconstrained objective space. This means that the constraints related to



(a) MOEA/D-DE



(b) MOEA/D-SBX

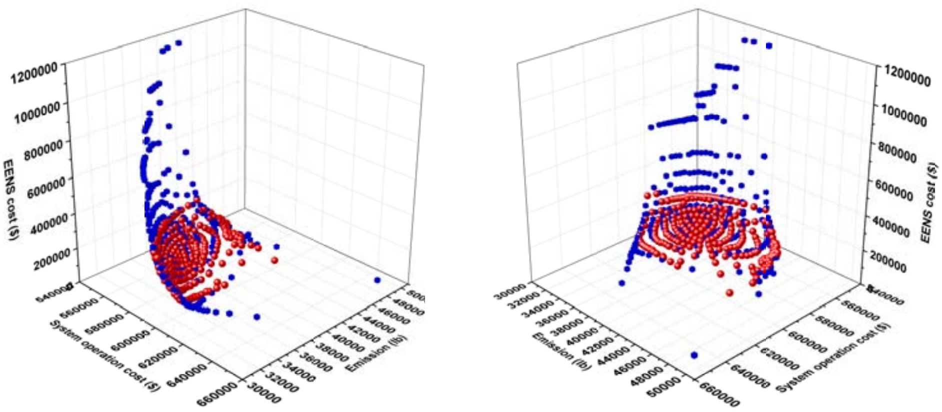


(c) NSGA-II-SBX

Fig. 5.4 3-D scatter plot of proposed MOEAs with best IGD metric on 20 unit system.

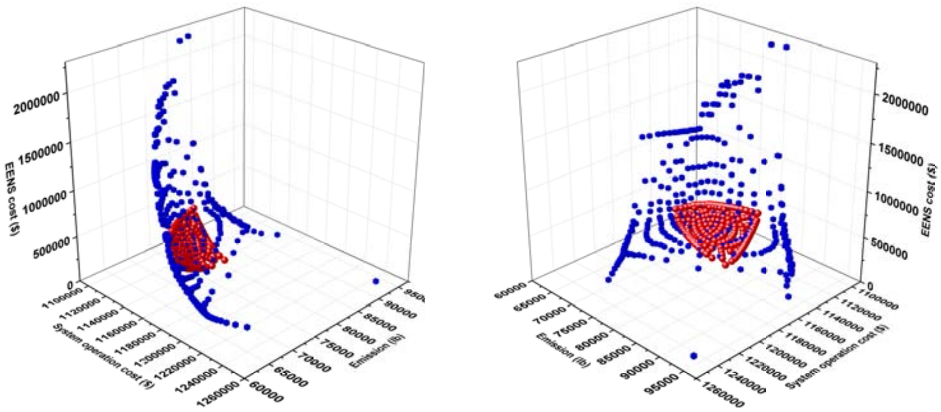
SOC_{max}, $Emis_{max}$ and $EENSC_{max}$ as mentioned in the problem formulation are relaxed in this case study. The rest of the algorithmic parameters and constraints remain the same. MOEA/D-DE is executed on 10 and 20 unit systems in the unconstrained objective space and the results obtained are compared with the results obtained in the constrained objective space.

Fig. 5.5a and 5.5b show the side view and the front view, respectively, of the distribution of the final non-dominated solutions found by MOEA/D-DE in the constrained and the unconstrained objective space on the 10 unit system. Similarly, Fig. 5.6a and 5.6b show the side view and the front view, respectively, of the distribution of the final non-dominated solutions found by MOEA/D-DE in the constrained and the unconstrained objective space on the 20 unit system.



(a) Side view (b) Front view

Fig. 5.5 Non-dominated solutions obtained by MOEA/D-DE in constrained objective space (represented by red balls) and unconstrained objective space (blue cubes) on the 10 unit system.



(a) Side view (b) Front view

Fig. 5.6 Non-dominated solutions obtained by MOEA/D-DE in constrained objective space (represented by red balls) and unconstrained objective space (blue cubes) on the 20 unit system.

It is observed from these figures that in both the cases i.e., on 10 and 20 unit system, the solutions found by MOEA/D-DE in the constrained objective space belong to a region which is a subset of the region in which MOEA/D-DE in the unconstrained objective space found the solutions.

Thus, the comparison highlights the following:

- MOEA/D-DE is able to perform satisfactorily in the constrained objective space as compared to the unconstrained objective space.
- The solutions explored by MOEA/D-DE in the unconstrained objective space illustrate the vastness of the objective space. Although, MOEA/D-DE is able to perform well in the unconstrained objective space yet constraining the objective space is better as it presents selected solutions to the system operator and can enhance decision making.
- The reason for which MOEA/D-DE is not able to maintain a uniform distribution of solutions on the trade-off surface may be because there is no explicit diversity maintenance operator in the framework of original MOEA/D [85].
- A method needs to be devised to improve the performance of MOEA/D-DE and obtain a better distribution of solutions in the middle of the trade-off surface.

5.7.4 Case Study 4 - Study on MOEA/D-DE with non-uniform weight vector distribution

Although, the case study 2 shows that MOEA/D-DE outperforms MOEA/D-SBX and NSGA-II-SBX on the three-objective optimization problem and the case study 3 shows that MOEA/D-DE is able to perform satisfactorily in the constrained objective space as compared to the unconstrained objective space yet, the clustering of solutions obtained by MOEA/D-DE along the boundary of the constrained objective space is a limitation as it reduces the number of solutions obtained in the middle of the trade-off surface.

As discussed in Chapter 4, different strategies like local search, etc. can be incorporated to enhance the performance of an MOEA. However, as in Chapter 4, non-uniform weight-vector distribution (NUWD) strategy was found to improve the performance of MOEA/D-DE; in this case study as well, a new NUWD strategy is incorporated within the framework of MOEA/D-DE for the three-objective optimization problem.

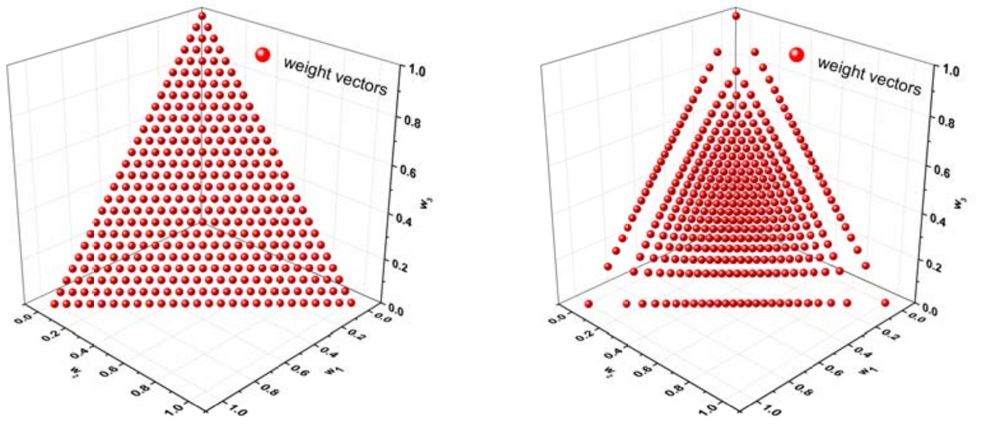
The target of the proposed NUWD strategy is to help MOEA/D-DE achieve a better distribution in the middle of the trade-off surface while maintaining the performance of MOEA/D-DE in terms of convergence throughout the trade-off surface. Thus, in the proposed NUWD strategy, search directions are concentrated more towards the middle with slight compromise along the edges i.e., more sub-problems are allocated towards the middle and relatively fewer sub-problems along the edges. The following function is selected to generate non-uniformly distributed weight vectors:

$$\lambda_i^{k'} = g(\lambda_i^k) = (acos(2\lambda_i^k - 1)/\pi) \quad i = 1, 2, \ldots, NP; k = 1, 2. \tag{5.20}$$

where $\lambda_i^{k'}$ replaces λ_i^k as input in the algorithm MOEA/D-DE.

Fig. 5.7a and 5.7b show the uniform weight-vector distribution employed for a three-objective optimization problem in the original MOEA/D-DE [86] and the non-uniformly distributed weight vectors generated using the proposed strategy employing the function mentioned above, respectively. It is observed in Fig. 5.7b that as desired, the NUWD strategy has weight vectors concentrated more towards the middle than towards the edges.

Next, the NUWD strategy is incorporated within MOEA/D-DE and the performance of the resulting algorithm, MOEA/D-DE/NUWD, is investi-

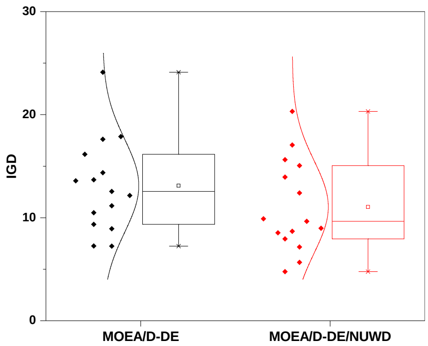


(a) (b)

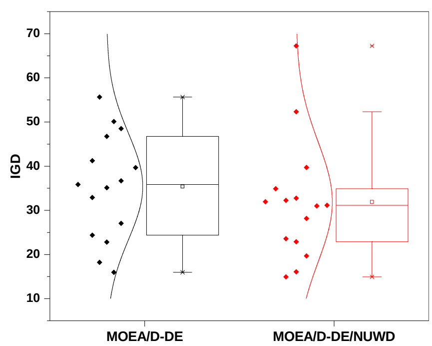
Fig. 5.7 (a) Uniform weight vector distribution in the original MOEA/D-DE for 3-objective optimization problem, (b) Proposed non-uniform weight vector distribution for 3-objective optimization problem.

gated by comparing it against MOEA/D-DE (i.e., with the UWD strategy). Fig. 5.8a, 5.8b and 5.8c show the comparison of MOEA/D-DE and MOEA/D-DE/NUWD on the basis of IGD metric for 10, 20 and 60 unit system, respectively. It is observed from the figures that on 10 and 20 unit system, MOEA/D-DE/NUWD performs slightly better than MOEA/D-DE while on 60 unit system, MOEA/D-DE performs slightly better than MOEA/D-DE/NUWD. Overall, the performance of MOEA/D-DE/NUWD seems comparable to that of MOEA/D-DE.

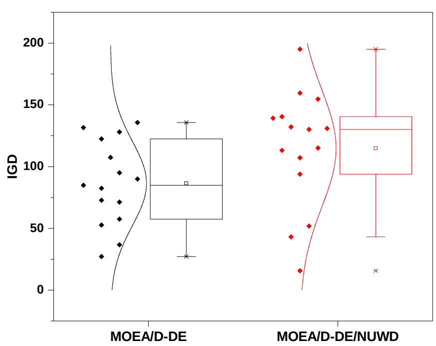
To further analyze how MOEA/D-DE and MOEA/D-DE/NUWD compare in the objective space, the distribution of the final non-dominated solutions with the lowest IGD values found by MOEA/D-DE and MOEA/D-DE/NUWD on 10, 20 and 60 unit system are plotted in Fig. 5.9a, 5.9b and 5.9c, respectively. It is visually evident from the figures that the proposed NUWD strategy enhances the performance of MOEA/D-DE/NUWD (in comparison to MOEA/D-DE) on different test systems by providing much better distribution of solutions towards the middle of the trade-off surface.



(a) 10 unit system



(b) 20 unit system

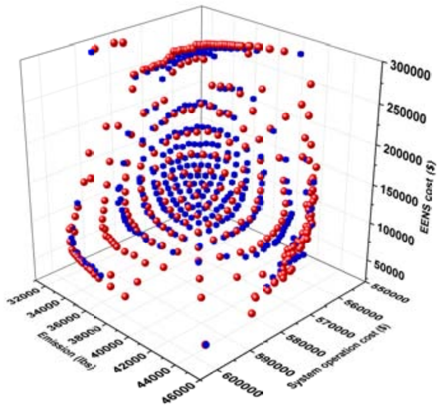


(c) 60 unit system

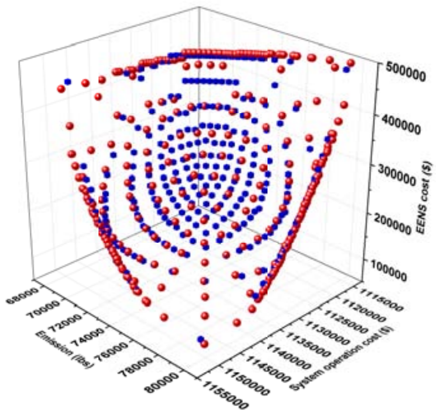
Fig. 5.8 IGD metric results for MOEA/D-DE with uniform and (proposed) non-uniform weight vector distribution.

The comparative analysis of MOEA/D-DE and MOEA/D-DE/NUWD leads to the following inferences:

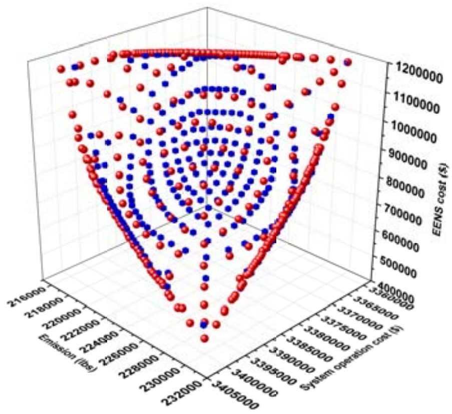
- The performance of MOEA/D-DE with uniform weight vector distribution strategy and the proposed non-uniform weight vector distribution strategy is comparable in terms of IGD metric comparison.
- However, MOEA/D-DE/NUWD provides much better distribution of solutions as compared to MOEA/D-DE.
- Even with MOEA/D-DE/NUWD, there is clustering of solutions towards the edges of the trade-off surface.



(a) 10 unit system



(b) 20 unit system



(c) 60 unit system

Fig. 5.9 The distribution of the final non-dominated solutions found (with the lowest IGD values) by MOEA/D-DE with uniform (represented by red balls) and (proposed) non-uniform weight vector distribution (blue cubes).

- Thus, there is scope for further improvement in the performance of the proposed MOEA/D-DE on the MOEER-UC problem.

5.7.5 Case Study 5 - Study on MOEA/D-DE with external archive

The case study 4 showed that although MOEA/D-DE with non-uniform weight-vector distribution can provide much better distribution of solutions than MOEA/D-DE yet there is scope for further improvement in achieving even better distribution. One of the strategy to obtain better distribution of solutions can be to use a larger population size. However, that may increase the computational cost manifold. Another strategy can be to incorporate an external archive into the algorithmic framework. Thus, in this case study, MOEA/D-DE with an external archive is investigated to in order to achieve a better distribution of solutions.

Several MOEAs have been proposed in the literature which maintain an external archive. The purpose of an archive can be twofold -

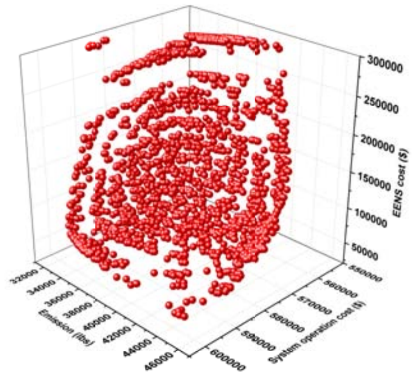
- It can be used to maintain an approximation of the set of non-dominated solutions visited by the MOEA during multi-objective optimization.

- Along with maintaining an approximation of the set of non-dominated solutions, it can be used to guide the search process of the MOEA as well.

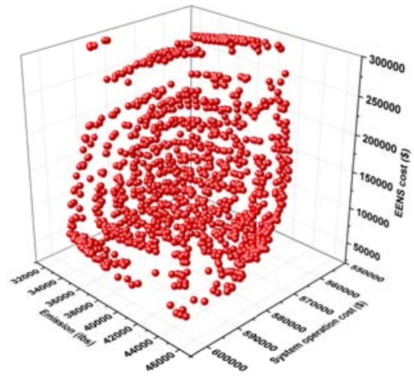
Some examples of archive based MOEAs are PAES [79], SPEA [80], SPEA2 [82], MOPSO [167], ϵ-MOEA [168]. One of the strategies of constructing archive can be to store all the non-dominated solutions found by the MOEA during the search process. However, this strategy is rarely used as it results in an unbounded archive and the decision maker would never like to have such a large set of solutions for decision making. Further, the size of the true non-dominated set may be exponentially large or even infinite and the complexity of the archive updating operator increases with the archive size. Thus, most of the archiving strategies presented in the literature use some method to restrict the archive size. Usually, all the non-dominated solutions are stored in the archive until it reaches it pre-defined limit. Thereafter, the archive size is limited using both Pareto-dominance and some diversity preservation technique. For example, to limit the size of archive, a clustering mechanism is used in SPEA [80] and SPEA-II [82] while an adaptive grid based mechanism is used in MOPSO [167].

The archiving strategy proposed in the original MOEA/D study [85] is an unbounded archive. Since, an unbounded archive is generally not desired (as discussed above), in this case study, MOEA/D-DE with an external archive is implemented where the external archive is updated using the ϵ-dominance principle as in ϵ-MOEA [168]. However, it is noted that unlike ϵ-MOEA, the ϵ-dominance based archive is used with MOEA/D-DE in this case study only to store an approximation of the set of non-dominated solutions and not to guide the algorithm in its search process. The advantage of an ϵ-dominance based archive is that it allows the decision maker to control the resolution of the Pareto set approximation by choosing an appropriate ϵ value corresponding to each objective.

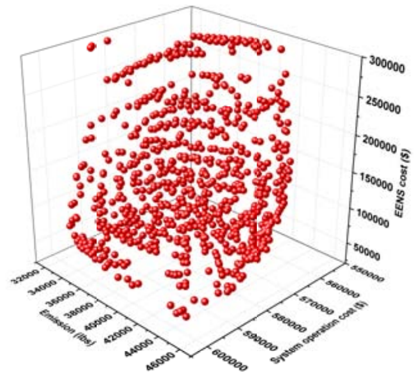
MOEA/D-DE with ϵ-dominance based external archive was implemented on 10 and 20 unit systems considering different ϵ values. Fig. 5.10a, 5.10b and 5.10c illustrate the distribution of the solutions obtained in the external archive by MOEA/D-DE on the 10 unit system with ϵ set as 100, 250 and 400, respectively, corresponding to each objective. Similarly, Fig. 5.11a, 5.11b and 5.11c illustrate the distribution of the solutions obtained in the external archive by MOEA/D-DE on the 20 unit system with ϵ set as 100, 250 and 400, respectively, corresponding to each objective. It is observed in these figures that MOEA/D-DE with external archive is able to provide significantly better distribution of solutions throughout the trade-off surface as compared to that obtained by MOEA/D-DE and MOEA/D-DE/NUWD. Further, it is observed that the resolution of the Pareto set approximation is efficiently controlled by ϵ value as for both the systems, ϵ at 250 returns a less crowded set of solutions than ϵ at 100 and ϵ at 400 returns a less crowded set of solutions than ϵ at both 100 and 250.



(a) Epsilon = 100



(b) Epsilon = 250



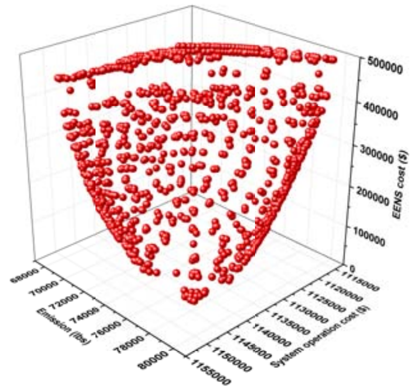
(c) Epsilon = 400

Fig. 5.10 The distribution of the final solutions obtained in the external archive by MOEA/D-DE with different settings of epsilon on the 10 unit system.

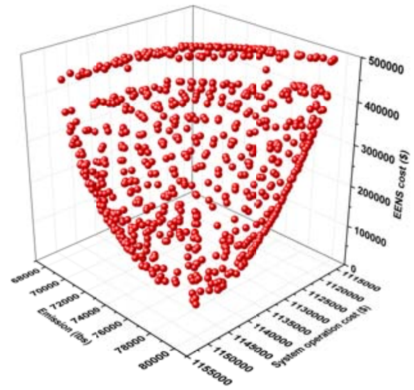
Thus, MOEA/D-DE with ϵ-dominance based external archive is established as the best algorithm among the proposed MOEAs in this Chapter for solving the MOEER-UC problem.

5.8 Summary

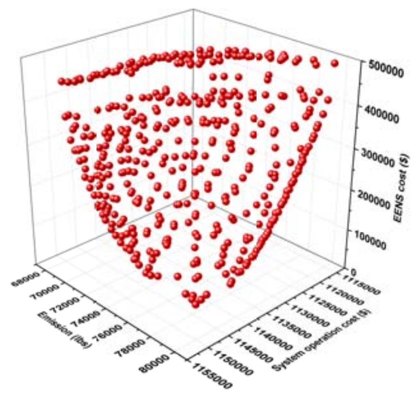
In this Chapter, NSGA-II-SBX, MOEA/D-SBX and MOEA/D-DE were efficiently extended and applied to solve the UC problem considering system operation cost, emission and reliability as the multiple conflicting objectives in uncertain environment. The three proposed MOEAs were exhaustively compared among themselves on a range of test systems and MOEA/D-DE



(a) Epsilon = 100



(b) Epsilon = 250



(c) Epsilon = 400

Fig. 5.11 The distribution of the final solutions obtained in the external archive by MOEA/D-DE with different settings of epsilon on the 20 unit system.

was found to significantly outperform both NSGA-II-SBX and MOEA/D-SBX in terms of IGD metric comparison.

However, a limitation was observed in the performance of MOEA/D-DE as many solutions were found to be clustered at the boundary of the objective space. Therefore, a non-uniform weight vector distribution strategy was proposed for MOEA/D-DE in the three-objective space. The comparative analysis of MOEA/D-DE and MOEA/D-DE with the proposed NUWD strategy i.e., MOEA/D-DE/NUWD revealed that the latter provides much better distribution of solutions than MOEA/D-DE in the middle of the trade-off surface. However, it was observed that with MOEA/D-

DE/NUWD as well, more than desired solutions were concentrated towards the edges of the trade-off surface.

Therefore, MOEA/D-DE with an external archive was implemented so that a better approximation set of the non-dominated solutions explored by the algorithm during the optimization can be obtained. The external archive was updated using ϵ-dominance principle. The observation of the archived solutions in the objective space illustrated that with MOEA/D-DE based on external archive, the system operator can efficiently obtain uniformly distributed solutions in the trade-off surface.

In this Chapter, UC problem was solved in an uncertain environment considering economic, emission and reliability objectives. In the next Chapter, the wind-thermal UC problem is solved as a three-objective optimization problem.

Chapter 6

Multi-objective Day-Ahead Thermal Generation Scheduling in Presence of Significant Wind Penetration

6.1 Introduction

In this Chapter, the three-objective unit commitment problem formulation considered in Chapter 5 is extended to include significant wind penetration. The multiple objectives considered remain the same as in Chapter 5 i.e., minimizing system operation cost, minimizing emission and maximizing reliability. The uncertainties occurring due to thermal unit outage, load forecast error and wind forecast error are captured using expected energy not served (EENS) reliability index while EENS cost is used to reflect the reliability objective. The UC problem in presence of significant wind penetration considering system operation cost, emission and reliability as the multiple objectives is a a nonlinear, mixed-integer, combinatorial, high-dimensional, highly constrained, three-objective optimization problem. In Chapter 5, among the proposed MOEAs i.e., NSGA-II-SBX, MOEA/D-SBX and MOEA/D-DE, for the three-objective UC problem in uncertain environment, MOEA/D-DE was found to be the best performing MOEA. Thus, in this Chapter, only MOEA/D-DE and its variants proposed in Chapter 5 are applied to the three-objective wind-thermal UC problem.

The rest of the Chapter is organized as follows. Section 6.2 discusses the related work on wind-thermal unit commitment problem. The proposed work and the motivation is presented in Section 6.3. The three-objective UC problem formulation in presence of significant wind penetration is presented in Section 6.4. The procedure for evaluating the reliability objective is presented in Section 6.5. The proposed algorithm is discussed in Section 6.6. The experimental study is presented in Section 6.7 and the Chapter is summarized in Section 6.8.

6.2 Related Work

As discussed in chapter 5, spinning reserve allows system operator (SO) to respond to unforeseen imbalances between load and generation caused by generation outages and load forecasting errors. Traditionally, the SR requirement has been based on deterministic criteria that the minimum amount of SR should be either a function of the largest capacity of the online generator, or greater than or equal to a fixed percentage (10%) of the load demand [1]. However, such deterministic criteria do not properly take into account the generation outages or the errors in load forecast i.e., the stochastic nature of the power system. Moreover, with the increased uncertainty due to unpredictable nature of wind, the deterministic criterion of deploying spinning reserve becomes all the more unreliable.

In [26, 169], special constraints relating to Up Spinning Reserve (USR) and Down Spinning Reserve (DSR) are adopted in the wind-thermal unit commitment problem. These additional reserve constraints are based on the notion that a large penetration of wind generation requires an increase in the SR requirement to minimize the risk of not meeting the demand and thus minimize the expected energy not served (EENS) cost. However, this approach is not economically justified as the reserve provision comes at a cost. Increasing SR requirement to accommodate higher wind power penetration will require a larger number of thermal generators to be synchronized. This can increase the system operation cost (SOC) to such a limit that it might be economically better to avoid the increase in SR requirement. Thus, there is a trade-off between the cost of providing reserve and the benefit it brings by reduction in the EENS cost.

In [170], an evolutionary iteration PSO (EIPSO) algorithm is presented to determine the optimal spinning reserve for a wind-thermal power system. The spinning reserve is distinguished into two types - up spinning reserve (USR) and down spinning reserve (DSR) and it is demonstrated that the optimal spinning reserve (both USR and DSR) of a wind-thermal power system is achieved when the total cost i.e., sum of total operation cost and EENS cost (also called outage cost) is minimum.

In [28], an approach is proposed in which the SR is scheduled to meet a pre-specified system (wind-thermal) reliability level. The disadvantage in this approach is that it is not possible to determine a system's optimal reliability level in advance. Thus, fixing the same reliability level for different systems would result in sub-optimal solutions.

In [171], the SR requirement is considered to be α times the standard deviation σ_{net} of the net forecast demand (load forecast – wind forecast) error. The drawback in this approach is that it procures larger amounts of

SR as the installed wind capacity (and wind power production) increases and does not take into account the cost and benefit of SR provision.

In [151], an approach based on minimizing the total cost (i.e., sum of SOC and EENS cost) is proposed to determine the optimal amount of SR for the traditional power system. In [172], the approach (based on minimizing the total cost) presented in [151] is extended to estimate the optimal SR requirement in systems with significant wind power penetration. However, the drawback in the approach presented in [151, 172] is that the SR optimization is not carried out as an inter-temporal optimization problem. Instead, in the first step, the inter-temporal couplings are neglected and the auxiliary computation determines the hourly SR using cost/benefit analysis. Thereafter, in the second step, the inter-temporal couplings are considered and the problem is solved as a reserve-constrained unit commitment problem. However, in this kind of approach, the risk cannot be traded among different hours and the obtained solution may be sub-optimal.

In [173], a stochastic optimization methodology based on scenario tree analysis to capture the uncertainty due to load forecast, wind forecast and unit outages is presented. The objective function to be minimized is considered as the expected cost of the system over the optimization period covering all of the scenarios. Stochastic and deterministic modes of optimization are compared and it is demonstrated that when the uncertainty of wind is taken into account as in stochastic optimization, it results in better performing and less costly schedules than the deterministic optimization. Further, it is demonstrated in [173] that more frequent scheduling of the system by adopting updated load and wind forecasts can lead to better capturing of the uncertainties and thus result in better performing schedules and more optimal results.

In [174], an improved gravitational search algorithm was proposed for wind-thermal unit commitment. The scenario analysis i.e., scenario generation and reduction method was adopted to incorporate the uncertainty due to load and wind forecasts on system operation. The objective function to be minimized is considered as the expected cost of the system over the optimization period covering all of the scenarios. In [175], a combination of quantum inspired binary gravitational search algorithm and chance constrained programming is proposed to solve the thermal UC problem with wind power integration. The objective function to be minimized is considered as the system operation cost in the study [175].

The impact of integrating large levels of wind generation on emissions reduction is studied in [176]. It is demonstrated in [176] that superior emission reduction benefits are observed when wind generation forecasts are included in the unit commitment and power dispatch decisions (the forecasted approach) rather than in the case if the wind generation is accommodated simply when its available (the fuel saver approach). The latter approach is called the fuel saver approach because it considers that the only benefit wind generation can provide is fuel-saving one. Thus, in such operational strategy whenever wind generation is available, some thermal units are deloaded to accommodate the wind generation. However, it assumes that a thermal unit can only deloaded to its minimum capacity and cannot be switched off. Therefore, the fuel saver approach is a simplistic strategy whereas in the forecasted approach, fewer thermal units are dispatched and thus the units run at higher efficiency (resulting in higher emission reduction). Further, it is also demonstrated in the study [176] that considerable CO_2 reductions can be obtained with increasing levels of installed wind capacity.

In [177], a methodology is presented to estimate the average displaced or reduced emission by integration of wind generation. It is demonstrated in [177] that the correlation factor between the wind generation time series and the power system's marginal emissions time series determines the average displaced emission by wind generation. If the correlation factor between the two time series is high, then the average displaced emission by wind generation is high and vice-versa.

A multi-objective evolutionary algorithm based on decomposition is proposed for economic emission dispatch of wind-thermal power system in [178]. The stochastic nature of wind power is modeled by Weibull probability distribution function and a chance-constrained programming method is adopted to simultaneously optimize cost and emission objectives of wind-thermal power system. The drawback in this approach is that the problem considered is economic emission dispatch while the unit commitment task is neglected.

Recently, a hybrid algorithm based on combination of PSO and sequential quadratic programming is proposed in [179] to solve the combined unit commitment and emission (CUCE) problem. In this study [179], the objective function to be minimized is considered as the sum of system operation cost and emission cost.

A detailed review on impacts of large-scale wind penetration on designing and operation of electric power systems is presented in [180]. Further, a detailed survey of models and algorithms for reliability based power systems planning and operation with wind power integration is presented in [181].

6.3 Proposed Work and the Motivation

The literature survey shows that the problem of wind-thermal unit commitment has been rarely addressed as a three-objective optimization problem considering simultaneous optimization of system operation cost, emission and reliability.

Thus, in this Chapter, the three-objective optimization model of maximizing reliability as an additional objective along with minimizing system operation cost and emission as presented in Chapter 5 for the MOEER-UC problem is adopted. The uncertainties occurring due to thermal unit outage, load forecast error and wind forecast error are taken into account. These uncertainties are captured using expected energy not served (EENS) reliability index and EENS cost is used to reflect the reliability objective. The three-objective wind-thermal UC problem is referred as MOWT-UC problem in this Chapter.

In Chapter 5, the MOEAs - NSGA-II-SBX, MOEA/D-SBX and MOEA-/D-DE were proposed for the MOEER-UC problem. Through comprehensive comparison, MOEA/D-DE was established as the best MOEA among the three MOEAs for solving the MOEER-UC problem. Since, the characteristics of the two problems i.e., MOEER-UC problem (solved in Chapter 5) and MOWT-UC problem addressed in this Chapter are the same because of the three-objective optimization model considered in both the problems; in this Chapter only MOEA/D-DE is applied to the MOWT-UC problem.

Along with MOEA/D-DE, the same variants of the algorithm as proposed in Chapter 5 i.e., MOEA/D-DE with non-uniform weight vector distribution and MOEA/D-DE with ϵ-dominance based external archive are applied to the MOWT-UC problem in this Chapter.

Thus, the motivation behind the work conducted in this Chapter is to demonstrate that the three-objective optimization model considered in the Chapter 5 can be conveniently extended to include significant wind generation. Further, the intention is to illustrate that the MOEA/D-DE proposed in Chapter 5 can efficiently obtain the trade-off optimal solutions for the MOWT-UC problem as well.

6.4 Problem Formulation

In this Section, the MOWT-UC problem formulation is presented.

6.4.1 Objective Function

1. System Operation Cost: The first objective function is to minimize the system operation cost (SOC), where SOC includes the fuel cost and the transition cost of all the thermal generating units over the entire scheduling horizon [139]. The fuel cost ${f_i}^t$ of thermal unit i is expressed as the quadratic function of its power output ${P_i}^t$ during hour t.

$$ {f_i}^t = a_i {P_i^t}^2 + b_i {P_i}^t + c_i \tag{6.1}$$

where a_i, b_i, c_i are the fuel cost coefficients of unit i.

The transition cost is the sum of the start-up costs and the shut-down costs. In this Chapter, the shut-down costs have not been taken into consideration in accordance with the literature [139] while the start-up cost is modelled as follows:

$$ SU_i^t = \begin{cases} HSC_i, & \text{if } \; MDT_i \leq T_{OFF,i}^t \leq MDT_i + T_{cold,i} \\ CSC_i, & \text{if } \; T_{OFF,i}^t > MDT_i + T_{cold,i} \end{cases} \tag{6.2}$$

where SU_i^t is the start-up cost of unit i at hour t, HSC_i and CSC_i represents the hot start cost and cold start cost of unit i, respectively, MDT_i represents the minimum down time of unit i, $T_{OFF,i}^t$ is the continuous off time of unit i up to hour t and $T_{cold,i}$ is the cold start cost of unit i.

Subsequently, the first objective function (F_1) is given by minimization of the following cost function [139].

$$F_1 = \sum_{t=1}^{T_{\max}} \sum_{i=1}^{N} \left(f_i^t . u_i^t + SU_i^t \left(1 - u_i^{t-1}\right) u_i^t \right) \tag{6.3}$$

where u_i^t represents the unit commitment status of unit i at hour t ($1 = ON$, $0 = OFF$), T_{max} is the number of hours in the scheduling horizon and N is the number of thermal generating units in the system.

2. Emission: The second objective function (F_2) is the reduction of emission of air-pollutants into the atmosphere [139].

$$F_2 = \sum_{t=1}^{T_{\max}} \sum_{i=1}^{N} \left(E_i^t . u_i^t \right) \tag{6.4}$$

where ${E_i}^t$ (lb) represents the quantity of pollutants produced by unit i at time t and is defined as

$${E_i}^t = a_{1i} {P_i^t}^2 + b_{1i} P_i^t + c_{1i} \tag{6.5}$$

and a_{1i}, b_{1i}, c_{1i} are the emission coefficients of unit i.

3. Expected Energy Not Served (EENS) Cost

The third objective function (F_3) is to maximize the reliability of the system. The function used to represent the reliability of the system is the expected energy not served (EENS) cost [151] which is defined as the product of the expected energy not served (EENS) and a value of lost load (VOLL) determined using survey [161]. It is noted that VOLL represents the average value (in $/MWh) that consumers place on the accidental loss of 1 MWh of electricity [151]. Since, predicting the generation outages and deviation of load demand from the forecasted demand during the actual implementation of a particular generation schedule is impossible, only an EENS cost (also called outage cost) can be computed. The EENS cost is given by

$$F_3 = VOLL \times EENS_{tot} \tag{6.6}$$

where $EENS_{tot}$ is total expected unserved energy for the entire scheduling horizon.

It is noted that the lower the EENS cost, the higher is the reliability of the system and vice-versa.

6.4.2 Constraints

1. System power balance: the total power generation by the wind-thermal system at hour t must be equal to the load demand L^t for that hour.

   $$\sum_{i=1}^{N} P_i^t + P_w^t = L^t, \quad t = 1, 2, ...T_{max} \tag{6.7}$$

   where P_i^t is the power produced by thermal unit i at hour t, P_w^t is the total wind power generation at hour t and L^t is the forecast demand at hour t.

2. Unit minimum up/down time: if a thermal unit i is turned on/off, it must remain on/off for at least its minimum up/down time duration.

   $$\begin{aligned} T_{ON,i}^t &\geq MUT_i \\ T_{OFF,i}^t &\geq MDT_i \end{aligned} \tag{6.8}$$

   where $T_{ON,i}^t$ and $T_{OFF,i}^t$ represent the continuous on and off time of unit i up to hour t, respectively.

3. Unit generation limits: for stable operation, the power output of each thermal generator is restricted within its limits:

   $$P_{min,i} \leq P_i^t \leq P_{max,i} \tag{6.9}$$

where $P_{min,i}$ and $P_{max,i}$ represent the rated lower and upper limit generation of unit i, respectively.

4. Maximum system operation cost:

   This constraint is incorporated as:

$$F_1 \leq SOC_{max} \tag{6.10}$$

   where F_1 represents the objective function system operation cost and SOC_{max} is the user-defined upper limit for solution's SOC.

5. Maximum Emission:

   This constraint is incorporated as:

$$F_2 \leq Emis_{max} \tag{6.11}$$

   where F_2 represents the objective function emission and $Emis_{max}$ is the user-defined upper limit for solution's emission.

6. Maximum EENS cost:

   This constraint is incorporated as:

$$F_3 \leq EENSC_{max} \tag{6.12}$$

   where F_3 represents the objective function EENS cost and $EENSC_{max}$ is the user-defined upper limit for solution's EENS cost.

7. Power output limits on wind generation system: The power output function with respect to the wind speed is given by [182]

$$P_w^t = \begin{cases} 0 & v_w^t \leq v_{ci} \text{ or } v_{ci} \leq v_w^t \\ P_{wn}\left(\frac{v_w^t - v_{ci}}{v_r - v_{ci}}\right), & v_{ci} \leq v_w^t \leq v_r \\ P_{wn} & v_r \leq v_w^t \leq v_{co} \end{cases} \tag{6.13}$$

where v_w^t is the forecast wind speed at hour t; v_{ci}, v_{co} and v_r are the cut-in, cut-out and rated wind turbine speed respectively; and P_{wn} is the equivalent rated power output for wind power generation.

In this Chapter, the MOWT-UC problem has been formulated by extending the MOEER-UC problem formulation considered in Chapter 5. The modifications made in the problem formulation considered in this Chapter as compared to the problem formulation considered in Chapter 5 are clearly highlighted below.

- The constraint related to system power balance is modified as the system now comprises of thermal generators as well as wind generators. Thus, at each hour the power output of the combined wind-thermal system should be equal to the load demand.
- The constraint related to power output limits on wind generation system is added as the power output of the wind generation system depends upon the hourly wind speed.

Thus, it is noted that the MOWT-UC problem is an extension of the MOEER-UC problem which was comprehensively solved in Chapter 5. Further, it is noted that the wind farm investment cost is not taken into account in this study.

6.5 Procedure for Calculation of EENS cost (i.e., Reliability objective)

The system reliability evaluation i.e., EENS cost evaluation procedure is modified in this Chapter (as compared to Chapter 5) as the uncertainty occurring due to error in wind forecast is also incorporated. The procedure for calculation of the EENS reliability index in presence of various uncertainties is described as follows.

6.5.1 Incorporating Uncertainty due to Thermal Unit Outage

Each thermal unit is considered as a two-state model, according to which a unit is either available or unavailable for generation. According to this model, the unavailability of the unit i during a short time interval LT (known as the system lead time) is given by

$$U_i(LT) = 1 - e^{\lambda_i LT} \tag{6.14}$$

where λ_i is the failure rate of unit i [162]. The probability $U_i(LT)$ given by (6.14) is known as the outage replacement rate (ORR) of the unit, i.e., the probability of losing capacity and not being able to replace it.

To calculate the EENS index for every chromosome, the conventional "loss of load" method is used, except that ORR is used instead of FOR (Forced Outage Rate) [162]. This method is based on the creation of the capacity outage probability table (COPT) according to the given load curve [163]. A COPT is formed for every hour using the ORR of all the committed units. The creation of COPT is based on the unit addition algorithm [163]. A COPT may be visualized as a table with n rows ($j = 1, 2, \ldots n$) and 3 columns. The first column represents n different generation levels that may be outaged. The second and third column represents the probability PR_j and the total capacity CR_j that remains in service corresponding to each outage level respectively. The reliability index $EENS_t$ (i.e., expected energy not served for each hour t) is calculated as follows:

$$EENS_t = \sum_{j=1}^{n} PR_j.LOSS_j.\,(L^t - CR_j), \quad t \in [1, T_{max}] \tag{6.15}$$

where $LOSS_j$ is given by

$$LOSS_j = \begin{cases} 1, & if\ CR_j < L^t \\ 0, & otherwise \end{cases} \tag{6.16}$$

The EENS index of the entire scheduling horizon, $EENS_{tot}$ is given by

$$EENS_{tot} = \sum_{t=1}^{T_{max}} EENS_t \tag{6.17}$$

6.5.2 Incorporating Uncertainty due to Thermal Unit Outage, Load Forecast Error and Wind Forecast Error

The procedure presented in the above Section represents the method to evaluate $EENS_{tot}$ if the uncertainty due to only thermal unit outage is considered and the uncertainty due to load forecast and wind forecast error is neglected. However, load forecast and wind forecast are generally associated with uncertainty and hence should be considered in the wind-thermal UC problem. It is an accepted practice to assume that the forecast load consists of actual load plus a normally distributed error [163]. The standard deviation (σ^t_{load}) of the load forecast error is equal to a percentage SL of the expected demand and depends upon the accuracy of the forecasting tool.

$$\sigma^t_{load} = SL \times L^t \tag{6.18}$$

The forecast of wind power production can also be assumed to be equal to the actual value plus a normally distributed error. The standard deviation (σ_{wind}^t) of the wind forecast error is approximated by [172]

$$\sigma_{wind}^t = \frac{1}{5} P_w^t + \frac{1}{50} P_{wn} \tag{6.19}$$

The net forecast demand is the difference between the forecast load and the forecast wind power. It is assumed that the errors in load forecast and wind power forecast are uncorrelated Gaussian stochastic variables [28, 172] and the standard deviation (σ_d^t) of the error on net forecast demand is

$$\sigma_d^t = \sqrt{(\sigma_{load}^t)^2 + (\sigma_{wind}^t)^2} \tag{6.20}$$

The distribution representing the net forecast demand can be divided into a discrete number of class intervals with the distribution mean being the net forecast demand and standard deviation given by (6.20). The load representing the class interval mid-point is assigned the designated probability for that class interval. It is recommended in [172] that a seven-step approximation (0, $\pm 1\sigma$, $\pm 2\sigma$, $\pm 3\sigma$) to the normal distribution (known as seven-step model) is adequate to represent the uncertainty in net demand forecast. Thus, with the assumption of net demand forecast uncertainty to be normally distributed and represented by the seven-step model, the EENS index calculation for each hour t is given by

$$EENS_t = \sum_{m=1}^{7} (EENS_t(m) PL(m)), \quad t \in [1, T_{max}] \tag{6.21}$$

where $PL(m)$ and $EENS_t(m)$ indicate the probability and EENS value for hour t associated with the discrete class interval m in the seven-step model, respectively.

It is noted that $EENS_t(m)$ is calculated using (6.15) and (6.16) by simply replacing L^t in the two equations by $L^t(m)$ where $L^t(m)$ represents the load for hour t associated with the discrete class interval m in the seven-step model.

Once $EENS_t$ for every hour t is evaluated using (6.21), the EENS index of the entire scheduling horizon, $EENS_{tot}$ (as discussed above as well) is given by

$$EENS_{tot} = \sum_{t=1}^{T_{max}} EENS_t \tag{6.22}$$

6.6 Proposed Algorithm

In the Section 6.4, it is highlighted that there are only two modifications which need to be made in the problem formulation of MOEER-UC (considered in Chapter 5) to formulate the MOWT-UC problem. Thus, the MOWT-UC problem is an extension of the MOEER-UC problem which was comprehensively solved in Chapter 5. Hence, as the characteristics of the two problems (i.e., MOEER-UC problem and MOWT-UC problem) are the same, MOEA/D-DE and its variants proposed in Chapter 5 i.e., MOEA/D-DE with non-uniform weight vector distribution and MOEA/D-DE with ϵ-dominance based external archive are applied to the MOWT-UC problem.

Since, the basic framework of MOEA/D-DE i.e., chromosome representation, crossover operators, mutation operators, constraint handling, etc. remains the same as presented in Chapter 5; to avoid repetition, these are not presented again. Thus, only the pseudo-code of MOEA/D-DE for the MOWT-UC problem is presented as follows.

6.6.1 Steps of the Proposed Algorithm MOEA/D-DE

Input

- NP: the number of subproblems considered in MOEA/D-DE i.e., the population size;
- $\lambda_1, \lambda_2, ..., \lambda_{NP}$: a set of NP weight vectors;
- T: the neighborhood size;
- δ: the probability that parent solutions are selected from the neighborhood;
- n_r: the maximal number of solutions that can be replaced by each child solution.
- z: the initial reference point $(z_1, z_2, z_3) = (10^{30}, 10^{30}, 10^{30})$. The reference point initially has very large dimensions and is updated during the evolution of population.

At each generation, MOEA/D-DE maintains the following:

- A population of NP solutions $x_1, x_2, \ldots, x_{NP}$, where x_i is the current solution to the i_{th} subproblem.
- $F(x_1), F(x_2), \ldots, F(x_{NP})$, where $F(x_i) = \{F_1(x_i), F_2(x_i), F_3(x_i)\} \forall i = 1, 2, \ldots, NP$.
- $CV(x_i)$ = total constraint violation of x_i $\forall i = 1, 2, \ldots, NP$.
- $z = (z_1, z_2, z_3)$, where z_1, z_2 and z_3 are the best values found so far for objective F_1, F_2 and F_3, respectively.

The steps executed are as follows.

- **Step 1: Calculation of net forecast load on thermal generators**

  Compute the net forecast load on thermal generators for each hour by taking difference between the forecast load demand and the forecast wind power corresponding to each hour.

- **Step 2: Initialization**
  - **Step 2.1** Compute the Euclidean distances between any two weight vectors and then calculate T closest weight vector to each λ_i. For all $i = 1, 2, \ldots, NP$, set $B(i) = \{i_1, i_2, \ldots, i_T\}$, where λ_j, $\forall j \in B(i)$ are T closest vectors to λ_i.
  - **Step 2.2** Generate initial population randomly.

- **Step 2.3** For all $i = 1, 2, \ldots, NP$, repair x_i for load demand equality constraint violation.
- **Step 2.4** Calculate $CV(x_i)$ and $F(x_i)$ i.e., $\{F_1(x_i), F_2(x_i), F_3(x_i)\}$.
- **Step 2.5** Update $z = (z_1, z_2, z_3)$ according to the condition: $z_j = \min_{1 \leq i \leq NP} F_j(x_i)$ if x_i is feasible.

- **Step 3: Update**
  For $i = 1, 2, \ldots, NP$, **do**

  - **Step 3.1 Selection of Mating/Update Range:** Uniformly generate random number $rand$ from [0,1]. Then,

$$P = \begin{cases} B(i), & \text{if } rand < \delta \\ 1, 2, \ldots, NP, & \text{otherwise} \end{cases}$$

  - **Step 3.2 Reproduction:**
    1. Randomly select three mutually exclusive indices r_1, r_2 and r_3 from P which are different from i.
    2. Decode x_k in UCM_k and RPM_k, where $k = i, r_1, r_2, r_3$.
    3. Generate a solution UCM_{child} using GA recombination operators on UCM_k, where $k = i, r_1$.
    4. Generate a solution RPM_{child} using DE recombination operators on RPM_k, where $k = r_1, r_2, r_3$.
    5. Encode UCM_{child} and RPM_{child} in x_{child}.
  - **Step 3.3 Repair:** Repair x_{child} for boundary constraint violation and load demand equality constraint violation.
  - **Step 3.4:** Calculate $CV(x_{child})$ and $F(x_{child})$ i.e., $\{F_1(x_{child}), F_2(x_{child}), F_3(x_{child})\}$.
  - **Step 3.5 Update of z:** For $j = 1, 2$ do
    1. If x_{child} is feasible and $z_j > F_j(x_{child})$ then set $z_j = F_j(x_{child})$
  - **Step 3.6 Replacement/Update of Solutions:** Set $c = 0$ and then do
    1. Set $flag = 0$.
    2. If $c = n_r$ or P is empty, $i = i + 1$ and go to **Step 3.1**, else randomly pick an index j from P.
    3. Determine if x_{child} replaces x_j or not according to the replacement rules.
    4. If x_{child} replaces x_j then $flag = 1$ and $c = c + 1$.
    5. If $flag = 1$, remove j from P and go to **Step 3.6.1**.

- **Step 4: Stopping Criteria**
  If termination criterion is satisfied, then stop else go to **Step 3**.

Output

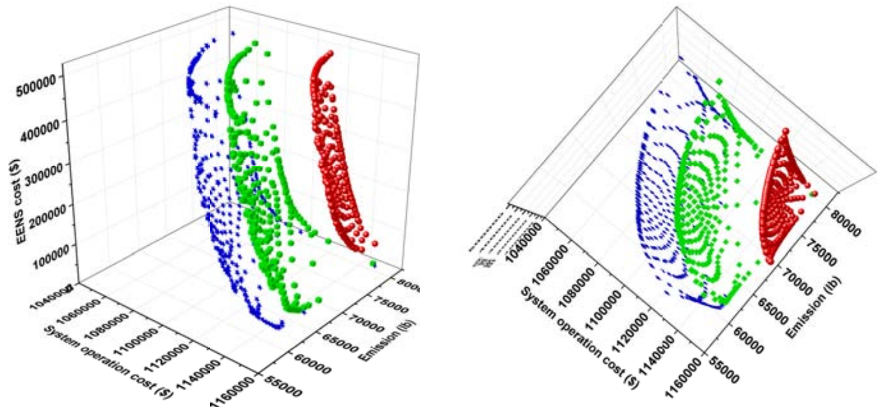
- Approximation to Pareto-optimal solutions: $\{x_1, x_2, \ldots, x_{NP}\}$.
- Approximation to Pareto-optimal front: $\{F(x_1), F(x_2), ..., F(x_{NP})\}$.

6.7 Experimental Study

In this Section, MOEA/D-DE and its variants are tested on the MOWT-UC problem for power system with 20 thermal units in a 24 hour scheduling horizon [139]. The lead time of the system is fixed as 4 hours [162] while the standard deviation (σ_{load}^t) of the load forecast error is assumed to be 5% of the hourly load demand as suggested in [164]. Further, VOLL is assumed to be 5000 $\$/MWh$ [151]. The wind turbine characteristics data is taken from [182] while the forecast wind velocity data is taken from [183].

At first, the effect of wind integration was analyzed by executing MOEA-/D-DE on the 20 unit system without any wind penetration and with 10% and 15% wind penetration. Fig. 6.1a and 6.1b illustrate the side view and top view for the distribution of the final non-dominated solutions found by MOEA/D-DE in different wind penetration scenarios. It is observed from the figures that with the wind penetration, the system operation cost and emission reduce considerably. However, it is noted that in this work, the wind farm investment cost is not taken into account (as mentioned above).

Next, MOEA/D-DE and its variants namely MOEA/D-DE/NUWD and MOEA/D-DE with ϵ-dominance based external archive were implemented on the 20 unit system considering 10% and 15% wind penetration. Fig. 6.2a, 6.2b and 6.2c show the distribution of the final non-dominated solutions found by MOEA/D-DE, MOEA/D-DE/NUWD and MOEA/D-DE with ϵ-dominance based external archive on 20 unit system with 10% wind penetration, respectively. Similarly, Fig. 6.3a, 6.3b and 6.3c show the

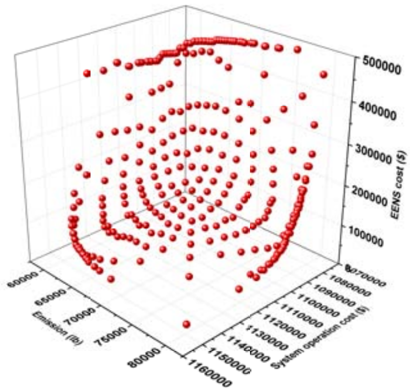


(a) Side view (b) Top view

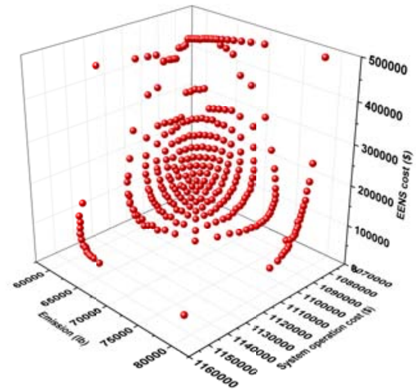
Fig. 6.1 Non-dominated solutions obtained by MOEA/D-DE in presence of (only) thermal generation (represented by red balls), 10% wind penetration (represented by green cubes) and 15% wind penetration (represented by blue stars) on the 20 unit system.

distribution of the final non-dominated solutions found by MOEA/D-DE, MOEA/D-DE/NUWD and MOEA/D-DE with ϵ-dominance based external archive on 20 unit system with 15% wind penetration, respectively.

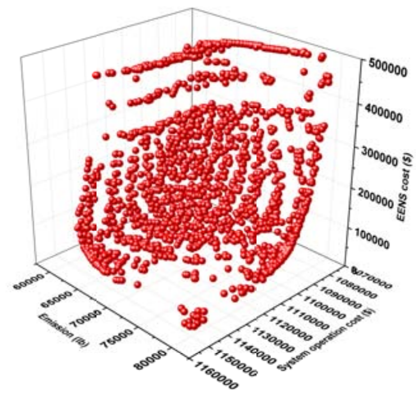
It is noted that in the case of MOWT-UC problem as well, the behavior of MOEA/D-DE and its variants is the same as was observed in the case of MOEER-UC problem in Chapter 5. The clustering of solutions obtained by MOEA/D-DE along the boundary of the constrained objective space (see Fig. 6.2a and 6.3a) is a limitation as it reduces the number of solutions obtained in the middle of the trade-off surface. However, MOEA/D-DE/NUWD is able to find a better distribution of solutions towards the middle of the trade-off surface (see Fig. 6.2b and 6.3b). Furthermore, MOEA/D-DE with external archive is able to provide significantly better distribution of solutions throughout the trade-off surface as compared to that obtained by MOEA/D-DE and MOEA/D-DE/NUWD (see Fig. 6.2c and 6.3c).



(a) MOEA/D-DE



(b) MOEA/D-DE/NUWD

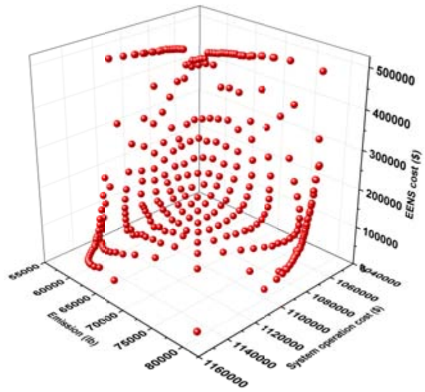


(c) MOEA/D-DE with external archive

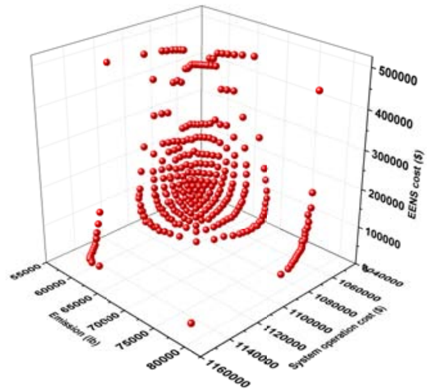
Fig. 6.2 The distribution of the final solutions obtained by MOEA/D-DE, MOEA/D-DE/NUWD and MOEA/D-DE with external archive for 10% wind penetration on the 20 unit system.

6.8 Summary

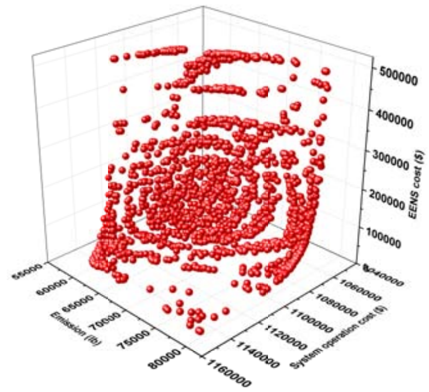
In this Chapter, the three-objective unit commitment problem presented in Chapter 5 was extended and significant wind penetration was efficiently included. The multiple-objectives of the wind-thermal UC problem remained the same i.e., system operation cost, emission and reliability. MOEA/D-DE and its variants proposed in Chapter 5 for the MOEER-UC problem were applied to the MOWT-UC problem considering 10% and 15% wind penetration. The results obtained demonstrated that MOEA/D-DE with ϵ-dominance based external archive presents uniformly distributed trade-off



(a) MOEA/D-DE



(b) MOEA/D-DE/NUWD



(c) MOEA/D-DE with external archive

Fig. 6.3 The distribution of the final solutions obtained by MOEA/D-DE, MOEA/D-DE/NUWD and MOEA/D-DE with external archive for 15% wind penetration on the 20 unit system.

optimal solutions for the MOWT-UC problem as well.

Chapter 7

Conclusions and Future Work

This Chapter summarizes the conclusions and the main contributions of the research work reported in this thesis and outlines directions for future work. Section 7.1 presents the conclusions of the thesis while Section 7.2 details the main contributions of thesis. Finally, Section 7.3 presents recommendations for future work.

7.1 Conclusions

The unit commitment (UC) is one of the most important problems in power system scheduling. The UC problem is mostly solved in the literature considering system operation cost as the single (economic) objective and the emission as well as the reliability aspects are generally neglected. The primary aim of the thesis was to consider emission and reliability as objectives along with system operation cost and solve the UC problem as a multi-objective optimization problem using evolutionary algorithms (EAs).

In Chapter 3, the UC problem in deterministic environment involving system operation cost as the single objective was tackled. An evolutionary optimization skeleton based on problem-specific: chromosome representation, genetic operators, and knowledge was developed. Further, a novel hybrid framework (termed hGADE) based on combination of genetic algorithm and differential evolution was proposed to solve the UC problem. This Chapter demonstrated the flexibility of the hGADE framework by hybridizing GA with 4 classical and 2 state-of-the-art self-adaptive DE variants. Further, the efficacy of the hGADE variants was highlighted by extensively comparing against a GA based approach. The best hGADE variants i.e., hGADE/current-to-rand/1, hGADE/JADE and hGADE/rand/1 were benchmarked and found to be efficient in achieving superior average cost and best cost solution.

In Chapter 4, the single-objective UC problem solved in Chapter 3 was extended to bi-objective UC problem and minimizing emission was considered as an additional objective along with minimizing system operation cost. The optimization skeleton developed for the single-objective UC problem in Chapter 3 (i.e., problem-specific: chromosome representation, genetic operators and knowledge) was efficiently embedded within the domination and decomposition based multi-objective optimization frameworks. Non-dominated sorting genetic algorithm II (NSGA-II) and multi-objective evolutionary algorithms based on decomposition (MOEA/D-SBX and MOEA/D-DE) were selected as the representative algorithms from the domination and decomposition frameworks, respectively. The algorithms were efficiently customized and applied to the multi-objective economic/emission unit commitment (MOEE-UC) problem. Further, the hybrid strategy between GA and DE, which was found to perform well in Chapter 3 on the single-objective UC problem was incorporated within MOEA/D-DE. The proposed MOEAs i.e., NSGA-II-SBX, MOEA/D-SBX and MOEA/D-DE were exhaustively compared among themselves on the MOEE-UC problem and MOEA/D-DE was found to significantly outperform the contender algorithms. Thereafter, MOEA/D-DE was benchmarked with the algorithms presented in the literature. It was observed that MOEA/D-DE was able to outperform the benchmark algorithms in terms of convergence and distribution throughout the trade-off front except at the extremes. Therefore, a novel non-uniform weight vector distribution (NUWD) strategy was proposed within the framework of MOEA/D-DE to bias the search direction of the algorithm towards the extremes. Although, MOEA/D-DE with the proposed NUWD strategy i.e., MOEA/D-DE/NUWD was able to capture the extremes better than MOEA/D-DE, a slight compromise was observed towards the middle of the trade-off front. Since, MOEA/D-DE and MOEA/D-DE/NUWD were found to be complementary to each other, an ensemble optimizer based on combination of MOEA/D-DE with uniform and non-uniform weight vector distribution strategy was proposed. The ensemble optimizer was found to enhance the overall performance of the algorithm. Moreover, the ensemble optimizer significantly outperformed the benchmark algorithms in obtaining better converged and uniformly distributed trade-off optimal solutions.

In Chapter 5, the bi-objective economic/emission UC problem (in deterministic environment) solved in Chapter 4 was extended to three-objective UC problem in uncertain environment and maximizing reliability was considered as an additional objective along with minimizing system operation cost and minimizing emission. The uncertainties occurring due to thermal generator outage and load forecast error were captured using expected energy not served (EENS) reliability index and EENS cost was used to reflect the reliability objective. The MOEAs developed for the bi-objective UC problem in Chapter 4 i.e., NSGA-II-SBX, MOEA/D-SBX and MOEA/D-DE were applied in this Chapter to solve the three-objective UC problem. The proposed MOEAs were exhaustively compared among themselves and

MOEA/D-DE was found to significantly outperform NSGA-II-SBX and MOEA/D-SBX. However, a limitation was observed in the performance of MOEA/D-DE in the sense that many solutions were found to be clustered towards the edges of the trade-off surface. Since, the non-uniform weight vector distribution (NUWD) strategy was found to efficiently bias the search direction of MOEA/D-DE on the bi-objective optimization problem in Chapter 4; a new NUWD strategy was proposed for the three-objective optimization problem in this Chapter. The proposed NUWD strategy was found to improve the performance of the algorithm in terms of obtaining better distribution of solutions towards the middle of the trade-off surface. However, more than desired solutions were still found to be clustered towards the edges of the trade-off surface. Thus, MOEA/D-DE with an ϵ-dominance based external archive was presented to overcome this limitation. MOEA/D-DE with external archive was found to obtain significantly better distributed solutions corresponding to the three-objective UC problem as compared to MOEA/D-DE and MOEA/D-DE with NUWD strategy.

In Chapter 6, the three-objective UC problem (in uncertain environment) solved in Chapter 5 was further extended to include significant wind penetration. The additional uncertainty due to wind forecast error was captured along with uncertainty due to thermal generator outage and load forecast error using expected energy not served (EENS) reliability index. The multiple objectives considered remained the same as that in Chapter 5 i.e., minimizing system operation cost, minimizing emission and maximizing reliability. In Chapter 5, MOEA/D-DE was established as the best MOEA among the proposed MOEAs for solving the three-objective UC problem. Since, the characteristics of the two problems i.e., problem solved in Chapter 5 and the problem considered in this Chapter remained the same because of the three-objective optimization model in both the problems; in this Chapter only MOEA/D-DE was implemented. Further, the variants of MOEA/D-DE i.e., MOEA/D-DE based on non-uniform weight vector distribution strategy and MOEA/D-DE with ϵ-dominance based external archive were also implemented to solve the three-objective wind-thermal UC problem. The experimental results revealed that MOEA/D-DE with ϵ-dominance based external archive was able to return uniformly distributed trade-off optimal solutions for the wind-thermal UC problem.

7.2 Main Contributions

The main contributions made with respect to different problems tackled in the thesis are summarized as follows:

7.2.1 Contributions related to single-objective UC problem

1. A novel framework based on hybrid of GA and DE such that GA explores the binary search space while DE explores the continuous search space, was developed for solving the single-objective UC problem. The contributions related to the proposed hGADE framework are further categorized as follows:

   (a) A total of 6 hybrid GA-DE variants were developed by integrating GA with - 1) 4 classical versions of DE algorithm namely, DE/rand/1, DE/rand/2, DE/current-to-rand/1 and DE-current-to-rand/2 [68, 74]; and 2) 2 self-adaptive versions of the DE algorithm, namely jDE [69] and JADE [71] and successfully tested on the UC problem.

(b) It was observed that the system operator can make remarkable cost savings by adopting the proposed hGADE algorithm (specifically the variants hGADE/current-to-rand/1, hGADE/-JADE and hGADE/rand/1) for solving the UC problem.

7.2.2 Contributions related to bi-objective UC problem considering deterministic environment

1. To the best of our knowledge, a first attempt was made to propose a MOEA/D-DE for solving the bi-objective UC problem considering economic and emission objectives. The contributions related to the proposed MOEA/D-DE are further categorized as follows:

   (a) The hybridization strategy between GA and DE algorithm was embedded within the framework of MOEA/D-DE such that GA explores the binary search space while DE explores the continuous search space. The proposed MOEA/D-DE was found to significantly outperform NSGA-II-SBX and MOEA/D-SBX on the bi-objective UC problem.

   (b) A novel non-uniform weight vector distribution strategy was proposed within the framework of MOEA/D-DE to bias the search direction of the algorithm towards the extremes of the trade-off front.

   (c) An ensemble optimizer based on combination of MOEA/D-DE with uniform and non-uniform weight vector distribution strategy was proposed. The ensemble optimizer, termed Enhanced-MOEA/D-DE, was found to present significantly better converged and distributed trade-off solutions than the algorithms proposed in the literature.

7.2.3 Contributions related to three-objective UC problem considering uncertain environment

1. To the best of our knowledge, a first attempt was made to propose a MOEA/D-DE for solving the three-objective UC problem considering economic, emission and reliability objectives. The contributions related to the proposed MOEA/D-DE are categorized as follows:

   (a) The proposed MOEA/D-DE based on hybrid strategy between GA and DE algorithm was found to significantly outperform NSGA-II-SBX and MOEA/D-SBX on the three-objective UC problem.

   (b) A novel non-uniform weight vector distribution strategy was proposed within the framework of MOEA/D-DE to bias the search direction of the algorithm and obtain better distribution of solutions towards the middle of the trade-off surface.

   (c) MOEA/D-DE with an ϵ-dominance based external archive was proposed for solving the three-objective UC problem. It was demonstrated that MOEA/D-DE with ϵ-dominance based external archive obtains much better uniformly distributed set of trade-off solutions than MOEA/D-DE without archive on the three-objective UC problem.

7.2.4 Contributions related to three-objective wind-thermal UC problem considering uncertain environment

1. To the best of our knowledge, a first attempt was made to propose a MOEA/D-DE for solving the three-objective UC problem in pres-

ence of significant wind penetration considering economic, emission and reliability objectives. MOEA/D-DE with an ϵ-dominance based external archive (which was found to be the best algorithm for three-objective UC problem), was proposed to solve the three-objective wind-thermal UC problem considering economic, emission and reliability objectives. The algorithm was found to obtain a uniformly distributed set of trade-off solutions in the archive for the three-objective wind-thermal UC problem.

7.3 Future Work

7.3.1 Further study related to extension of UC problem

- The power system scheduling problems can be made more realistic by adding the network security constraints. Thus, the proposed algorithms in the thesis can be extended to solve security-constrained unit commitment (SCUC) problem [184–188].

7.3.2 Further study related to hGADE algorithm

- The proposed hybridization strategy between GA and DE in the hGADE framework can be tested on other challenging real-world mixed-integer optimization problems.

- The proposed hGADE framework is generic and depending upon the problem requirements or choice, the user may easily integrate other discrete and/or real parameter operators in the framework for solving challenging single-objective mixed-integer optimization problems.

Thus, for example, hybridizing GA with other well known optimizers like PSO [64], EDA [66], etc for solving mixed-integer optimization problems and comparing the performance of different hybrid EAs can be a very interesting study.

7.3.3 Further study related to MOEA/D-DE algorithm

- Similarly to the hGADE algorithm, the proposed hybrid MOEA/D-DE (presented in Chapter 4, 5 and 6) based on combining the strengths of GA and DE is a generic algorithm which can be tested on other challenging multi-objective mixed integer optimization problems.

- The proposed non-uniform weight vector distribution strategy for bi-objective optimization problem (as in Chapter 4) and for three-objective optimization problem (as in Chapter 5) is a generic algorithmic component and can be integrated within the framework of MOEA/D to bias the search direction of MOEA/D and tested on other problems. Moreover, other non-uniform weight vector distribution strategies can also be investigated within the MOEA/D framework.

- An ensemble optimizer (termed Enh-MOEA/D-DE) based on combination of MOEA/D-DE with uniform and non-uniform weight vector distribution strategy was proposed in Chapter 4 for solving the bi-objective economic/emission UC problem. However, in the ensemble optimizer, there was no migration (i.e., communication) between the component MOEAs. Thus, a parallel island model [189] based on integration of MOEA/D-DE with uniform and non-uniform weight-vector distribution strategy, which involves communication (i.e., migration) between the component MOEAs, can be proposed to enhance the performance of Enh-MOEA/D-DE.

- In Chapter 5 and 6, it was observed that in the trade-off surface obtained using MOEA/D-DE for the three-objective UC problems, many solutions were clustered at the edges of the surface. The reason (as was mentioned in the Chapter as well) may be that there is no explicit diversity maintenance operator in the framework of MOEA/D. Recently, Gee *et al.*[190] proposed an online diversity assessment technique in evolutionary multi-objective optimization. According to this technique, the diversity of the population can be evaluated online i.e., during the search process. This technique can also measure the diversity loss caused by any individual in the population and the algorithm can then perform a diversity-preservation selection based on this information. In [190], this technique was incorporated within the MOEA/D and the technique was demonstrated to enhance the diversification of the solution set obtained by the algorithm.

  Thus, an interesting future work can be to implement MOEA/D-DE with such online diversity assessment technique and diversity-preservation based selection so as to investigate if it results in diversification of the trade-off solutions obtained by MOEA/D-DE on the three-objective UC problem.

7.3.4 Further study related to application of other MOEAs

- In this thesis, NSGA-II-SBX, MOEA/D-SBX and MOEA/D-DE were the three MOEAs which were applied to different multi-objective UC problems. However, the three MOEAs used the same optimization skeleton (i.e., problem-specific: chromosome representation, genetic operators and knowledge). Thus, an interesting future study can be to embed the optimization skeleton within other popular MOEAs like indicator based MOEA - HypE [191], etc and compare the performance of different MOEAs on the multi-objective UC problems.

References

[1] A. J. Wood and B. Woolenberg, *Power Generation Operation and Control*. New York: Wiley, 1996.

[2] J. J. Shaw, "A direct method for security-constrained unit commitment," *Power Systems, IEEE Transactions on*, vol. 10, no. 3, pp. 1329–1342, Aug 1995.

[3] D. Sharma, A. Trivedi, D. Srinivasan, and L. Thillainathan, "Multi-agent modeling for solving profit based unit commitment problem," *Applied Soft Computing*, vol. 13, no. 8, pp. 3751 – 61, 2013.

[4] T. Senjyu, K. Shimabukuro, K. Uezato, and T. Funabashi, "A fast technique for unit commitment problem by extended priority list," *IEEE Transactions on Power Systems*, vol. 18, no. 2, pp. 882 – 888, 2003.

[5] S. Patra, S. Goswami, and B. Goswami, "Fuzzy and simulated annealing based dynamic programming for the unit commitment problem," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5081 – 6, 2009.

[6] J. A. Muckstadt and R. C. Wilson, "An application of mixed-integer programming duality to scheduling thermal generating systems," *IEEE Transactions on Power Apparatus and Systems*, vol. Pas-87, no. 72, pp. 1968 – 78, 1968.

[7] A. I. Cohen and M. Yoshimura, "A branch and bound algorithm for unit commitment," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-102, no. 2, pp. 444 – 51, 1983.

[8] W. Ongsakul and N. Petcharaks, "Unit commitment by enhanced adaptive Lagrangian relaxation," *IEEE Transactions on Power Systems*, vol. 19, no. 1, pp. 620 – 8, 2004.

[9] S. A. Kazarlis, A. G. Bakirtzis, and V. Petridis, "A genetic algorithm solution to the unit commitment problem," *IEEE Transactions on Power Systems*, vol. 11, no. 1, pp. 83 – 92, 1996.

[10] D. Datta, "Unit commitment problem with ramp rate constraint using a binary-real-coded genetic algorithm," *Applied Soft Computing*, vol. 13, no. 9, pp. 3873 – 83, 2013.

[11] K. A. Juste, H. Kita, E. Tanaka, and J. Hasegawa, "An evolutionary programming solution to the unit commitment problem," *IEEE Transactions on Power Systems*, vol. 14, no. 4, pp. 1452 – 9, 1999.

[12] J. Valenzuela and A. E. Smith, "A seeded memetic algorithm for large unit commitment problems," *Journal of Heuristics*, vol. 8, no. 2, pp. 173 – 95, 2002.

[13] T. O. Ting, M. V. C. Rao, and C. K. Loo, "A novel approach for unit commitment problem via an effective hybrid particle swarm optimization," *IEEE Transactions on Power Systems*, vol. 21, no. 1, pp. 411 – 18, 2006.

[14] X. Yuan, A. Su, H. Nie, Y. Yuan, and L. Wang, "Unit commitment problem using enhanced particle swarm optimization algorithm," *Soft Computing*, vol. 15, no. 1, pp. 139 – 48, 2011.

[15] D. N. Simopoulos, S. D. Kavatza, and C. D. Vournas, "Unit commitment by an enhanced simulated annealing algorithm," *IEEE Transactions on Power Systems*, vol. 21, no. 1, pp. 68 – 76, 2006.

[16] T. W. Lau, C. Y. Chung, K. P. Wong, T. S. Chung, and S. L. Ho, "Quantum-inspired evolutionary algorithm approach for unit commitment," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1503 – 12, 2009.

[17] D. Datta and S. Dutta, "A binary-real-coded differential evolution for unit commitment problem," *International Journal of Electrical Power & Energy Systems*, vol. 42, no. 1, pp. 517 – 24, 2012.

[18] K. Chandrasekaran, S. Hemamalini, S. P. Simon, and N. P. Padhy, "Thermal unit commitment using binary/real coded artificial bee colony algorithm," *Electric Power Systems Research*, vol. 84, no. 1, pp. 109 – 19, 2012.

[19] K. Chandrasekaran, S. P. Simon, and N. P. Padhy, "Binary real coded firefly algorithm for solving unit commitment problem," *Information Sciences*, vol. 249, pp. 67 – 84, 2013.

[20] X. Yuan, B. Ji, S. Zhang, H. Tian, and Y. Hou, "A new approach for unit commitment problem via binary gravitational search algorithm," *Applied Soft Computing Journal*, vol. 22, pp. 249 – 260, 2014.

[21] A. Trivedi, N. M. Pindoriya, and D. Srinivasan, "Modified NSGA-II for day-ahead multi-objective thermal generation scheduling," in *2010 9th International Power and Energy Conference (IPEC 2010)*, Piscataway, NJ, USA, 2010, pp. 752 – 7.

[22] A. Trivedi, N. M. Pindoriya, D. Srinivasan, and D. Sharma, "Improved multi-objective evolutionary algorithm for day-ahead thermal generation scheduling," in *2011 IEEE Congress of Evolutionary Computation, CEC 2011*, New Orleans, LA, United states, 2011, pp. 2170 – 2177.

[23] A. Trivedi, D. Srinivasan, D. Sharma, and C. Singh, "Evolutionary multi-objective day-ahead thermal generation scheduling in uncertain environment," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1345 – 1354, 2013.

[24] D. Li, S. Das, A. Pahwa, and K. Deb, "A multi-objective evolutionary approach for generator scheduling," *Expert Systems with Applications*, vol. 40, no. 18, pp. 7647 – 7655, 2013.

[25] P. A. Ruiz, C. R. Philbrick, E. Zak, K. W. Cheung, and P. W. Sauer, "Uncertainty management in the unit commitment problem," *Power Systems, IEEE Transactions on*, vol. 24, no. 2, pp. 642–651, May 2009.

[26] C.-L. Chen, "Optimal wind thermal generating unit commitment," *Energy Conversion, IEEE Transactions on*, vol. 23, no. 1, pp. 273–280, March 2008.

[27] J. Wang, M. Shahidehpour, and Z. Li, "Security-constrained unit commitment with volatile wind power generation," *Power Systems, IEEE Transactions on*, vol. 23, no. 3, pp. 1319–1327, Aug 2008.

[28] R. Doherty and M. O'Malley, "A new approach to quantify reserve demand in systems with significant installed wind capacity," *Power Systems, IEEE Transactions on*, vol. 20, no. 2, pp. 587–595, May 2005.

[29] K. Deb, *Multi-objective optimzation using evolutionary algorithms.* Singapore: John Wiley & Sons, 2001.

[30] K. C. Tan, E. F. Khor, and T. H. Lee, *Multiobjective Evolutionary Algorithms and Applications*, ser. Advanced Information and Knowledge Processing. Springer, 2005.

[31] K. P. Dahal, K. C. Tan, and P. I. Cowling, Eds., *Evolutionary Scheduling*, ser. Studies in Computational Intelligence. Springer, 2007, vol. 49.

[32] P. P. Bonissone, R. Subbu, N. Eklund, and T. R. Kiehl, "Evolutionary algorithms + domain knowledge = real-world evolutionary computation," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 256 – 80, 2006.

[33] K. S. Hindi, H. Yang, and K. Fleszar, "An evolutionary algorithm for resource-constrained project scheduling," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 5, pp. 512–518, Oct 2002.

[34] L. Minku, D. Sudholt, and X. Yao, "Improved evolutionary algorithm design for the project scheduling problem based on runtime analysis," *Software Engineering, IEEE Transactions on*, vol. 40, no. 1, pp. 83–102, Jan 2014.

[35] K. Dahal, G. M. Burt, J. NcDonald, and A. Moyes, "A case study of scheduling storage tanks using a hybrid genetic algorithm," *Evolutionary Computation, IEEE Transactions on*, vol. 5, no. 3, pp. 283–294, Jun 2001.

[36] L. Tang, Y. Zhao, and J. Liu, "An improved differential evolution algorithm for practical dynamic scheduling in steelmaking-continuous casting production," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 2, pp. 209–225, April 2014.

[37] J. Zhong, M. Shen, J. Zhang, H.-H. Chung, Y. Shi, and Y. Li, "A differential evolution algorithm with dual populations for solving periodic railway timetable scheduling problem," *Evolutionary Computation, IEEE Transactions on*, vol. 17, no. 4, pp. 512–527, Aug 2013.

[38] Y. Ping and J. Minghai, "A chaotic particle swarm optimization algorithm for the jobshop scheduling problem," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*, July 2014, pp. 218–222.

[39] M. Dorigo and L. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *Evolutionary Computation, IEEE Transactions on*, vol. 1, no. 1, pp. 53–66, Apr 1997.

[40] H.-K. Tsai, J.-M. Yang, Y.-F. Tsai, and C.-Y. Kao, "An evolutionary algorithm for large traveling salesman problems," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 34, no. 4, pp. 1718–1729, Aug 2004.

[41] C. K. Goh, E. J. Teoh, and K. C. Tan, "A hybrid evolutionary approach for heterogeneous multiprocessor scheduling," *Soft Computing*, vol. 13, no. 8-9, pp. 833 – 846, 2009.

[42] M. Marichelvam, T. Prabaharan, and X. S. Yang, "A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 2, pp. 301–305, April 2014.

[43] N. B. Ho and J. C. Tay, "Solving multiple-objective flexible job shop problems by evolution and local search," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 38, no. 5, pp. 674–685, Sept 2008.

[44] S. Nguyen, M. Zhang, M. Johnston, and K. C. Tan, "Automatic design of scheduling policies for dynamic multi-objective job shop scheduling via cooperative coevolution genetic programming," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 2, pp. 193–208, April 2014.

[45] S. Salinas, M. Li, and P. Li, "Multi-objective optimal energy consumption scheduling in smart grids," *Smart Grid, IEEE Transactions on*, vol. 4, no. 1, pp. 341–348, March 2013.

[46] V. A. Shim, K. C. Tan, J. Y. Chia, and J. K. Chong, "Evolutionary algorithms for solving multi-objective travelling salesman problem," *Flexible Services and Manufacturing Journal*, vol. 23, no. 2, pp. 207 – 41, 2011.

[47] V. Shim, K. Tan, and C. Cheong, "A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 42, no. 5, pp. 682–691, Sept 2012.

[48] C. Y. Cheong, K. C. Tan, D. K. Liu, and C. J. Lin, "Multi-objective and prioritized berth allocation in container ports," *Annals of Operations Research*, vol. 180, pp. 63 – 103, 2010.

[49] C. Y. Cheong, K. C. Tan, and B. Veeravalli, "A multi-objective evolutionary algorithm for examination timetabling," *Journal of Scheduling*, vol. 12, no. 2, pp. 121 – 146, 2009.

[50] K. C. Tan, T. H. Lee, Y. H. Chew, and L. H. Lee, "A hybrid multiobjective evolutionary algorithm for solving truck and trailer vehicle routing problems," in *2003 Congress on Evolutionary Computation*, vol. 3, 2003, pp. 2134 – 41.

[51] K. C. Tan, Y. H. Chew, and L. H. Lee, "A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows," *Computational Optimization and Applications*, vol. 34, no. 1, pp. 115 – 51, 2006.

[52] K. C. Tan, C. Y. Cheong, and C. K. Goh, "Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation," *European Journal of Operational Research*, vol. 177, no. 2, pp. 813 – 39, 2007.

[53] J. Xiong, J. Liu, Y. Chen, and H. Abbass, "A knowledge-based evolutionary multiobjective approach for stochastic extended resource investment project scheduling problems," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 5, pp. 742–763, Oct 2014.

[54] J. H. Holland, *Adaptation in natural and artificial systems.* The University of Michigan Press, 1975.

[55] D. E. Goldberg, *Genetic algorithms for search, optimization and machine learning.* Reading, MA: Addison-Wesley, 1989.

[56] J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection.* Cambridge, MA, USA: MIT Press, 1992.

[57] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence through Simulated Evolution.* New York: John Wiley & Sons, 1966.

[58] H.-P. Schwefel, *Numerical Optimization of Computer Models.* New York, NY, USA: John Wiley & Sons, Inc., 1981.

[59] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, June 2001.

[60] R. Storn and K. Price, "Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces," CA, 1995, TR-95–012, Tech. Rep., 1995.

[61] K. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization (Natural Computing Series)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[62] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 26, no. 1, pp. 29–41, Feb 1996.

[63] M. Dorigo and T. Stützle, *Ant colony optimization*. Scituate, MA, USA: Bradford Company, 2004.

[64] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.

[65] H. Mühlenbein and G. Paass, "From recombination of genes to the estimation of distributions i. binary parameters," in *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, ser. PPSN IV, 1996, pp. 178–187.

[66] H. Mühlenbein, J. Bendisch, and H.-M. Voigt, "From recombination of genes to the estimation of distributions ii. continuous parameters," in *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, ser. PPSN IV, 1996, pp. 188–197.

[67] I. Boussaid, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences*, vol. 237, pp. 82 – 117, 2013.

[68] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4 – 31, 2011.

[69] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646 – 57, 2006.

[70] S. Das, A. Abraham, U. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 3, pp. 526–553, June 2009.

[71] J. Zhang and A. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945 – 58, 2009.

[72] A. Qin, V. Huang, and P. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 2, pp. 398–417, April 2009.

[73] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 1, pp. 55–66, Feb 2011.

[74] F. Neri and V. Tirronen, "Recent advances in differential evolution: A survey and experimental analysis," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 61 – 106, 2010.

[75] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: a survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32 – 49, 2011.

[76] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multi-objective optimization: formulation discussion and generalization," in *Proceedings of the 5th International Conference on Genetic Algorithms*, 1993, pp. 416–423.

[77] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, Sep. 1994.

[78] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in *In Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, 1994, pp. 82–87.

[79] J. Knowles and D. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, June 2000.

[80] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *Evolutionary Computation, IEEE Transactions on*, vol. 3, no. 4, pp. 257–271, Nov 1999.

[81] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182 – 97, 2002.

[82] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design, Optimisation, and Control*. CIMNE, Barcelona, Spain, 2002, pp. 95–100.

[83] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 28, no. 3, pp. 392–403, Aug 1998.

[84] A. Jaszkiewicz, "On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - a comparative experiment," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 4, pp. 402–412, Aug 2002.

[85] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712 – 731, 2007.

[86] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284 – 302, 2009.

[87] C.-M. Chen, Y. Chen, and Q. Zhang, "Enhancing MOEA/D with guided mutation and priority update for multi-objective optimization," in *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, May 2009, pp. 209–216.

[88] T.-C. Chiang and Y.-P. Lai, "MOEA/D-AMS: Improving MOEA/D by an adaptive mating selection mechanism," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*, June 2011, pp. 1473–1480.

[89] S. Jiang, Z. Cai, J. Zhang, and Y.-S. Ong, "Multiobjective optimization by decomposition with Pareto-adaptive weight vectors," in *Natural Computation (ICNC), 2011 Seventh International Conference on*, vol. 3, July 2011, pp. 1260–1264.

[90] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, Jun. 2014.

[91] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Simultaneous use of different scalarizing functions in MOEA/D," in *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '10. New York, NY, USA: ACM, 2010, pp. 519–526.

[92] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 3, pp. 450–455, June 2014.

[93] H. Sato, "Inverted PBI in MOEA/D and its impact on the search performance on multi and many-objective optimization," in *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*, ser. GECCO '14. New York, NY, USA: ACM, 2014, pp. 645–652.

[94] M. Gong, F. Liu, W. Zhang, L. Jiao, and Q. Zhang, "Interactive MOEA/D for multi-objective decision making," in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '11. New York, NY, USA: ACM, 2011, pp. 721–728.

[95] S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes," *Evolutionary Computation, IEEE Transactions on*, vol. 16, no. 3, pp. 442–446, June 2012.

[96] H. Li and D. Landa-Silva, "An adaptive evolutionary multi-objective approach based on simulated annealing," *Evolutionary Computation*, vol. 19, no. 4, pp. 561–595, Dec 2011.

[97] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evolutionary Computation*, vol. 22, no. 2, pp. 231–264, June 2014.

[98] H. R. Topcuoglu, B. Demiroz, and M. Kandemir, "Solving the register allocation problem for embedded systems using a hybrid evolutionary algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 5, pp. 620 – 34, 2007.

[99] C.-F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 2, pp. 997 – 1006, 2004.

[100] Y. Fu, M. Ding, C. Zhou, and H. Hu, "Route planning for unmanned aerial vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 6, pp. 1451 – 65, 2013.

[101] N. Krasnogor and J. Smith, "A tutorial for competent memetic algorithms: model, taxonomy, and design issues," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 5, pp. 474 – 88, 2005.

[102] X. Chen, Y.-S. Ong, M.-H. Lim, and K. C. Tan, "A multi-facet survey on memetic computation," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 5, pp. 591 – 607, 2011.

[103] F. Neri and C. Cotta, "Memetic algorithms and memetic computing optimization: A literature review," *Swarm and Evolutionary Computation*, vol. 2, pp. 1 – 14, 2012.

[104] H. Duan, Q. Luo, Y. Shi, and G. Ma, "Hybrid particle swarm optimization and genetic algorithm for multi-UAV formation reconfiguration," *IEEE Computational Intelligence Magazine*, vol. 8, no. 3, pp. 16 – 27, 2013.

[105] I. Ciornei and E. Kyriakides, "Hybrid ant colony-genetic algorithm (GAAPI) for global continuous optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 1, pp. 234 – 245, 2012.

[106] J. Sun, Q. Zhang, and E. P. Tsang, "DE/EDA: A new evolutionary algorithm for global optimization," *Information Sciences*, vol. 169, no. 3-4, pp. 249 – 262, 2005.

[107] S. Ghosh, S. Das, S. Roy, S. K. M. Islam, and P. N. Suganthan, "A differential covariance matrix adaptation evolutionary algorithm for real parameter optimization," *Information Sciences*, vol. 182, no. 1, pp. 199 – 219, 2012.

[108] M. G. Epitropakis, V. P. Plagianakos, and M. N. Vrahatis, "Evolving cognitive and social experience in particle swarm optimization through differential evolution: a hybrid approach," *Information Sciences*, vol. 216, pp. 50 – 92, 2012.

[109] B. Xin, J. Chen, J. Zhang, H. Fang, and Z.-H. Peng, "Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: a review and taxonomy," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 5, pp. 744 – 67, 2012.

[110] J. A. Vrugt and B. A. Robinson, "Improved evolutionary optimization from genetically adaptive multimethod search," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 3, pp. 708 – 11, 2007.

[111] J. A. Vrugt, B. A. Robinson, and J. M. Hyman, "Self-adaptive multimethod search for global optimization in real-parameter spaces," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 243 – 259, 2009.

[112] V. A. Shim, K. C. Tan, and H. Tang, "Adaptive memetic computing for evolutionary multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. PP, no. 99, 2014.

[113] J. Vesterstrm and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," in *Proceedings of the 2004 Congress on Evolutionary Computation, CEC2004*, vol. 2, 2004, pp. 1980 – 1987.

[114] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, pp. 115 – 48, 1995.

[115] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 6, pp. 580 – 93, 2002.

[116] N. Padhye, K. Deb, and P. Mittal, “Boundary handling approaches in particle swarm optimization,” in *Advances in Intelligent Systems and Computing*, vol. 201, no. 1, 2013, pp. 287 – 298.

[117] K. Deb, “An efficient constraint handling method for genetic algorithms,” *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2-4, pp. 311 – 38, 2000.

[118] H. K. Singh, A. Isaacs, T. Ray, and W. Smith, “Infeasibility driven evolutionary algorithm (IDEA) for engineering design optimization,” in *AI 2008: Advances in Artifical Intelligence. 21st Australasian Joint Conference on Artificial Intelligence*, Berlin, Germany, 2008, pp. 104 – 15.

[119] E. Mezura-Montes and C. A. Coello Coello, “A simple multimembered evolution strategy to solve constrained optimization problems,” *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 1, pp. 1 – 17, 2005.

[120] ——, “Constraint-handling in nature-inspired numerical optimization: Past, present and future,” *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 173 – 194, 2011.

[121] R. Mallipeddi and P. N. Suganthan, “Ensemble of constraint handling techniques,” *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 4, pp. 561 – 79, 2010.

[122] J. Derrac, S. Garcia, D. Molina, and F. Herrera, “A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms,” *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3 – 18, 2011.

[123] T. Gjengedal, “Emission constrained unit-commitment (ECUC),” *Energy Conversion, IEEE Transactions on*, vol. 11, no. 1, pp. 132–138, Mar 1996.

[124] J. Mary Anita and I. Jacob Raglend, “Solution of emission constrained unit commitment problem using shuffled frog leaping algorithm,” in *Circuits, Power and Computing Technologies (ICCPCT), 2013 International Conference on*, March 2013, pp. 93–98.

[125] B. Saravanan and E. R. Vasudevan, “Emission constrained unit commitment problem solution using invasive weed optimization algorithm,” in *Advances in Electrical Engineering (ICAEE), 2014 International Conference on*, Jan 2014, pp. 1–6.

[126] R. T. F. A. King and H. C. S. Rughooputh, “Elitist multiobjective evolutionary algorithm for environmental/economic dispatch,” in *Evolutionary Computation, 2003. CEC ’03. The 2003 Congress on*, vol. 2, Dec 2003, pp. 1108–1114.

[127] S. Dhanalakshmi, S. Kannan, K. Mahadevan, and S. Baskar, "Application of modified NSGA-II algorithm to combined economic and emission dispatch problem," *International Journal of Electrical Power & Energy Systems*, vol. 33, no. 4, pp. 992 – 1002, 2011.

[128] M. A. Abido, "Environmental/economic power dispatch using multiobjective evolutionary algorithms," *Power Systems, IEEE Transactions on*, vol. 18, no. 4, pp. 1529–1537, Nov 2003.

[129] ——, "Multiobjective evolutionary algorithms for electric power dispatch problem," *Trans. Evol. Comp*, vol. 10, no. 3, pp. 315–329, Sep. 2006.

[130] L. Wang and C. Singh, "Environmental/economic power dispatch using a fuzzified multi-objective particle swarm optimization algorithm fuzzified multi-objective particle swarm optimization algorithm," *Electric Power Systems Research*, vol. 77, no. 12, pp. 1654 – 64, 2007.

[131] S. Agrawal, B. K. Panigrahi, and M. K. Tiwari, "Multiobjective particle swarm algorithm with fuzzy clustering for electrical power dispatch," *Trans. Evol. Comp*, vol. 12, no. 5, pp. 529–541, Oct. 2008.

[132] Y. Zhang, D.-W. Gong, and Z. Ding, "A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch," *Inf. Sci.*, vol. 192, pp. 213–227, Jun. 2012.

[133] J. Cai, X. Ma, Q. Li, L. Li, and H. Peng, "A multi-objective chaotic particle swarm optimization for environmental/economic dispatch," *Energy Conversion and Management*, vol. 50, no. 5, pp. 1318 – 1325, 2009.

[134] ——, "A multi-objective chaotic ant swarm optimization for environmental/economic dispatch," *International Journal of Electrical Power & Energy Systems*, vol. 32, no. 5, pp. 337 – 344, 2010.

[135] Y. Lu, J. Zhou, H. Qin, Y. Wang, and Y. Zhang, "Environmental/economic dispatch problem of power system by using an enhanced multi-objective differential evolution algorithm," *Energy Conversion and Management*, vol. 52, no. 2, pp. 1175 – 1183, 2011.

[136] A. Ghasemi, "A fuzzified multi objective interactive honey bee mating optimization for environmental/economic power dispatch with valve point effect," *International Journal of Electrical Power & Energy Systems*, vol. 49, no. 0, pp. 308 – 321, 2013.

[137] R. Zhang, J. Zhou, L. Mo, S. Ouyang, and X. Liao, "Economic environmental dispatch using an enhanced multi-objective cultural algorithm," *Electric Power Systems Research*, vol. 99, no. 0, pp. 18 – 29, 2013.

[138] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, "A simulated annealing-based multiobjective optimization algorithm: AMOSA," *Evolutionary Computation, IEEE Transactions on*, vol. 12, no. 3, pp. 269–283, June 2008.

[139] Y.-F. Li, N. Pedroni, and E. Zio, "A memetic evolutionary multi-objective optimization method for environmental power unit commitment," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 2660 – 9, 2013.

[140] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602 – 622, 2014.

[141] A. E. Eiben and S. K. Smit, "Parameter tuning for configuring and analyzing evolutionary algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 19 – 31, 2011.

[142] K. Sindhya, K. Miettinen, and K. Deb, "A hybrid framework for evolutionary multi-objective optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 17, no. 4, pp. 495–511, Aug 2013.

[143] E.-G. Talbi, S. Mostaghim, T. Okabe, H. Ishibuchi, G. Rudolph, and C. A. Coello Coello, "Parallel approaches for multiobjective optimization," vol. 5252 LNCS, Tiergartenstrasse 17, Heidelberg, D-69121, Germany, 2008, pp. 349 – 372.

[144] L. T. Anstine, R. E. Burke, J. E. Casey, R. Holgate, R. S. John, and H. G. Stewart, "Application of probability methods to the determination of spinning reserve requirements for the pennsylvania-new jersey-maryland interconnection," *IEEE Transactions on Power Apparatus and Systems*, vol. 68, pp. 726 – 735, 1963.

[145] H. B. Gooi, D. P. Mendes, K. R. W. Bell, and D. S. Kirschen, "Optimal scheduling of spinning reserve," *IEEE Transactions on Power Systems*, vol. 14, no. 4, pp. 1485 – 92, 1999.

[146] D. Chattopadhyay and R. Baldick, "Unit commitment with probabilistic reserve," in *2002 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No.02CH37309)*, vol. vol.1, Piscataway, NJ, USA, 2002, pp. 280 – 5.

[147] F. Bouffard and F. D. Galiana, "An electricity market with a probabilistic spinning reserve criterion," *Power Systems, IEEE Transactions on*, vol. 19, no. 1, pp. 300–307, Feb 2004.

[148] D. N. Simopoulos, S. D. Kavatza, and C. D. Vournas, "Reliability constrained unit commitment using simulated annealing," *Power Systems, IEEE Transactions on*, vol. 21, no. 4, pp. 1699–1706, Nov 2006.

[149] L. Wang and C. Singh, "Unit commitment considering generator outages through a mixed-integer particle swarm optimization algorithm," *Applied Soft Computing*, vol. 9, no. 3, pp. 947 – 953, 2009.

[150] C. A. Georgopoulou and K. C. Giannakoglou, "Two-level, two-objective evolutionary algorithms for solving unit commitment problems," *Applied Energy*, vol. 86, no. 7–8, pp. 1229 – 1239, 2009.

[151] M. A. Ortega-Vazquez and D. S. Kirschen, "Optimizing the spinning reserve requirements using a cost/benefit analysis," *Power Systems, IEEE Transactions on*, vol. 22, no. 1, pp. 24–33, Feb 2007.

[152] K. Chandrasekaran and S. P. Simon, "Multi-objective scheduling problem: Hybrid approach using fuzzy assisted cuckoo search algorithm," *Swarm and Evolutionary Computation*, vol. 5, no. 0, pp. 1 – 16, 2012.

[153] P. Kall and S. W. Wallace, *Stochastic programming*, 1994.

[154] Q. P. Zheng, J. Wang, and A. L. Liu, "Stochastic optimization for unit commitment–a review," *Power Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–12, 2014.

[155] L. Wu, M. Shahidehpour, and T. Li, "Stochastic security-constrained unit commitment," *Power Systems, IEEE Transactions on*, vol. 22, no. 2, pp. 800–811, May 2007.

[156] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*, ser. Princeton Series in Applied Mathematics. Princeton University Press, October 2009.

[157] D. Bertsimas, E. Litvinov, X. A. Sun, J. Zhao, and T. Zheng, "Adaptive robust optimization for the security constrained unit commitment problem," *Power Systems, IEEE Transactions on*, vol. 28, no. 1, pp. 52–63, Feb 2013.

[158] A. Street, F. Oliveira, and J. M. Arroyo, "Contingency-constrained unit commitment with n - k security criterion: A robust optimization approach," *Power Systems, IEEE Transactions on*, vol. 26, no. 3, pp. 1581–1590, Aug 2011.

[159] Q. Wang, J.-P. Watson, and Y. Guan, "Two-stage robust optimization for n-k contingency-constrained unit commitment," *Power Systems, IEEE Transactions on*, vol. 28, no. 3, pp. 2366–2375, Aug 2013.

[160] P. Xiong and P. Jirutitijaroen, "An adjustable robust optimization approach for unit commitment under outage contingencies," in *Power and Energy Society General Meeting, 2012 IEEE*, July 2012, pp. 1–8.

[161] K. K. Kariuki and R. N. Allan, "Evaluation of reliability worth and value of lost load," *Generation, Transmission and Distribution, IEE Proceedings-*, vol. 143, no. 2, pp. 171–180, Mar 1996.

[162] D. N. Simopoulos, S. D. Kavatza, and C. D. Vournas, "Reliability constrained unit commitment using simulated annealing," *Power Systems, IEEE Transactions on*, vol. 21, no. 4, pp. 1699–1706, Nov 2006.

[163] R. Billinton and R. N. Allan, *Reliability evaluation of power systems.* London: Plenum, 1996.

[164] R. N. Allan, R. Billinton, and N. M. K. Abdel-Gawad, "The IEEE reliability test system - extensions to and evaluation of the generating system," *Power Engineering Review, IEEE*, vol. PER-6, no. 11, pp. 24–24, Nov 1986.

[165] L. Rachmawati and D. Srinivasan, "Multiobjective evolutionary algorithm with controllable focus on the knees of the Pareto front," *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 4, pp. 810–824, Aug 2009.

[166] I. Das, "On characterizing the 'knee' of the pareto curve based on normal-boundary intersection," *Structural Optimization*, vol. 18, no. 2-3, pp. 107 – 115, 1999.

[167] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 8, no. 3, pp. 256–279, June 2004.

[168] K. Deb, M. Mohan, and S. Mishra, "Evaluating the ϵ-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions," *Evolutionary Computation*, vol. 13, no. 4, pp. 501–525, 2005.

[169] V. S. Pappala, I. Erlich, K. Rohrig, and J. Dobschinski, "A stochastic model for the optimal operation of a wind-thermal power system," *Power Systems, IEEE Transactions on*, vol. 24, no. 2, pp. 940–950, May 2009.

[170] T.-Y. Lee, "Optimal spinning reserve for a wind-thermal power system using EIPSO," *Power Systems, IEEE Transactions on*, vol. 22, no. 4, pp. 1612–1621, Nov 2007.

[171] M. Black and G. Strbac, "Value of bulk energy storage for managing wind power fluctuations," *Energy Conversion, IEEE Transactions on*, vol. 22, no. 1, pp. 197–205, March 2007.

[172] M. A. Ortega-Vazquez and D. S. Kirschen, "Estimating the spinning reserve requirements in systems with significant wind power generation penetration," *Power Systems, IEEE Transactions on*, vol. 24, no. 1, pp. 114–124, Feb 2009.

[173] A. Tuohy, P. Meibom, E. Denny, and M. O'Malley, "Unit commitment for systems with significant wind penetration," *Power Systems, IEEE Transactions on*, vol. 24, no. 2, pp. 592–601, May 2009.

[174] B. Ji, X. Yuan, Z. Chen, and H. Tian, "Improved gravitational search algorithm for unit commitment considering uncertainty of wind power," *Energy*, vol. 67, no. 0, pp. 52 – 62, 2014.

[175] B. Ji, X. Yuan, X. Li, Y. Huang, and W. Li, "Application of quantum-inspired binary gravitational search algorithm for thermal unit commitment with wind power integration," *Energy Conversion and Management*, vol. 87, pp. 589 – 598, 2014.

[176] E. Denny and M. O'Malley, "Wind generation, power system operation, and emissions reduction," *Power Systems, IEEE Transactions on*, vol. 21, no. 1, pp. 341–347, Feb 2006.

[177] H. A. Gil and G. Joos, "Generalized estimation of average displaced emissions by wind generation," *Power Systems, IEEE Transactions on*, vol. 22, no. 3, pp. 1035–1043, Aug 2007.

[178] Y. Zhu, J. Wang, and B. Qu, "Multi-objective economic emission dispatch considering wind power using evolutionary algorithm based on decomposition," *International Journal of Electrical Power & Energy Systems*, vol. 63, no. 0, pp. 434 – 445, 2014.

[179] Y. Zhang, F. Yao, H. H. C. Iu, T. Fernando, and H. Trinh, "Wind–thermal systems operation optimization considering emission problem," *International Journal of Electrical Power & Energy Systems*, vol. 65, no. 0, pp. 238 – 245, 2015.

[180] J. Kabouris and F. D. Kanellos, "Impacts of large-scale wind penetration on designing and operation of electric power systems," *Sustainable Energy, IEEE Transactions on*, vol. 1, no. 2, pp. 107–114, July 2010.

[181] J. Lin, L. Cheng, Y. Chang, K. Zhang, B. Shu, and G. Liu, "Reliability based power systems planning and operation with wind power integration: A review to models, algorithms and applications," *Renewable and Sustainable Energy Reviews*, vol. 31, no. 0, pp. 921 – 934, 2014.

[182] Y. M. Atwa, E. F. El-Saadany, M. M. A. Salama, and R. Seethapathy, "Optimal renewable resources mix for distribution system energy loss minimization," *Power Systems, IEEE Transactions on*, vol. 25, no. 1, pp. 360–370, Feb 2010.

[183] R.-H. Liang and J.-H. Liao, "A fuzzy-optimization approach for generation scheduling with wind and solar energy systems," *Power Systems, IEEE Transactions on*, vol. 22, no. 4, pp. 1665–1674, Nov 2007.

[184] Z. Li and M. Shahidehpour, "Security-constrained unit commitment for simultaneous clearing of energy and ancillary services markets," *Power Systems, IEEE Transactions on*, vol. 20, no. 2, pp. 1079–1088, May 2005.

[185] Y. Fu, M. Shahidehpour, and Z. Li, "Security-constrained unit commitment with ac constraints," *Power Systems, IEEE Transactions on*, vol. 20, no. 2, pp. 1001–1013, May 2005.

[186] A. Lotfjou, M. Shahidehpour, Y. Fu, and Z. Li, "Security-constrained unit commitment with ac/dc transmission systems," *Power Systems, IEEE Transactions on*, vol. 25, no. 1, pp. 531–542, Feb 2010.

[187] L. Wu, M. Shahidehpour, and T. Li, "Stochastic security-constrained unit commitment," *Power Systems, IEEE Transactions on*, vol. 22, no. 2, pp. 800–811, May 2007.

[188] J. Wang, M. Shahidehpour, and Z. Li, "Security-constrained unit commitment with volatile wind power generation," *Power Systems, IEEE Transactions on*, vol. 23, no. 3, pp. 1319–1327, Aug 2008.

[189] A. Mambrini and D. Sudholt, "Design and analysis of adaptive migration intervals in parallel evolutionary algorithms," in *GECCO 2014 - Proceedings of the 2014 Genetic and Evolutionary Computation Conference*, Vancouver, BC, Canada, 2014, pp. 1047 – 1054.

[190] S. Gee, K. C. Tan, V. Shim, and N. Pal, "Online diversity assessment in evolutionary multiobjective optimization: A geometrical perspective," *Evolutionary Computation, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[191] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, March 2011.

Appendix A

Table A.1 Generating unit data for the (base) ten unit system

	Unit 1	**Unit 2**	**Unit 3**	**Unit 4**	**Unit 5**	**Unit 6**	**Unit 7**	**Unit 8**	**Unit 9**	**Unit 10**
P_{max} (MW)	455	455	130	130	162	80	85	55	55	55
P_{min} (MW)	150	150	20	20	25	20	25	10	10	10
$a(\$/MW^{2h})$	0.00048	0.00031	0.002	0.00211	0.00398	0.00712	0.00079	0.00413	0.00222	0.00173
$b(\$/MWh)$	16.19	17.26	16.6	16.5	19.7	22.26	27.74	25.92	27.27	27.79
$c(\$/h)$	1000	970	700	680	450	370	480	660	665	670
$a_1(lb/MW^{2h})$	0.0046	0.0046	0.0068	0.0068	0.0042	0.0042	0.0465	0.0465	0.0465	0.0470
$b_1(lb/MWh)$	-0.5112	-0.5112	0.5455	-0.5455	0.3277	0.3277	-3.9023	-3.9023	-3.9524	-3.9864
$c_1(lb/h)$	42.90	42.90	40.27	40.27	13.86	13.86	330.00	330.00	350.00	360.00
$MUT(h)$	8	8	5	5	6	3	3	1	1	1
$MDT(h)$	8	8	5	5	6	3	3	1	1	1
$HSC(\$)$	4500	5000	550	560	900	170	260	30	30	30
$CSC(\$)$	9000	10000	1100	1120	1800	340	520	60	60	60
$T_{cold}(h)$	5	5	4	4	4	2	2	0	0	0
$I_{state}(h)$	8	8	-5	-5	-6	-3	-3	-1	-1	-1

Table A.2 Forecast load demand data for the (base) ten unit system

Hour	**Demand (MW)**	**Hour**	**Demand (MW)**	**Hour**	**Demand (MW)**
1	700	9	1,300	17	1,000
2	750	10	1,400	18	1,100
3	850	11	1,450	19	1,200
4	950	12	1,500	20	1,400
5	1,000	13	1,400,	21	1,300
6	1,100	14	1,300	22	1,100
7	1,150	15	1,200	23	900
8	1,200	16	1,050	24	800

Algorithm 5: Pseudo-code for Repair Operation

```
begin
    \\ Calculating power output generation for each hour;
    Init UCM: UCM ⟵ ZEROS(N, T_max) ;
    Init PowerOutput: PowerOutput ⟵ ZEROS(T_max) ;
    for time = 1 : T_max do
        PowerOutput(time) ⟵ SUM(UCM(1 : N, time) · RPM(1 : N, time)) ;
    end
    \\ Repair Operation;
    init tolerance ⟵ 10^-6;
    for time = 1 : T_max do
        if PowerOutput(time) < LoadDemand(time) then
            Gap = LoadDemand(time) − PowerOutput(time);
            \\ Incrementing power output of committed generators according to ascending
            order of PL to meet load demand;
            for p = 1 : 1 : 10 do
                for unit = 1 : N do
                    if ucm(unit, time) == 1&&priority(unit) ==
                    p&&RPM(unit, time) < Pmax(unit) then
                        Diff = Pmax(unit) − RPM(unit, time);
                        if Gap ≤ Diff then
                            RPM(unit, time) = RPM(unit, time) + Gap;
                            Gap = 0.0;
                        else
                            RPM(unit, time) = Pmax(unit);
                            Gap = Gap − Diff;
                        end
                    end
                    if Gap < tolerance then
                        break;
                    end
                end
                if Gap < tolerance then
                    break;
                end
            end
        else
            if PowerOutput(time) > LoadDemand(time) then
                Gap = PowerOutput(time) − LoadDemand(time);
                \\ Decrementing power output of committed generators according to
                descending order of PL to meet load demand;
                for p = 10 : −1 : 1 do
                    for unit = 1 : N do
                        if ucm(unit, time) == 1&&priority(unit) ==
                        p&&RPM(unit, time) > Pmin(unit) then
                            Diff = RPM(unit, time) − Pmin(unit);
                            if Gap ≤ Diff then
                                RPM(unit, time) = RPM(unit, time) + Gap;
                                Gap = 0.0;
                            else
                                RPM(unit, time) = Pmin(unit);
                                Gap = Gap − Diff;
                            end
                        end
                        if Gap < tolerance then
                            break;
                        end
                    end
                    if Gap < tolerance then
                        break;
                    end
                end
            end
        end
    end
end
```