Investigating Evolutionary Computation with Smart Mutation for three types of Economic Load Dispatch Optimisation Problem

Sunny Orike

Submitted for the degree of Doctor of Philosophy

Heriot-Watt University, Edinburgh, United Kingdom

Department of Computer Science

Supervisor:

Professor David Wolfe Corne

January, 2015

Abstract

The Economic Load Dispatch (ELD) problem is an optimisation task concerned with how electricity generating stations can meet their customers' demands while minimising under/over-generation, and minimising the operational costs of running the generating units. In the conventional or Static Economic Load Dispatch (SELD), an optimal solution is sought in terms of how much power to produce from each of the individual generating units at the power station, while meeting (predicted) customers' load demands. With the inclusion of a more realistic dynamic view of demand over time and associated constraints, the Dynamic Economic Load Dispatch (DELD) problem is an extension of the SELD, and aims at determining the optimal power generation schedule on a regular basis, revising the power system configuration (subject to constraints) at intervals during the day as demand patterns change.

Both the SELD and DELD have been investigated in the recent literature with modern heuristic optimisation approaches providing excellent results in comparison with classical techniques. However, these problems are defined under the assumption of a regulated electricity market, where utilities tend to share their generating resources so as to minimise the total cost of supplying the demanded load. Currently, the electricity distribution scene is progressing towards a restructured, liberalised and competitive market. In this market the utility companies are privatised, and naturally compete with each other to increase their profits, while they also engage in bidding transactions with their customers. This formulation is referred to as: Bid-Based Dynamic Economic Load Dispatch (BBDELD).

This thesis proposes a Smart Evolutionary Algorithm (SEA), which combines a standard evolutionary algorithm with a "smart mutation" approach. The so-called 'smart' mutation operator focuses mutation on genes contributing most to costs and penalty violations, while obeying operational constraints. We develop specialised versions of SEA for each of the SELD, DELD and BBDELD problems, and show that this approach is superior to previously published approaches in each case. The thesis also applies the approach to a new case study relevant to Nigerian electricity deregulation. Results on this case study indicate that our SEA is able to deal with larger scale energy optimisation tasks.

Dedication

This thesis is dedicated to the memories of my beloved parents – Chief Daniel A. P. Orike and Mrs Bernice N. Orike; who died on 26th July 2013 and 12th December 2013 respectively, at the peak of my PhD.

You were both wonderful parents and invested immensely in our lives. It is just unfortunate you that did not live a little bit longer to reap the fruits. You will surely be missed by all of us.

It is well!

Acknowledgement

First, I thank the Almighty God for He is my 'Source'.

I am highly indebted to my principal supervisor – Professor D. W. Corne, for his invaluable advice, encouragement, guidance and provision of all needed support, right from conception to completion of this PhD.

I appreciate the tireless efforts of my academic mentor – Engr. Professor C. O. Ahiakwo of Electrical Engineering Department, Niger Delta University, Wilberforce Island, Nigeria, for motivating me, providing all necessary resources/data needed from the power industry and painstakingly proof-reading this thesis.

I appreciate the fatherly role and concern to me, of Engr. Professor M. J. Ayotamuno – the Deputy Vice Chancellor of Rivers State University of Science and Technology, Port Harcourt, Nigeria; all through the period of this programme.

My special thanks also go to the Managements of Niger Delta Development Commission (NDDC) of the Federal Republic of Nigeria, for funding this research; and the Rivers State University of Science and Technology, Port Harcourt, Nigeria (my employer) for supporting me financially.

Lastly (but not the least), I thank my dear wife Dayibara, my two daughters: Chukwuoma and Chukwumbana for their patience, tolerance, understanding, support and prayers all through the period of this PhD; and my son Chukwumela, who was born on 29th December 2014, just before the final submission of this thesis.

May God bless you all!

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ACO	-	Ant Colony Optimisation
ANN	-	Artificial Neural Network
AIS	-	Artificial Immune System
APBIL	-	Adaptive Population-Based Incremental Learning
BCO	-	Bee Colony Optimisation
BBDELD	-	Bid-Based Dynamic Economic Load Dispatch
BBO	-	Biogeography-Based Optimisation
BEA	-	Basic Evolutionary Algorithm
BPE	-	Bureau for Public Enterprises
CEED	-	Combined Economic and Emission Dispatch
CGA	-	Conventional Genetic Algorithm
DE	-	Differential Evolution
DELD	-	Dynamic Economic Load Dispatch
DISCO	-	Distribution Company
DP	-	Dynamic Programming
DR	-	Down Ramp
EA	-	Evolutionary Algorithm
ECELD	-	Emission Constrained Economic Load Dispatch
ECN	-	Electricity Corporation of Nigeria
ECJ	-	Evolutionary Computation in Java
EDA	-	Estimation of Distribution Algorithm
ELD	-	Economic Load Dispatch
EP	-	Evolutionary Programming
EPIC	-	Electric Power Implementation Committee
EPSRA	-	Electric Power Sector Reform Act
ES	-	Evolution Strategy
FL	-	Fuzzy Logic
GA	-	Genetic Algorithm
GENCO	-	Generation Company
GP	-	Genetic Programming
GSO	-	Group Search Optimiser
HNN	-	Hopfield Neural Network

HS	-	Harmony Search
HSI	-	Habitat Suitability Index
IGA	-	Improved Genetic Algorithm
IP	-	Interior Point
IPP	-	Independent Power Producers
IPQP	-	Interior Point Quadratic Programming
ISO	-	Independent System Operator
LIM	-	Lambda Iteration Method
LP	-	Linear Programming
MA	-	Memetic Algorithm
μGA	-	Micro Genetic Algorithm
MGSO	-	Modified Group Search Optimiser Method
MOEA	-	Multi-Objective Evolutionary Algorithm
MOOP	-	Multi-Objective Optimisation Problem
MSFLA	-	Modified Shuffled Frog Leaping Algorithm
MTS	-	Multiple Tabu Search
MU	-	Multiplier Updating
NDA	-	Niger Dam Authority
NEPA	-	National Electric Power Authority
NEPP	-	National Electric Power Policy
NERC	-	Nigeria Electricity Regulation Commission
NESCO	-	National Electricity Supply Company
NP	-	Non-deterministic Polynomial-time
PCIPQP	-	Predicted-Corrected Interior Point Quadratic Programming
PBDELD	-	Price-Based Dynamic Economic Load Dispatch
PHCN	-	Power Holding Company of Nigeria
PMBGA	-	Probabilistic Model Building Genetic Algorithms
IPP	-	Independent Power Provision
MGSO	-	Modified Group Search Optimiser
MSFLA	-	Modified Shuffled Frog Leaping Algorithm
NELMCO	-	Nigerian Electricity Liability Management Company
NIBET	-	Nigerian Bulk Electricity Trading
NSGA II	-	Non-dominated Sorting Algorithm
OCDD	-	Optimal Control Dynamic Dispatch
ODD	-	Optimal Dynamic Dispatch

PF	-	Penalty Factor
PS	-	Producer Scrounger
PSO	-	Particle Swarm Optimisation
QP	-	Quadratic Programming
REA	-	Rural Electrification Agency
RMHC	-	Random Mutation Hill Climber
ROI	-	Region of Interest
SA	-	Simulated Annealing
SADP	-	Successive Approximation Dynamic Programming
SAHC	-	Steepest Ascent Hill Climber
SAW	-	Stepwise Adaptation of Weights
SEA	-	Smart Evolutionary Algorithm
SELD	-	Static Economic Load Dispatch
SF	-	Scaling Factor
SFLA	-	Shuffled Frog Leaping Algorithm
SI	-	Swarm Intelligence
SIV	-	Suitability Index Variables
SLP	-	Successive Linear Programming
SP	-	Social Profit
SQP	-	Sequential Quadratic Programming
SRM	-	Structural Risk Minimisation
SVM	-	Support Vector Machines
TCPC	-	Technical Committee on Privatisation and Commercialisation
TRANSCO	-	Transmission Company
TS	-	Tabu Search
TSP	-	Travelling Salesman Problem
UR	-	Up Ramp
QN	-	Quasi Newton
VEGA	-	Vector Evaluated Evolutionary Algorithm

Chapter 1

Introduction

1.1 Research Context

Most optimisation problems in real world applications are affected by conditions that change with time; this makes the ability to solve problems in such dynamic environments both important and challenging [1]. This is the case for problems in the context of electrical power systems, where there are constant changes affecting the power variables, problem scenarios and operational constraints, resulting in the optimal solutions changing over time. In recent decades, there has been a tremendous growth and increased interest in generation, transmission and distribution of electricity. Since it is very uneconomical to store electrical power over a period of time, stakeholders in the industry continually aim to ensure that a balance is achieved between demand and supply of electricity [2]. Realisation of an effective, reliable, secure and economic allocation of the consumers' power demands among the generating units creates dynamism in the sector.

Energy is a very important and indispensable tool for socio-economic growth and national development of any country. Electrical energy is the most widely used form of energy in the world [3], and its demand is on a daily increase. Electrical power engineers are therefore constantly engaged with every step in the process of generation, transmission and distribution of electrical energy, and they encounter challenging problems in their quest to deliver secure, reliable, and continuous operation of the sector. It is a topmost concern of every nation (both developed and developing) to establish and maintain an efficient electricity industry. Research that addresses optimisation (or other) problems that arise in this context are therefore becoming both necessary and timely.

The Economic Load Dispatch (ELD) problem is aimed at determining the optimal power generation schedule for online generators (typically the units in a power station) on a near-real time basis. The generated power of each unit is determined with respect to a predicted load demand. The Dynamic Economic Load Dispatch (DELD) problem, an extension of the Static Economic Load Dispatch problem (SELD) [4], is one of several optimisation problems that need repeatedly to be solved in the electricity sector. Solving the SELD minimises the total generation cost among the committed units, satisfying all constraints, but under the assumption of a static level of demand. But, in practical systems involving ramp-rate limits (which constrain the changes that can be made to the settings of an individual generator between periods); operational decisions at a given hour will affect the decision at a later hour. Due to the change in load conditions arising from these limits, the power generation has to be altered to meet the demand [4, 5]. This is a major limitation of the SELD, which is addressed in the formulation of the DELD. The DELD takes into consideration the dynamic costs involved in changing from one output level to another [6], and is therefore a more applicable formulation of the ELD problem as it is faced by generating stations worldwide; but it is also a more difficult and complex optimisation problem. Until now, the DELD has been treated as a series of unconnected static problems.

Both the SELD and DELD have been investigated in the recent stochastic optimisation literature with modern meta-heuristic approaches providing excellent results in comparison with classical techniques. However, these optimisation problems are defined under the assumption of a regulated electricity market, where utilities tend to share their generating resources so as to minimise the total cost of supplying the demanded load. In many national and regional contexts, the electricity distribution scene is progressing towards a restructured, liberalised and competitive market, where the utility companies are privatised. Utility companies compete with each other to increase their profits, and remain concerned with optimising the allocation of load among their generating units (essentially the SELD and DELD), but are also engaged in bidding-based transactions from their customers. In simpler terms, for example, generating companies make bids in the form of how much they will charge the customer for meeting their demand in the next time period, and the customer is free to go with the lowest bidder. Issues of optimisation in this deregulated scenario, particularly in regard to economic load dispatch, are considered in [7, 8]. Deregulation shifts the goal of the ELD problem from the traditional cost minimisation towards maximisation of *social* *profit*, in which costs are minimised while utilities compete for market share. From another viewpoint, formulations of this so-called *profit-based dynamic economic load dispatch* problem are concerned with ensuring high profits through customer benefits [9, 10, 11, 12, 13].

In recent years, there has being a growing interest in the application of evolutionary algorithms (EAs) to power systems optimisation. EAs have successfully been used to solve a range of electrical power systems optimisation problems, including electrical power operations, ELD problems, unit commitment problems, planning and scheduling of distribution systems, control of reactive power, etc [9]. In comparison with other search methods that have classically been used for such problems, EAs tend to find good solutions in relatively shorter time. However, the literature of EA applications in this area, particularly as regards ELD problems, reveals that many different algorithms have been explored (particle swarm optimisation, evolution strategies, differential evolution, including many variants of these, and similar algorithms [4]); but there have been no attempts to design variants of such approaches that are intelligently tailored to the specific features of ELD problems.

1.2 Thesis Contributions

The following are the main contributions of this thesis:

- We contribute the development and demonstration of a "smart mutation" based approach in the context of applying EAs to certain types of real-world problems. Instead of using a generic/off-the-shelf EA-based optimisation package, we implement a smart evolutionary algorithm (SEA), which combines a standard EA with a "smart mutation" operator that is tailored to the problem domain. This operator focuses mutation on genes contributing most to cost and penalty violations in the fitness function. We demonstrate that this approach is successful on a range of problems in the electricity supply industry. We also contribute investigation and analysis of three distinct variants of the smart mutation approach.
- We describe and evaluate an SEA-based approach to solving Static Economic Load Dispatch (SELD) problems. This new approach to SELD is shown to outperform all previously published approaches, on the basis of the common published test problems used in the literature, comprising three benchmark cases

involving 6, 15 and 20 generating units respectively. On the larger two of these problems we find better solutions than have so far been reported in the literature; and (where sufficient information is available) in some cases statistical tests confirm the superiority of SEA on these problems to other recent algorithms including Particle Swarm Optimisation, Evolution Strategies, Differential Evolution, and others.

- Taking the SEA-based approaches that provided superior results in SELD problems, we adapt, describe and evaluate same to solving Dynamic Economic Load Dispatch (DELD) problems. Here, we contribute a dynamic optimisation approach to the ELD. Experiments on benchmark problems from the literature confirm the quality of this approach, and also show that it tends to be superior to alternatives that do not exploit its dynamic nature.

- Building on the successes of the SEA-based approaches on the SELD and DELD problems, we describe and evaluate novel approaches to solving the Bid-Based Dynamic Economic Load Dispatch (BBDELD) problem in a deregulated electricity market. We find our SEA approach, when adapted to the BBDELD context, outperforms previously published BBDELD results on the basis of the common published test problems used in the literature – systems of 3 generators, 2 customers in 2 dispatch periods, and 6 generators, 2 customers in 2 dispatch periods. We also define and show results on a new and larger test case – a system of 10 generators, 6 customers in 12 hourly dispatch periods. The new test case is also a contribution of this thesis.

- We contribute a further large-scale test case of the BBDELD with application to Nigerian electrical power industry, and propose a solution model for the country's deregulated electricity market. This involves extending the larger BBDELD test case to 24 hourly dispatch periods, and utilising test data that corresponds to aspects of the Nigerian market. Solutions relating to this large test case (total load demand of 10,500MW) have previously been investigated for a single dispatch period in the context of an SELD. We demonstrate that our approach (especially M3_SEA3) is able to deal with larger scale energy optimisation tasks. Details of this contribution are available from http://is.gd/orike_research.

1.3 Thesis Structure

The remainder of this thesis is organised as follows:

Chapter 2 describes the background of the research, and is in two parts. The first part introduces various concepts that are central to the thesis. These are optimisation as a problem solving tool; electrical energy and power, power generation; optimal power flow, and the distinction between unit commitment and ELD problems. Chapter two goes on to describe the various formulations of ELD problems, reviews the historical developments in the solution approaches, and identifies the two categories of ELD problems. The second part presents an in-depth tutorial and review of EAs in relation to power system problems. It is followed by a discussion of the concept of multi-objective EAs, and other non-evolutionary search algorithms, and concludes with a review of approaches that are related to smart mutation.

Chapter 3 describes and evaluates a new EA for the SELD problem. The approach combines a standard EA with a smart mutation operator. It considers practical instances of SELD problems involving: minimum/maximum generation limits, power balance, ramp rates and prohibited operating zones. The chapter presents a detailed description of the smart evolutionary algorithm (SEA) developed in the thesis, and performs tuning of genetic parameters to select values for experimental runs based on the SEA. Three versions of the smart mutation operator were tested on three benchmark cases involving 6, 15 and 20 generating units; the ones commonly explored in recent literature, with a focus on the larger problem cases. The results were compared with those reported for a range of recent alternative algorithms.

Chapter 4 describes and evaluates a new EA for the DELD problem, by extending the SELD formulation to a dynamic context. It shows the steps involved in DELD formulation, and reviews the recent literature of related work in DELD problems. From the results that had provided superior results on the three versions of the smart mutation operator from SELD, it adapts algorithms for the dynamic case, and investigates three dynamic optimisation approaches. It compares the results of two problem cases (involving a total of 18 instances of the problems) with each other, and with identified methods from literature.

Chapter 5 describes and evaluates a new EA for maximising social profit in the context of BBDELD formulation in a deregulated electricity environment. It distinguishes between the two profit-based dispatch transactions that operate in a deregulated electricity market: price-based DELD and bid-based DELD, reviewing related work in BBDELD. It describes the solution optimisation procedure, involving three bidding strategies, compares results of the best two versions of the smart mutation operator with previous results available in literature. The chapter ends with defining and showing results on a new, larger test case for the bid-based problem.

Chapter 6 takes a case study approach, relating the research in the previous chapters to the context of the Nigerian electrical power system that has recently been deregulated and privatised. A constrained elitist genetic algorithm was implemented and tested on data representing Nigeria's power system, whose results outperformed two related approaches in the literature. This chapter adapts the BBDELD formulation of Chapter 5 to Nigeria's deregulated electricity market using two versions of the smart mutation operator, with results showing high social profits, thus confirming that the SEA is able to deal well with this larger scale energy optimisation task.

Chapter 7 concludes the thesis. It shows a summary of each chapter as well as the main findings, and ends with our suggestions regarding potentially important and fruitful lines of follow-on research that builds on what has been contributed in this thesis.

1.4 Thesis Publications

1. S. Orike and D. W. Corne, "Evolutionary Algorithms for Bid-Based Dynamic Economic Load Dispatch: A Large-Scale Test Case," in *Proceedings of the 2014 IEEE Symposium Series on Computational Intelligence (SSCI 2014)*, Orlando, Florida, USA, 9th – 12th December, 2014.
2. S. Orike and D. W. Corne, "An Evolutionary Algorithm for Bid-Based Dynamic Economic Load Dispatch in a Deregulated Electricity Market," in *Y. Jin and S. A. Thomas (Eds.), 13th IEEE UK Workshop on Computational Intelligence (UKCI 2013)*, University of Surrey, Guildford, 9th – 11th September 2013.
3. S. Orike and D. W. Corne, "Constrained Elitist Genetic Algorithm for Economic Load Dispatch: Case Study on Nigerian Power System," *International Journal of Computer Applications*, vol. 76, no. 5, pp. 27 – 33, August 2013, Foundation of Computer Science, New York, USA.

4. S. Orike and D. W. Corne, "A Memetic Algorithm for Dynamic Economic Load Dispatch Optimisation," in *Proceedings of the 2013 IEEE Symposium Series on Computational Intelligence (SSCI 2013)*, Singapore, 16th - 19th April 2013.
5. S. Orike and D. W. Corne, "Improved Evolutionary Algorithms for Economic Load Dispatch Optimisation Problems," in *P. De Wilde, G. M. Coghill and A. V. Kononova (Eds.), 12th IEEE UK Workshop on Computational Intelligence (UKCI 2012)*, Heriot-Watt University, Edinburgh, 5th – 7th September 2012.

1.5 Glossary of Terms

A high-level definition of the key terms (words and phrases) as used in the thesis.

Algorithm: A step-by-step procedure for calculations. Computer algorithms are used for computations, data processing or manipulation, and logic reasoning.

Deregulation: A process of moving electrical power industry from government ownership to private ownership, which results in consumers making their choices of electricity suppliers.

Economic Load Dispatch: A process of allocating or distributing system load demands to the various generating units with the aim of minimising total generation cost.

Electrical Power: The rate of flow of electrical energy. It is computed as energy expended divided by time.

Elitism: This is a strategy that ensures that the best solutions of the previous generation are maintained in the next population.

Energy Clearing Price: The price of energy at which power supplied is equal to power demanded.

Evolutionary Algorithms: Computer programs that mimic the process of natural biological evolution. They search a population of potential solutions to a particular problem using the concept of survival of the fittest to realise the best solution.

Fitness: A single figure of merit, based on an objective function which shows how close a given solution is, to achieving its targeted set aims.

Fuel Cost Curve: The input-output curve of a thermal unit, measured in dollars per mega watt hour ($/MWh).

Heat-Rate Curve: The input-output curve of a thermal unit, measured in British thermal unit per mega watt hour (Btu/MWh).

Incremental Cost: The derivative of the generation cost with respect to the generated power.

Independent System Operator: A market operator that regulates the activities of the participating companies and stakeholders, receives bids from generating companies and customers, and performs economic load dispatch with a view of maximising social profits.

Memetic Algorithm: An evolutionary algorithm that combines local search method to improve its fitness by hill-climbing.

Objective Function: A function that maps values of different variables to real-life numbers. It is either a cost or loss function, used in mathematical optimisation, statistics or decision theory.

Optimisation: A method of finding the best possible solution to a problem given a set of limitations or constraints. It is achieved by adjusting the inputs to, or characteristics of a device, mathematical process, or experiment to find the minimum or maximum output/result.

Social Profit: The difference between total customer benefits and total generating costs in the entire dispatch period.

Spot Prices: A set of market prices at each dispatch period.

Unit Commitment: The process of selecting optimally out of the available generating sources to operate in order to meet the expected load. It determines the unit start-up and shut-down schedules, with the aim of minimising system expenditure.

Chapter 2

Background on Power System Optimisation and Evolutionary Algorithms

This chapter describes the application domain background of the research. First, we introduce various pertinent concepts such as optimisation, electrical energy and power, optimal power flow, unit commitment, and economic load dispatch (ELD) problems. We describe the various formulations of ELD problems, present the historical developments in the solution approaches, and identify the two categories of ELD problems. The chapter also presents an in-depth tutorial and review of EAs in relation to power system problems. This is followed by a discussion of the concept of multi-objective EAs, and then other (non-evolutionary) search algorithms. We conclude with a review of elements of the literature that has used ideas related to our smart mutation operator approach.

2.1 The Concept of Optimisation

Optimisation simply means the process of making things better. An optimisation problem is one that finds an optimal or near-optimal solution from a set of feasible solutions, using some proven measures or techniques for evaluating such solutions. The ELD problem is an optimisation task, concerned with how electricity generating companies can meet their customers' power demands while minimising under/over-generation, and minimising their operational costs (the costs of running the generating units). Optimisation is a process of refining something. It involves minimising or maximising a given quantity; an art and science of allocating resources to the best possible effect. According to [14], optimisation is defined as: *"A process of adjusting the inputs to, or characteristics of a device, mathematical process, or experiment to find the minimum or maximum output/result"*.

2.1.1 Optimisation Process

An optimisation process is a method that attempts to find the best possible solution to a problem given a set of limitations or constraints. Figure 2.1 diagrammatically describes the concept of optimisation. The input (variables) is the independent quantity, the output (cost) is the dependant quantity, and the evaluation function measures the quality of the current solution being presented by the input. The feedback loop iterates the optimisation process, until a stopping criterion is reached. This is a quantitative decision, which could be that the best possible (optimal) output is found, or a maximum number of allowed runs of the evaluation function is reached.

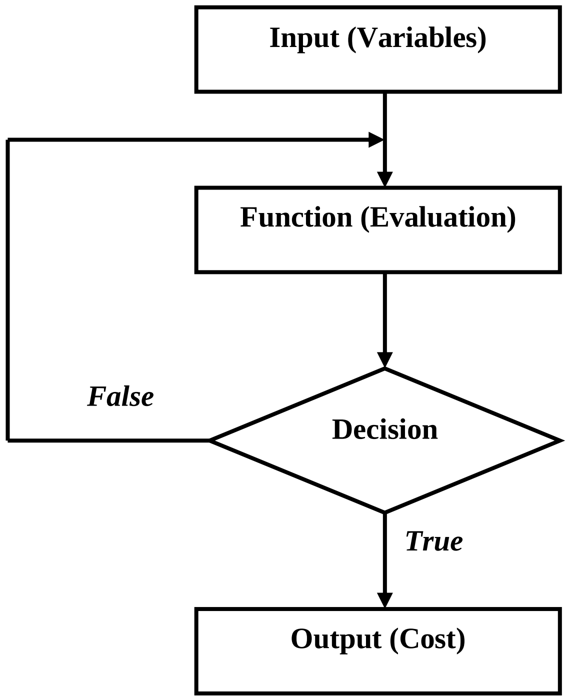

Figure 2.1: A flow diagram showing the concept of optimisation

Optimisation is analogous to the root-finding process in calculus, but while the latter searches for zeros of a function, the former finds zeros of the function derivatives. A major difficulty with optimisation, unlike root-finding is determining if a given minimum or maximum is a global optimum (the best possible solution available) or a local optimum. The concept of optimisation is very important in industrial planning and control, resource allocation, scheduling, decision making, and almost every other area of industry and science.

2.1.2 Categories of Optimisation

There are several categories of optimisation [14]: Trial/error and function optimisation; single-objective and multi-objective optimisation; static and dynamic optimisation; discrete and continuous variables optimisation; constrained and unconstrained optimisation; randomised and minimum/maximum seeking optimisation.

In trial and error optimisation (preferred by most experimentalists), the variables are adjusted without much knowledge of the process which generates the output; while function optimisations (preferred by theoreticians) are described by means of tried/proven formulae, and optimal solutions result from various mathematical manipulations of these formulae. Single-objective optimisation involves only a single variable, while multi-objective optimisation involves more than one variable. In dynamic optimisation, the output changes with time; while it is independent of time in static optimisation. In discrete optimisations, the variables contain finite number of values within a given range; while in continuous optimisations, the variables have infinite number of values. In constrained optimisations, there are variables equalities or inequalities incorporated inside the functions to be optimised, while in unconstrained optimisations, the variables can take any values. Traditional optimisation minimises or maximises a given function. Some predetermined sequences of steps are used in moving from one solution to another; while random optimisation makes use of probability during computations.

2.1.3 The Optimisation Cycle

The Optimisation process, as a cycle, passes through various phases [15]. Figure 2.2 illustrates an optimisation cycle.

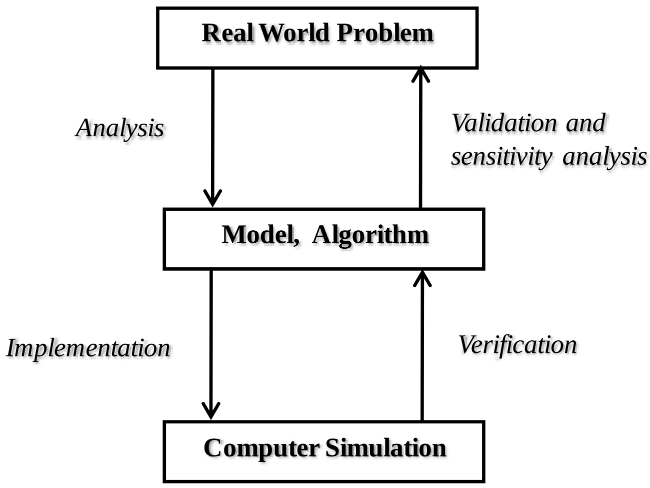

Figure 2.2: Optimisation as a process

The first phase (analysis), extracts the essential and relevant elements from a complex and detailed real world problem to create a model. Through implementation, an algorithm (solution technique) is formulated and applied to the model. This phase constructs the actual programming codes for computer simulation, from where results are generated. Verification takes computer simulation back to model/algorithm, which aims at ensuring that the simulation results are correct. Finally, in moving from the model and algorithm to the real world problem, validation and sensitivity analysis considers the appropriateness of the results, the need for model modification or choice of another solution algorithm. While validation ensures that the model or solution technique is appropriate for the real situation, sensitivity analysis considers the effects of small changes in the data on the results.

2.1.4 Optimisation as a Problem Solving Technique

Optimisation is a problem solving technique; it applies to problems that exist in all fields of human endeavour. An optimisation problem exists when there is disparity between a present state and a desired state [16]. Solutions to problems are ways of allocating the available resources so as to reduce the disparity, with the aim of transforming the former into the latter. A good problem solving technique requires identification and consideration of the stakeholders in the problem solving approach, the objectives to be achieved, the variables involved in the problem, the constraints (if they exist), solution methods, and evaluation of the final solution.

Economic load dispatch is an optimisation task, aimed at determining the best/optimal combination of power generators' outputs that has the lowest generation cost, while minimising under/over generation, and helping electricity generating stations meet their customers' demands. Each of these objectives is equally important, and there is no additional knowledge regarding the problem. It is assumed that a potential solution to this problem could be described by means of *decision vector* $(x_1, x_2, ..., x_n)$ in a decision space X. A function $f : X \rightarrow Y$ evaluates the quality of a specific solution by assigning it an *objective vector* $(y_1, y_2, ..., y_n)$ in the *objective space* Y. Assuming that the objective space is a subset of the real numbers, that is, $Y \subset \Re^n$ and that it is required from the optimisation to maximize the single objective; in such an objective optimisation, a solution $x^{(1)} \in X$ is better than another $x^{(2)} \in X$ if and only if $y^{(1)} > y^{(2)}$ where $y^{(1)} = f(x^{(1)})$ and $y^{(2)} = f(x^{(2)})$. Although there may be several potential solutions in decision space, with all of them mapped to the same objective, there is only one optimal solution [17].

2.1.5 Multi-Objective Optimisation Problems

A Multi-Objective Optimisation Problem (MOOP) seeks to minimise *k* components of a vector function *f*, with respect to a vector variable $x = (x_1, x_2, ..., x_k)$ in a universe, *U*. According to Osyczka (1985) [18], MOOP is:

> *"A vector of decision variables which satisfies constraints and optimises a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. It means finding a solution which would give the values of all the objective functions acceptable to the designer".*

In mathematical notation, a MOOP can be generally defined as:

$$Min\ f(x) = \begin{bmatrix} f1(x) \\ f2(x) \\ . \\ . \\ fk(x) \end{bmatrix} \tag{2.1}$$

MOOPs do not in general lead to single solutions, but a set of solutions, with the solutions having a good balance among the objectives. In realising this balance, a notion of optimality is required. The most commonly adopted notion of optimality, called Edgeworth-Pareto optimality (or simply Pareto optimality) [19], was originally introduced by Francis Y. Edgeworth in 1881, and later generalised by Vilfred Pareto in 1896. A solution to a MOOP is said to be Pareto optimal if and only if there is no other solution that would reduce some criteria without producing a corresponding increase in any of the other criteria. The corresponding mapped points in objective space are usually referred to as non-dominated solutions [19]. The use of this concept therefore produces sets of solutions called the Pareto optimal sets, and not single solutions. The variables contained in their solutions sets are said to be non-dominated, and the plot of objective functions containing such non-dominated variables is called a Pareto front.

2.2 Electrical Energy and Power

Energy and power are two different terms that are most often mixed up in usage. It is therefore necessary to distinguish between the two terms as used in this work.

Energy is the capability to do work. It is an indispensable tool for socio-economic growth and development of any nation, and available in two forms: renewable and non-renewable energy [3]. Renewable energy is derived from natural resources such as: sun, wind, water, etc, which are naturally replenished; while non-renewable energy is derived from fossil fuels (coal, petroleum, natural gas) and nuclear energy. Power on the other hand, is the rate of doing work. It is the work done or energy expended in a unit time. Electrical power is the rate of flow of electrical energy.

The world's energy demand is on the increase and to meet up with this trend, appropriate and timely up to date studies need to be constantly carried out. The electrical energy industry is the largest energy industry in the world, as the operations of other industries and sectors depend on electricity [20]. All aspects of any nation's economy depend on electricity, and both developed as well as developing countries in the world aim to establish and maintain an effective electricity sector. Electrical Power Engineers are constantly engaged with every step in the process of generation, transmission, distribution and utilisation of electrical energy, and they encounter challenging problems in their quest to deliver increasing amounts of electrical energy in an economical manner [21, 22]. Analysis and computation of the flow of electrical power is a systematic procedure that involves not only the electricity grid (generation**,** transmission and distribution of electric power), but also the devices connected to the systems such as generators, motors and transformers; and evaluation of the system's performance. A typical power system, identifying the problem domain is shown in Figure 2.3:

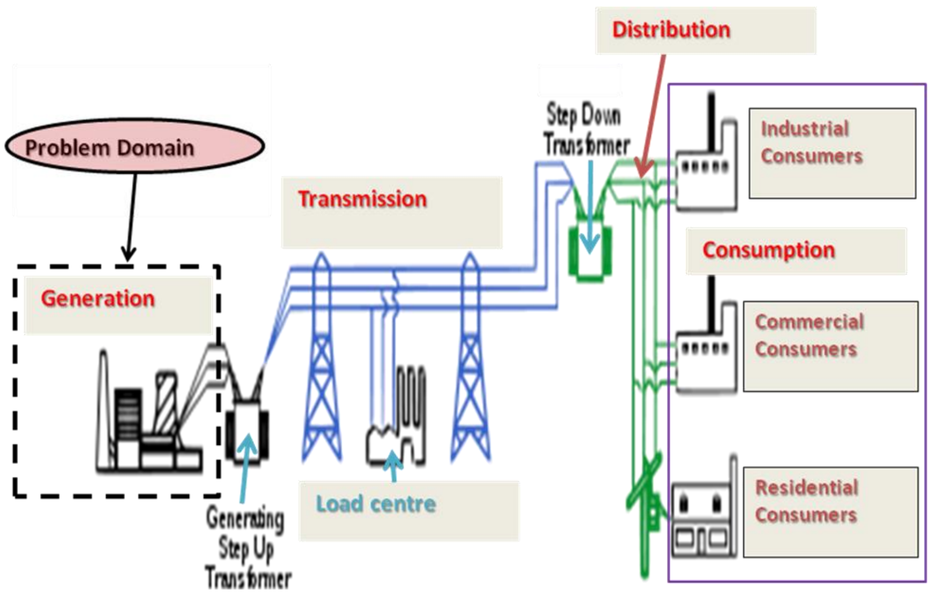

Figure 2.3: A typical power system representation showing the problem domain

No two electrical power systems are exactly the same, but they share a range of common characteristic features. As depicted in Figure 2.3, power generation is achieved by means of generators (synchronous machines driven by turbines). The generated power is conveyed from the power stations over long distances by means of transmission network through load centres to distribution network, and finally to the consumers. Electrical power is generated, transmitted and distributed by 3-phase alternating current systems, with voltage and frequency levels required to remain within tolerance levels to ensure high quality output.

Interests in power system studies date back to the works of William Gilbert, George Ohm, Michael Faraday and James Clerk Maxwell. William Gilbert is regarded as the father of electricity and magnetism and also the first Electrical Engineer; George Ohm quantified the relationship between the electric current and voltage; Michael Faraday discovered the principle of electromagnetic induction that explains the operation of generators and transformers; James C. Maxwell published a unified theory of electricity and magnetism [23]. The introduction of computers subsequently facilitated the adoption of a very wide range of approaches in the analysis and research studies for optimum performance in electrical systems.

2.2.1 Power Generation

This is the process of obtaining electricity from the source. Electricity is generated at power stations by electro-mechanical generators driven by heat engines, fuelled by chemical combustion, nuclear fission, and kinetic energy of flowing water, wind, solar photovoltaic and geothermal power. There are two forms of power generation: centralised and distributed generation. Centralised generation provides good economies of scale, but leads to power losses through the transmission of electricity over long distances. Distributed generation, also known as on-site, dispersed, embedded, or decentralised generation, generates electricity from many small energy sources [24]. It reduces the amount of energy lost in transmission, as well as size and number of power lines that must be constructed. Distributed power sources includes: Solar panels, wind turbines, natural gas-fired micro-turbines, photo-voltaic sources, etc. The major problem with distributed generation is high cost. The distributed generation system is very unconventional, even though it has numerous advantages over the traditional method of burning fossil. Generation of electricity from fossil fuels releases hydrocarbon emission which is a very serious pollutant to the environment.

2.2.2 Power Transmission

Power transmission is the transfer of electrical power from generating stations to sub-stations located in the neighbourhood of the demand or consumers. When interconnected, the transmission lines become high voltage transmission networks (power grids) [24]. Electrical power is transmitted at high voltages to reduce the energy lost in long distance transmission. Transmission is done through overhead power lines, as well as underground power cables. A transmission substation decreases the voltage of incoming electricity, allowing it to connect from long distance high voltage transmission, to local lower voltage distribution. Transmission efficiency is improved by increasing the voltage using a step-up transformer, which reduces the current in the conductors, while balancing the power transmitted with the power generated.

2.2.3 Power Distribution and Consumption

Electricity distribution is the final stage in the delivery of electricity to end users, through a distribution network. There are two types of distribution networks: Radial and interconnected networks [24]. A radial network leaves the station and passes through the network area with no normal connection to any other supply, typical of long rural lines with isolated load areas; while an interconnected network has multiple connections to other points of supply, found in urban areas. The benefit of the latter is that in the event of a fault or maintenance, a small area of network can be isolated and the remainder kept on supply. Electricity has become more of a commodity, with China being both the largest producer as well as consumer, followed by USA. Tables 2.1 and 2.2 respectively show the world's top ten producers and consumers of electricity in 2012 [25].

S/N	*Country*	*Production (in TWh)*
1	China	4,926
2	USA	4,295
3	India	1,087
4	Russia	1,064
5	Japan	1,057
6	Canada	646
7	Germany	623
8	Brazil	561
9	France	559
10	South Korea	526

Table 2.1: World's top 10 electricity producers in 2012

S/N	*Country*	*Consumption (in TWh)*
1	China	4,281
2	USA	3,798
3	Japan	971
4	Russia	878
5	India	804
6	Germany	535
7	Canada	504
8	Brazil	498
9	South Korea	492
10	France	448

Table 2.2: World's top 10 electricity consumers in 2012

2.3 Optimal Power (Load) Flow

In order to improve the performance of an electrical power system, three basic things are required [20, 21, 22]:

1. The model of the system;
2. The objective function;
3. The optimisation approach to be adopted.

An optimisation problem is typically formulated as a mathematical model, aimed at minimising undesirable quantities or maximising desirable ones, subject to some constraints. In power system operation and planning, such undesirable quantities are: Cost of running the generating units, transmission loss, distribution errors, etc [26]. The desirable quantities include: Output, profit and efficiency. The model is represented by a set of linear or non-linear equations, describing the optimal or steady-state operation of the system. This could be formulated as minimising or maximising a scalar objective, *x* through the optimal control of a vector, *u*, of control parameters, stated mathematically as:

Minimise/Maximise: $$F(x,u) \tag{2.2}$$

Subject to:

$$g(x,u) = 0 \tag{2.3}$$

$$h(x,u) \leq 0 \tag{2.4}$$

Where:

g (x, u) = Set of non-linear equality constraints (power balance);

h (x, u) = Set of inequality constraints (limits in generators outputs);

x = Vector of dependent variables (generating cost);

u = A vector of control variables (power output).

Common objectives in a power system optimisation are: minimising costs, minimising under/over generated power output, minimising gaseous emissions from thermal plants (oxides nitrogen, sulphur and carbon dioxide), etc [27, 28, 29, 30, 31]. Equality constraints include: generated power balance, while inequality constraints include: generators' output limits, ramp-rate limits, prohibited operating zones, security and emission constraints [32, 33, 34, 35]. Algorithms used to achieve the above objectives are called objective functions.

Power system optimisation is primarily aimed at achieving optimality in power output. Hence, power system problems are also called optimal power flow (OPF) problems [36, 37]. There are two major phases involved in OPF problems. The first is unit commitment (UC), while the second is economic load dispatch (ELD).

2.3.1 Unit Commitment (UC)

Unit commitment (pre-dispatch) determines the unit start-up and shut-down schedules, with the aim of minimising system expenditure [21]. It selects optimally out of the available generating sources to operate to meet the expected load. As a generation resources management operation scheduling task, unit commitment lies between economic dispatch and production/maintenance activities in an hourly decision time interval during a one-day or one-week schedule period.

Unit commitment schedules the ON and OFF times of generating units, and calculates the minimum cost of hourly generation schedules, considering the start up/down rates and minimum up/down times. The schedule takes the following factors into consideration: unit operating constraints and costs, generation and reserve constraints, as well as plant start-up and network constraints [21]. The overall objective function of unit commitment problems in the sum of the fuel costs of all the generating units over time. This is mathematically represented by (2.5), subject to the conditions in (2.6) to (2.8). The units are expected to maintain a given amount of power reserve, P_{Ri}.

$$F_T = \sum_{t=1}^{T}\sum_{i=1}^{N}\left[u_{i,t}.F_i(P_{i,t}) + SU_{i,t} + SD_{i,t}\right] \tag{2.5}$$

Subject to:

$$\sum_{i=1}^{N} P_{i,t} \geq P_{D,t} + P_{R,t} \tag{2.6}$$

$$P_{i,\max} = P_{i,\max} - P_{R,i} \tag{2.7}$$

$$P_{D,t} + P_{L,t} \leq \sum_{i=1}^{N} P_{i,t} - \sum_{i=1}^{N} P_{R,t} \tag{2.8}$$

Where:

F_T	=	*Total operating cost;*
$u_{i,t}$	=	*Commitment state of the i^{th} unit at hour t;*
F_i	=	*Fuel cost of the i^{th} unit at hour t;*
P_i	=	*Power output of the i^{th} unit at hour t;*
$SU_{i,t}$	=	*Start-up cost of the i^{th} unit at hour t;*
$SD_{i,t}$	=	*Shut-down cost of the i^{th} unit at hour t;*
$P_{D,t}$	=	*Power demand at hour t;*
$P_{R,t}$	=	*Spinning reserve at hour t;*
$P_{L,t}$	=	*Power loss at hour t;*
N	=	*Total number of generating units.*

Solution approaches that have been proposed to solving unit commitment problems include: priority list [21], Lagrangian relaxation [21], dynamic programming [21], evolutionary algorithm [38, 39], particle swarm optimisation [40]. Priory list is a fast method, but gives schedules with higher operating cost [41]. Lagrangian relaxation method is thought to be the best optimiser for large scale unit commitment problems, but there is no guarantee for optimal solution [12]. The rest of the approaches are considered in ELD, as they have common applications to both tasks. UC technique is also utilised in DELD.

2.3.2 Economic Load Dispatch (ELD)

Economic load dispatch (also called online economic dispatch) allocates, distributes or "dispatches" system load demands to the various generating units with the aim of minimising power generation cost [21]. Consider the operating cost of a thermal power plant; Figure 2.4 is a unit's input-output curve of the plant, known as heat-rate curve [42].

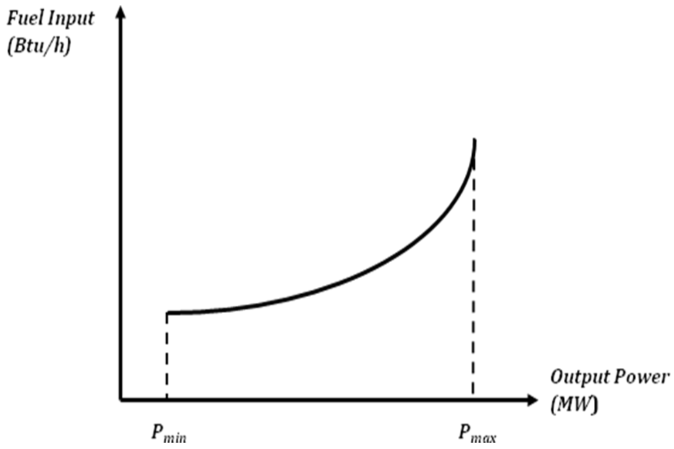

Figure 2.4: Heat-rate curve

Input to the plant is measured in British thermal unit per hour (Btu/h), while output is measured in Megawatt (MW). Converting the input of the heat-rate curve from Btu/h to $/h produces the fuel-cost curve, shown in Figure 2.5, with the cost mathematically defined in (2.9), and (2.10) shows the derivative of the fuel-cost against output power.

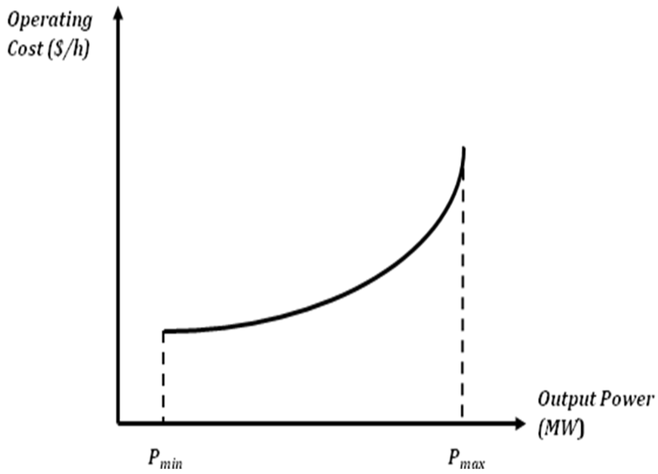

Figure 2.5: Fuel cost curve

$$C_i = a_i P_i^2 + b_i P_i + c_i \quad (\$/h) \tag{2.9}$$

$$\frac{dC_i}{dP_i} = 2a_i P_i + b_i \quad (\$/MWh) \tag{2.10}$$

A plot of (2.10) gives the incremental fuel-cost curve, shown in Figure 2.6:

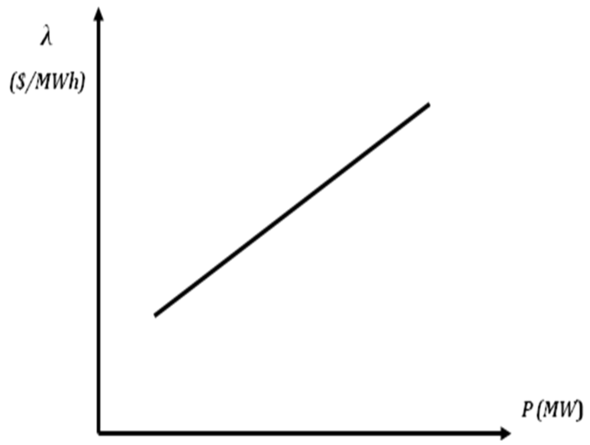

Figure 2.6: Incremental fuel cost curve

The incremental fuel-cost curve measures the cost of producing the next power increment. To illustrate the effect of incremental fuel-cost curve in sharing power generation between generators to minimise total fuel cost, consider a generating plant with two units as shown in Figure 2.7:

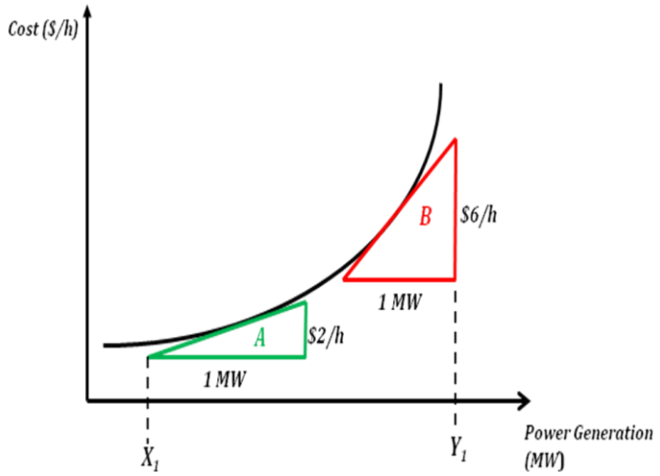

Figure 2.7: Illustration of Economic Load Dispatch with 2 generating units

Increasing power generation by 1MW in Unit A costs $2 per hour, whereas there is a cost saving of $6 per hour in reducing generation by 1MW in Unit B. To run the plant economically, generators with lower costs should be run more often. In the above case, it will be more economical to shift the production of Units *A* and *B* from X_1 and Y_1 to X_2 and Y_2, as shown in Figure 2.8, with net savings of $4 per hour. In a system with many units, the shift is repeated until incremental costs are all equal. Power production is shifted from the more expensive units (B in this case) to the less expensive units (A in this case). This is the basis for economic load dispatch.

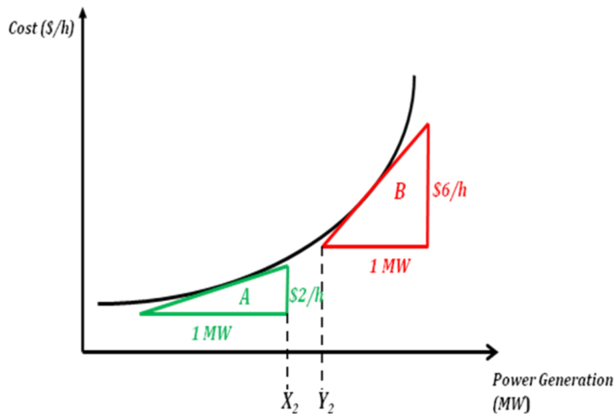

Figure 2.8: Shifting generation in units to minimise total fuel costs

2.4 Formulation of ELD Problems

In the research literature that has investigated ELD problems, various formulations have been studied that invariably make several simplifying assumptions. This has typically involved the following scenarios, which are elaborated in the remainder of this subsection:

1. Smooth fuel cost functions.
   - Neglecting both generators' limits and power loses;
   - Neglecting power loses but considering generators' limits;
   - Considering both generators' limits and power loses;
2. Non-smooth fuel cost functions.
3. Multiple fuels.
4. Ramp-rate limits.
5. Prohibited operating zones.

2.4.1 Smooth Fuel Cost functions

Traditionally, ELD problems are assumed to have quadratic, convex but smooth cost functions. Consider a system of three generators *G1*, *G2* and *G3*; delivering powers Pg_1, Pg_2 and Pg_3 respectively, connected to a common bus, as shown in Figure 2.9. P_D is the total powers demend.

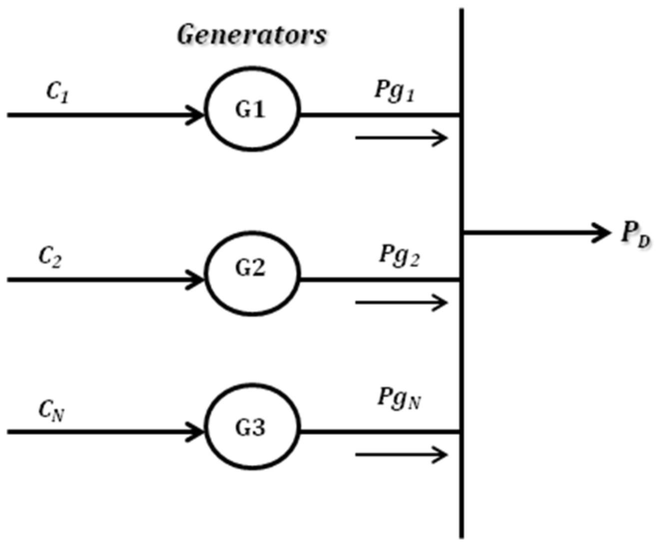

Figure 2.9: Generating units connected to a common bus

Each unit has its own cost function, C_i. The task here is to find the combination of the real power generation for all the units such that the total generation cost, C_T is minimised.

$$Min\, C_T = \sum_{i=1}^{N} C_i \tag{2.11}$$

$$= \sum_{i=1}^{N} a_i Pg_i^2 + b_i Pg_i + c_i \tag{2.12}$$

Subject to:

$$\sum_{i=1}^{N} Pg_i = P_D \quad \forall i = 1, 2, ..., N \tag{2.13}$$

This condition holds true for the first scenario where both generators' limits and power loses are neglected. This is the case of generating units within a power plant, or when different plants are physically located close to each other. Considering generators' limits (second scenario), the real power generated by each generator must be between minimum and maximum defined capacities, represented by the following inequality constraints.

$$Pg_{i,\min} \le Pg_i \le Pg_{i,\max} \qquad \forall i = 1, 2, ..., N \tag{2.14}$$

If the generating plants are not physically located very close to each other, then network transmission loses need to be considered. Figure 2.10 is the schematic illustration of such system.

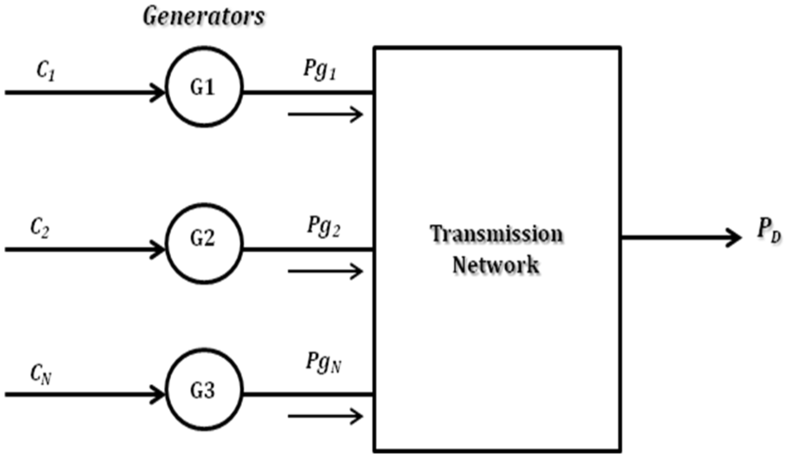

Figure 2.10: Generating units connected through transmission network

The total power generated must be balanced, and equal to the sum of total power demand and power loss.

$$\sum_{i=1}^{N} Pg_i = P_D + P_L \qquad \forall i = 1, 2, \ldots, N \tag{2.15}$$

The power loss is a function of generating units, whose calculation is by means of B-coefficient/matrix method. The loss formula, which expresses the total transmission losses in terms of source powers, was first derived by E. E. George in 1943 as [42]:

$$\begin{aligned} P_L &= B_{11}Pg_1^2 + B_{22}Pg_2^2 + B_{33}Pg_3^2 + \ldots + B_{nn}Pg_n^2 \\ &\quad + 2B_{12}Pg_1Pg_2 + 2B_{13}Pg_1Pg_3 + \ldots \\ &\quad + 2B_{23}Pg_2Pg_3 + \ldots 2B_{mn}Pg_mPg_n \\ &= \sum_m \sum_n Pg_m B_{mn} Pg_n \end{aligned} \tag{2.16}$$

Where:

Pg_m, Pg_n	=	*Source loadings;*
B_{mn}	=	*Transmission loss formula coefficients.*

The determination of the B_{mn} coefficients was a very tedious and rigorous procedure, even for a system of few generators. In 1950, J. B. Ward, J. R. Eaton and H. W. Hale described an application of network analyser developed for the loss formula determination [42]. But as the number of generators gradually increased, the mathematical calculations involved were drastically very large, which posed a great challenge. In 1951, G. Kron, working with G. W. Stagg and L. K. Kirchmayer improved and simplified the formulation in terms of the quadratic, linear and constant terms. The resulting formula, named after Kron was expressed as [42]:

$$P_L = \sum_m \sum_n Pg_m B_{mn} Pg_n + \sum_n B_{n0} Pg_n + B_{00} \tag{2.17}$$

For a system of N generators with a square matrix (dimension the same as N), the Kron's formula, also known as the B-matrix loss formula [43], is simplified as [54]:

$$P_L = \sum_{i=1}^{N} \sum_{j=1}^{N} Pg_i B_{ij} Pg_j + \sum_{i=1}^{N} B_{0i} Pg_i + B_{00} \quad \forall i,\ j = 1, 2, ..., N \tag{2.18}$$

Where C_T it the total generation cost; C_i is the generation cost of unit, i; Pg_i is the power generation of unit, i; N is the number of generating units; a_i, b_i, c_i are the cost coefficients of unit, i; P_D is the power (load) demand; P_L is the power loss; B_{ij} is the ij^{th} element of the loss coefficient square matrix of the same dimension as Pg_i, B_{0i} is the i^{th} element of the loss coefficient vector of the same length as Pg_i, B_{00} is the loss coefficient constant, Pg_i and Pg_j are the active power generation of units i and j respectively.

2.4.2 Non-Smooth Fuel Cost functions

In real world scenarios, larger generators, especially those with multi-valve steam turbine do not have smooth fuel cost functions. The ripples that result from the valve-points make their non-convex cost functions exhibit greater variations. Figure 2.11 shows a cost curve with five valve-points.

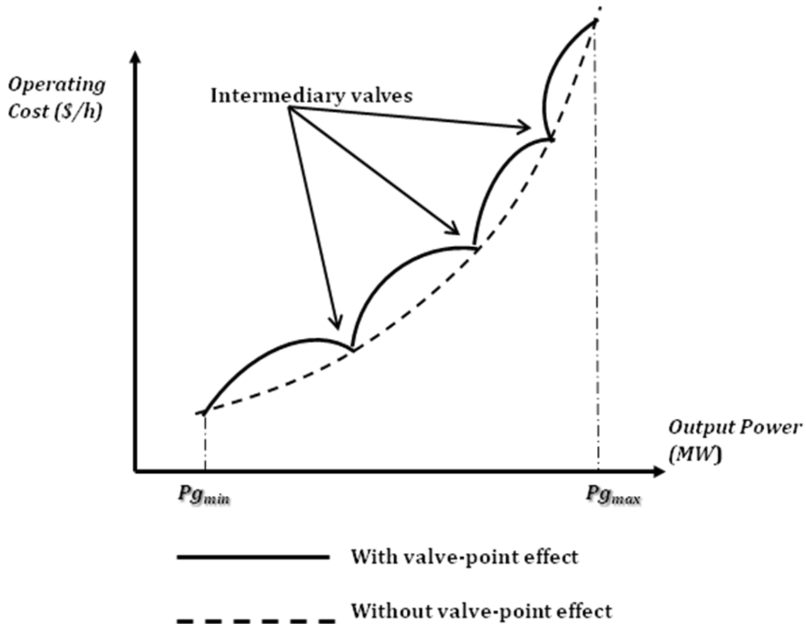

Figure 2.11: Generator cost curve with valve-points

The overall cost function is therefore modified with the incorporation of sinusoidal function to the initial quadratic function [44, 45, 46]:

$$C_i = a_i Pg_i^2 + b_i Pg_i + c_i + \left| e_i Sin\left(f_i \left(Pg_{i,\min} - Pg_i \right) \right) \right| \tag{2.19}$$

Where e_i *and* f_i are the fuel cost coefficients of i^{th} unit with valve-point effects. The complexity of realising an optimum solution in this formulation is increased by the presence of the sinusoidal component of the cost function which models the valve-point effects, creating ripples in the cost curve, thereby populating the search space with a number of local minima.

2.4.3 Multiple Fuels

This is a more practical representation of generating units in a real life power system. Various types of generating plants with multiple fuel sources (e.g. coal, natural gas, oil, etc) are available, each with different numbers of generating units. The overall cost function is expressed as several piecewise quadratic functions, reflecting the effects of the fuel type, with the generators identifying the most economic fuel to burn while in operation [44, 45]. This is referred to as a hybrid cost function [44]. For a generator with multiple fuels, the incremental cost functions are represented as shown in Figure

2.12, while the ELD problems with both multiple fuels and valve-point effects are modelled as shown in (2.20).

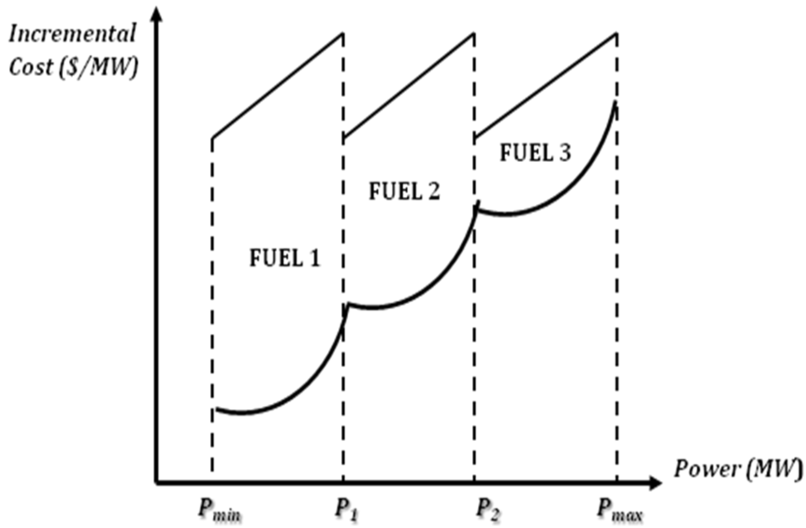

Figure 2.12: Incremental cost function of a generator with multi-fuel options

$$C_i = \begin{cases} a_{i1}Pg_i^2 + b_{i1}Pg_i + c_{i1} + \left| e_{i1}Sin\left(f_{i1}(Pg_{i1,\min} - Pg_{i1})\right)\right|, & for\ fuel\ 1, \quad Pg_{i,\min} \le Pg_i \le Pg_{i1} \\ a_{i2}Pg_i^2 + b_{i2}Pg_i + c_{i2} + \left| e_{i2}Sin\left(f_{i2}(Pg_{i2,\min} - Pg_{i2})\right)\right|, & for\ fuel\ 2, \quad Pg_{i1} \le Pg_i \le Pg_{i2} \\ \ldots & \\ a_{ik}Pg_i^2 + b_{ik}Pg_i + c_{ik} + \left| e_{ik}Sin\left(f_{ik}(Pg_{ik,\min} - Pg_{ik})\right)\right|, & for\ fuel\ k, \quad Pg_{i,k-1} \le Pg_i \le Pg_{i,\max} \end{cases} \tag{2.20}$$

Where a_{ik}, b_{ik} *and* c_{ik} are cost coefficients of i^{th} unit for fuel type k, where k = 1, 2… k, and e_i *and* f_i are cost coefficients of i^{th} unit with valve-point effects.

2.4.4 Ramp-Rates Limit

The generating units' ramp-rates limit restricts their operating range in a given generation period. A ramp-rate is the rate of change in output from a power plant. Usually based on a manufacturer's specification, it is the amount of load added to the generating plant to control the amount of load per unit time that can be added to the generator to keep it from being "overloaded". Based on these rates, the subsequent outputs of the units should be within their ranges, with (2.14) modified as (2.21) [34]:

$$\max(P_{i\min}, P_i^0 - DR_i) \le P_i \le \min(P_{i\max}, P_i^0 + UR_i) \tag{2.21}$$

Where P_i^0 is the previous output, P_i is the present output, DR_i and UR_i are the down and up ramp rate limits (in MW/h).

2.4.5 Prohibited Operating Zones

Besides each generator having operating capacity which cannot be exceeded at any time, a typical thermal unit may have a steam valve whose vibration in the shaft bearing during operation may result in interference and a discontinued fuel cost characteristic [47]. Also, mechanical faults might introduce restrictions in some plants' components, and the operating range of the whole unit may not be available for load allocation. These disjoined sections of the cost-output curve between the minimum and maximum power outputs are called prohibited zones, as shown in Figure 2.13 [79]. Practical operations of the power plant involve adjusting the power output of the units such that they must be outside the prohibited zones.

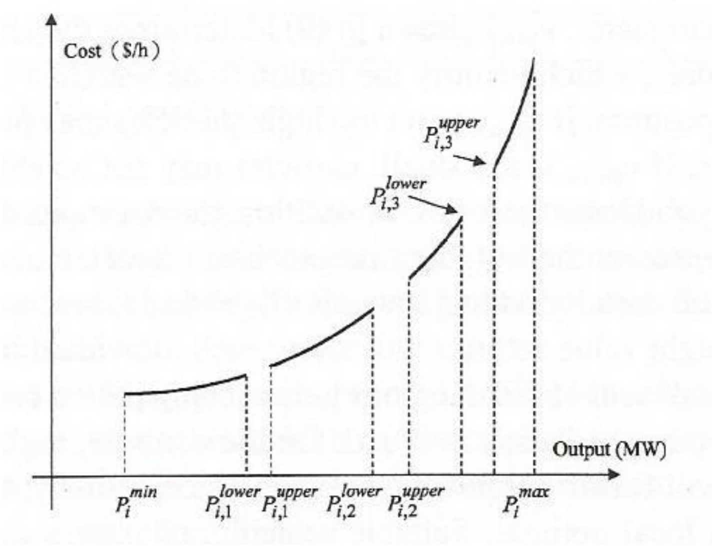

Figure 2.13: Prohibited operating zones and generation limits for a generator [79]

The feasible operating zones are described by the following inequality constraints:

$$\begin{aligned} &Pg_i^{\min} \le Pg_i \le Pg_{i,1}^{l} \\ &Pg_{i,k-1}^{u} \le Pg_i \le Pg_{i,k}^{l}, \quad \forall k = 2,3,\ldots,zo_i \\ &Pg_{i,zo_i}^{u} \le Pg_i \le Pg_i^{\max} \end{aligned} \tag{2.22}$$

Where $Pg_{i,k}^{l}$ and $Pg_{i,k}^{u}$ are the lower and upper bounds of the k^{th} prohibited zone of unit i, respectively, k is the index of prohibited zones, zo_i.

2.5 Types of ELD Problems

Generally, two types or categories of ELD problems exist in the literature. They are static economic load dispatch (SELD) and dynamic economic load dispatch (DELD) problems [40, 51]. SELD problems handle only a single load level. The variables (generator outputs) do not vary with time. Details of SELD implementation are described in Chapter 3. DELD, also known as optimal dynamic dispatch (ODD) [40], deals with variations of load demands due to the dynamic nature of power systems. The initial formulation of ODD problems was as an optimal control problem, which modelled the power system generation using state equations, where the state variables were the generators outputs, and the control inputs were the generators' characteristics. This produced the generator trajectory as optimal output for a given initial generation. Problem cases involving DELD are described in Chapter 4.

2.6 Review of Solution Approaches to Power System Optimisation

Current problems facing development of solution algorithms to power system optimisation problems include, but are not limited to: increasing the size of the power system, on-line applications of automatic control, and the need to achieve optimal output. Several contributions have been made to tackle these problems. It is worthy of note however, that no single method of the existing solutions meets all the desirable requirements of an ideal solution. In any particular situation, choice of any method is a good compromise between the various capabilities of the existing methods [49]. Power system optimisation problems have the following characteristics [20, 21, 22]:

1. They are formulated on systems operating at steady state conditions, with equations consisting of a set of non-linear equations to be solved simultaneously;
2. Solutions to these equations are not unique;
3. The numbers of unknown variables in the equations are usually more than the number of equations.

A good solution algorithm must have properties from a combination of some of the following features: high speed, low storage requirements, reliability, versatility in handling various adjustments, simplicity in programming and high-degree of accuracy. The approaches fall under three categories:

1. Mathematical programming methods;
2. Artificial intelligent methods;
3. Hybrid methods.

2.6.1 Mathematical Programming Methods

Matrix algebra was the earliest analogue mathematical optimisation method used in power system studies. One of these was the bus admittance matrix method [50]. The method enjoyed simplicity, ease of formulation and modification, but very tedious and time consuming, and the results obtained were prone to unavoidable human errors, as well as fatigue on the part of the researcher. Most researches concentrated only on power problems involving very small equations whose matrices can be conveniently handled manually [20]. With the advent of digital computers, the solutions to power problems were reduced to sets of algorithms solved by computers. The first of these methods was the Gauss-Seidel iterative algorithm using the nodal-admittance matrix method, with the first successful computer program developed by Ward and Hale in 1956 [20]. Like the bus admittance matrix method, the Gauss-Seidel iterative algorithm was simple in structure, with lower memory requirement and less computational time per iteration. But the major problem was the slow rate of convergence, therefore large number of iterations are needed, especially in large systems where the algorithm was discovered to be obsolete, as the number of iterations required for a solution was found to be relatively high. The need to study power problems for large systems became necessary with the increase in high voltage interconnections between systems. A more successful algorithm and the most universally acceptable replacement for the Gauss-Seidel method was the Newton-Raphson method [20]. It was found to be very suitable for large-scale power systems, with more degree of accuracy, convergence after a few iterations, which are independent on the system size. However, it was a difficult solution technique, as calculations were complex, hence more computer time per iteration was involved, and large computer memory required.

Other traditional mathematical optimisation methods include: Linear and non-linear programming, gradient-based method, dynamic programming, interior point method, Lambda iteration [20, 40, 51], fast-decoupled algorithms [49], and integer programming [52]. In most of these algorithms, optimality of solutions was mathematically formulated, and could be applied to large-scale problems. They have no problem-specific parameters, and most of them have high computational efficiency, with ease of implementation. However, solutions obtained using them have their inherent implementation limitations. The solutions for large-scale systems are not very simple. Many of the techniques fail to get optimal solutions, with a possibility of being stuck in local optima [51]. Linear programming encounters poor computation efficiency; while dynamic programming suffers from the curse of dimensionality, a process whereby the dimensions of the ELD problem becomes too large that it requires massive computational effort [26, 51]. These problems affect their application to practical generator problems with ramp rates, valve-point effects and prohibited operating zones constraints. Hence, solutions to ELD problems through mathematical formulation have minimal applications.

2.6.2 Artificial/Computational Intelligent Methods

Despite the successes of mathematical programming methods, the identified problems have pushed developments in ELD optimisation in the recent decades from the traditional iterative methods to stochastic search methods. Most power problems require the use of facilities to store human knowledge, operators' judgment/experience, and variations in load and network uncertainties [26]. Artificial intelligence is defined as the intelligence of machines or software systems. A leading and successful approach to artificial intelligence is the computational intelligence. As a scientific as well as engineering discipline, the purpose (science) of computational intelligence is the understanding of underlying principles behind intelligent agents; while the methodology (engineering) involves the design, implementation and experimentation with systems which perform tasks that are considered intelligent [82]. Intelligent agent senses its environment, taking action that increases its success chances. The activities of these agents that are considered intelligent include: knowledge, reasoning, learning, planning, communication, perception, manipulation, etc. Computational intelligent algorithms are of two broad categories: *machine learning algorithms* and *nature-inspired algorithms*.

2.6.2.1 Machine Learning Algorithms

Machine learning deals with the study and design of systems that have the capability of learning from data. The systems work by prediction, based on previously known properties learned from the data. The most common and widely used machine learning algorithms in solving ELD problems are: (1) Artificial Neural Networks, (2) Fuzzy Logic, (3) Expert Systems, (4) Bayesian Networks, and (5) Support Vector Machines.

(1) Artificial Neural Networks

Artificial Neural Networks (ANN), inspired by the work of Warren McCulloch and Walter Pitts (a logical calculus of the ideas imminent in Nervous Activity) [176], models the construction and behaviour of the human brain using man made technology. An ANN is a layered network of artificial neurons, working in analogy to biological neurons. The advantages of ANNs that facilitate its usage are speed, robustness, ability to learn, ability to adapt to training data, and appropriateness for non-linear modelling. However, for complex problems ANNs need to have many inputs and/or several layers of inputs; with consequently many parameters, much time is required for training, and good results are far from guaranteed. ANN has been applied to load forecasting, voltage security, monitoring and control, unit commitment, economic load dispatch, fault diagnosis, security assessment and power system stability in [65]. The Hopfield neural network approach was developed for the ELD in [56], involving 20 generating units, generator limits, power balance, and transmission loss.

(2) Fuzzy Logic

Fuzzy Logic (FL), developed by Lotfi A. Zadeh in 1964 [69], is a superset of Boolean logic, and handles the concept of partial truth. It provides rules and functions that permitted natural language queries, a means of calculating intermediate values between absolute TRUE and absolute FALSE, with resulting values between 0 and 1. FL aims to address uncertainty and imprecision which widely exist in the engineering problems and was first used in 1979 for solving power system problems, with application to voltage and reactive power control, load forecasting, fault diagnosis, power system protection/relaying, stability, and power system control [26]. Its application to active power generation is very limited, unless as a hybrid tool with other approaches.

(3) Expert Systems

Expert Systems, first proposed by Feigenbaum *et al* in 1971 [193], are knowledge-based methods, which uses the knowledge gained from application domain experts, available via a suitable rule-based interface, to solve problems that are difficult enough to require human expertise for their solution [26, 73]. The systems are permanent and consistent in their results. They can be easily be transferred, reproduced and documented. In the recent decades, expert system has been applied in the areas of power system planning, alarm processing, fault diagnosis, power system protection, reactive power/voltage control, as well as load, bid and price forecasting [26, 74]. However, expert system suffers knowledge bottleneck problem [73], as it does not learn or adapt to new situations.

(4) Bayesian Networks

A Bayesian network is a probabilistic graphical model whose nodes represent a set of random variables, connected by edges which represent conditional dependencies or relationships through a directed acyclic graph [194, 195]. Each node has a conditional probability table, which are learnt from a given network structure with independence assumptions. The number of parameters in the network is proportional to the directed edges in the network due to the independent assumptions. The Bayesian learning is a complex process that requires a thorough knowledge of the problem domain; it is intuitive in nature. Bayesian networks are widely used in power system components fault diagnosis and detection.

(5) Support Vector Machines

Originated from supervised learning system and statistical learning theory of Vapnik in 1998 [196], Support Vector Machine (SVM) is a relatively new machine learning technique based on the structural risk minimisation (SRM) where data are mapped from a given dimensional space into a higher one, and from this higher dimensional space, an optimal separating hyper-plane is constructed to solving quadratic programming problems [73]. Under the SRM principle, the SVM approach reduces the number of learning operations and minimises the generalisation error on the test data set. SVM is a regression prediction/text classification tool that uses machine learning theory to maximise prediction accuracy while at the same time avoiding data that does fall in the particular classifying zone. SVMs are generally used in electrical power systems load forecasting and prediction. In [197], SVM was used to optimise voltage level between buses, where it converged generator voltages and angles at optimum minima, and also proved effective in adjusting control variables for power system restoration after a blackout.

2.6.2.2 Nature-Inspired Algorithms

These are algorithms developed to solving optimisation problems by drawing inspiration from nature [198]. These algorithms are classified as: Biological-inspired (Bio-inspired), Physics and Chemistry-based algorithms [199, 200]. Bio-inspired algorithms are population-based algorithms which imitate biological processes (evolution and natural hereditary). Two classes of these algorithms are: Evolutionary Algorithm (EA) and Swarm Intelligence (SI). Physics and Chemistry-based algorithms are developed by drawing inspiration from some physical or chemical laws. Common examples are Simulated Annealing (SA) and Harmony Search (HS). Recently, Memetic Algorithms (MA) [201], have been developed for solving ELD problems, most of which combine the evolutionary nature of EAs with the swarm behaviour of SIs. Examples include: Artificial Immune Systems, Shuffled Frog Leaping Algorithm, Group Search Optimiser and Biogeography-Based Optimisation.

(1) Evolutionary Algorithm

This is a non-deterministic, nature-inspired, population-based optimisation approach which mimics the process of biological evolution, and bear direct analogy to Charles Darwin's principle of natural selection/theory of survival of the fittest. It is used in solving NP-hard (**N**on-deterministic **P**olynomial-time hard) problems. There are several varieties of Evolutionary Algorithm (EA) based on alternative solution structures, search operators and implementation aspects. Common examples are: Genetic Algorithm, Evolutionary Programming, Genetic Programming, Evolution Strategies, Differential Evolution, and Estimation of Distribution Algorithms.

(a) Genetic Algorithm

Developed by John Holland in the 1960s in USA [48], Genetic Algorithm (GA) is motivated by nature's wisdom and evolutionist theory, based on the concept of natural selection and origin of species [63]. Naturally, stronger species reproduce more, and pass on their genetic structure to their future generations, while the weaker ones face the problem of extinction. GA can be considered as the basis for all other evolutionary algorithms, hence a detailed description is made under the subject of evolutionary algorithm in a later section of this chapter. Common applications of GA to power system optimisation are: transmission expansion planning, power control, unit commitment, economic dispatch, and hydrothermal scheduling [26]. GA remains a major and growing area of active research in EA and the entire field of computational intelligence [68].

(b) Evolutionary Programming

First conceived by Lawrence J. Fogel in the early 1960s [202], Evolutionary Programming (EP) is a mutation-based EA (genetic linkage between parents and their offspring), initially applied to discrete search spaces. The structure of the chromosome is fixed as in GA, while the parameters are allowed to gradually evolve. The approach was extended to real-parameter optimisation problems by David Fogel [2]. EP relies mostly on mutation and selection (without crossover). The major disadvantage of EP is its slow convergence. In [167], several variants of EP were developed to solve the SELD problem with non-smooth cost function, including: Classical EP (with Gaussian mutation), Cauchy-mutation-based EP, mean of Guassian and Cauchy mutations, Cauchy-mutation with empirical learning rate, mean of Guassian and Cauchy mutations with empirical learning rate. The effectiveness of these approaches was verified with a large-scale test system of 40 generators.

(c) Genetic Programming

Genetic Programming (GP), named by John Koza in 1990 [77], is a specialisation of GA, but where each individual is a computer program that performs a user-defined task. While GA works on fixed-length chromosomes, GP manipulates variable length structures. Originally, GA was mainly used in solving relatively simple problems due to its computational intensive nature; and in EA to evolve computer programs, represented in the memory as a function tree. But with improvements in the technology, its usage was extended to quantum computing, sorting, searching, games theory and electronic design, but with minimal applications to electrical power system problems. Sean Luke *et al* [203], developed and ECJ (Evolutionary Computation in Java), an EA/GP project toolkit which consists of several packages - classes, methods and objects that are designed to carry out several activities, specified in parameters files.

(d) Evolution Strategy

Initially proposed in the early 1960s and further developed in the 1970s by P. Bienert, I. Rechenberg and H. P. Schwefel [2], Evolution Strategy (ES) also employs real-coded variables, relying on mutation and selection as its search operators, with a population size of “one” [73]. Although ES shares a lot of similarities with GA, however, ES operates only on floating point vectors, uses mutation as a primary operator, and emphasises more on behavioural (than genetic) link between parents and offspring. Efficient ES was used to solve a non-convex SELD problem in [65], and tested with systems of 6, 10, 15 and 40 generators. Their results were superior to those of GA and PSO with common data and the same number of fitness evaluations.

(e) Differential Evolution

Differential Evolution (DE) was first proposed by Storn and Price in 1995 [67], as an improved version of GA [35]. It uses few control parameters (which require minimum tuning, and remain fixed all through the process of optimisation. As in GA, DE makes use of three basic genetic operators: Mutation, Crossover and Selection; but like ES, it relies mainly on mutation to achieve its solutions. Mutation generates replica (mutant) vectors of the current population by introducing new parameters. Crossover combines the parameters of the mutant vector with those of a parent vector selected from the population to generate a trial vector. During selection, the trial vector competes with the parent vector, the better one progresses to the next generation. The advantages of DE include: ease of use, simple structure and implementation, fastness, and local searching property. But there is a disadvantageous tendency of being trapped in a local optimum or attaining premature convergence [109]. DE has a wide range of application to solving power system optimisation problems, including: SELD [35, 45, 54, 78], DELD [108, 109], and BBDELD [137].

(f) Estimation of Distribution Algorithm

Estimation of Distribution Algorithm (EDA), also known as Probabilistic Model Building Genetic Algorithm (PMBGA) [66], is motivated by the idea of discovering and exploiting interactions between variables in the solution. It estimates a probability distribution from population of solutions, and samples it to generate the next population. Through this, it builds probabilistic models to prevent disruption of solutions, as they move on from one generation to another. EDA does not only solve difficult problems, but also provides information on how the problem was previously solved [204]. This is a feature that makes the algorithm stand out from other optimisation techniques. At moment, power system application of EDA is very limited, especially in the area of controller design. In [205], adaptive population-based incremental learning (APBIL) was used to optimally tune power system stabilizer parameters.

(2) Swarm Intelligence

Swarm Intelligence (SI) is an computational intelligent algorithm that is inspired by natural swarm behaviour (of insects, birds and fishes) in developing solutions to problems [82]. SI originated from the study of colonies, or swarms of social organisms and consists of three main algorithms: Ant Colony Optimisation, Particle Swarm Optimisation, and Bee Colony Optimisation.

(a) Ant Colony Optimisation

Initially proposed by Marco Dorigo in 1992 [70], Ant Colony Optimisation (ACO) deals with artificial systems, taking inspiration from the behaviour of real ants (seeking optimal path between their colony and a source of food) used in solving discrete optimisation problems. The idea is that when one ant finds a good path (that is, the shortest path) from the colony to a food source, other ants are more likely to follow that path. This automatically leads to all the ants following a single path. ACO has a characteristic positive feedback for recovery of good solutions and distributed computation which helps in avoiding premature convergence, but its main drawback is poor computational complexity [73]. ACO has a wide range of learning applications, including: ordering, scheduling, vehicle routing, classification, and assignment problems. The main application of ACO to power system is in the area of determining the shortest route for transmission networks [26, 73].

(b) Particle Swarm Optimisation

Introduced by Kennedy and Eberhart in 1995 [71] [206], Particle Swarm Optimisation (PSO) is a population-based, global optimisation approach, modelled on the social behaviour of organisms, such as flocking of birds and schooling of fishes [35]. Potential solutions (called particles) move around the search (problem) space, following the current particles’ best fitness. Each particle keeps track of its position coordinate in the search space, and moves towards the best known position. Thus, the movement of the particles is determined by both their best locally know positions, *pbest*; as well as overall best positions, *gbest*, which are periodically updated as the particles discover new better positions [73]. Each particle has a memory feature which enables it to remember the best position on the search space it visited. The population of the particles is called a swarm. Since its introduction, there exist several modifications and variants of PSO, including: binary PSO, real-valued PSO, integer PSO, vector PSO, Gaussian PSO, adaptive PSO, discrete PSO [207]. PSO is the most widely used of the three SI approaches in solving ELD problems in recent decades, and although it has the ability of quick convergence through exploration, but it is very slow in exploitation. It encounters a problem while escaping from local optima when stuck. Applications of PSO to power systems are in the areas of unit commitment, ELD, power and voltage control, power system reliability and security, load forecasting and generator scheduling. In [33], an improved PSO was proposed for solving ELD problems taking into consideration the non-linearity features of power generators such as ramp-rates limits, prohibited operating zones as well as non-smooth cost functions with valve-point effects, transmission lines losses, and spinning reserve constraints. The algorithm was validated for two test systems consisting of 15 and 20 thermal generating units.

(c) Bee Colony Optimisation

Proposed by Karaboga in 2005 [72], Bee Colony Optimisation (BCO) is an optimisation algorithm inspired by intelligent food foraging behaviour of swarm of honey bees. Three groups of bees make up a colony: employed bees, onlookers and scouts. For each food source, there is an assumption of only one employed bee; therefore the number of employed bees equals the number of food sources around the hive. Several mechanisms (such as waggle dance) are employed to find optimal location of food sources and search new ones [60]. An onlooker watches the dances of employed bees and chooses food sources depending on dances, while a scout is an employed bee whose food source has been abandoned, and hence starts to find a new food source. The position of a food source represents a candidate solution to the problem; the amount/quality of a food source is the fitness, while the number of the employed bees is equal to the number of solutions in the population. An employed bee also has a memory feature to track the position of new food source. Application of BCO to real-world problems are in the areas of structural optimisation, estimation and classification problems, and power electronic design. In [60], a honey BCO was used to solve the ELD problem with generator constraints such as ramp-rate and prohibited operating zones; and tested with 6 and 15 units systems, where better results (in terms of efficiency and computational time) were obtained, in comparison with other conventional approaches.

(3) Artificial Immune Systems

The Artificial Immune Systems (AIS), inspired by the works of Farmer, Packard and Perelson (1986) [75], are a class of computationally intelligent systems based on the principles and processes of the immune system. An AIS-based algorithm combines the characteristics of learning and memory of the system to solve a problem, which investigates the application of the immune systems to solving computational problems from mathematics, science and engineering. AISs adopt learning, memory and associative retrieval system to solve recognition, classification and optimisation tasks. Four common optimisation techniques and algorithms used in AIS are: clonal selection algorithm, negative selection algorithm, immune network algorithm and dendritic cell algorithm. Common application areas of AIS include: functions approximation, pattern recognition, anomaly detection, network security, noise tolerance [61]. In [62], an adaptive clonal selection algorithm was implemented involving several cloning, mutation and selection approaches for solving power problems, to determine the optimal active power generated in power systems. It was tested with a system of 18 generating units, and shown to be a promising technique for solving complicated optimisation problems in power system operations.

(4) Shuffled Frog Leaping Algorithm

Proposed by Eusuff and Lansey in 2003 [177], the Shuffled Frog Leaping Algorithm (SFLA) is a recent nature-inspired MA, which involves a set of frogs that co-operate with each other to achieve a unified behaviour for the whole system [59]. With its simplicity, robustness and fastness, it proves to be very efficient in calculating the global optima of many problems with large search space, and has been used to find high quality solutions to complex power problems. In [59], a new modified shuffle frog leaping algorithm (MSFLA) was proposed for a practical non-smooth ELD problem in an attempt to reduce processing time and improve quality of solutions. It was tested on two systems consisting of 6 and 40 thermal units to confirm efficiency of approach.

(5) Group Search Optimiser

The Group Search Optimiser (GSO) is a population-based search algorithm which draws inspiration from group-living, typically found in the animal community [174, 208]. It employs the concept of resources searching mechanism (a movement behaviour whereby animals search for food, mate or nesting sites), using producer-scrounger (PS) model to design optimal searching strategies. Although individual animals can try to search sparse resources which are randomly located alone, but group search aids them in information sharing which helps in increasing search rates and reduce success variance. A group consists of producers, scroungers and rangers. The behaviour of the first two is based on the PS model, while the third perform random walk motions. The larger the group, the smaller the proportion of individuals need to guide the group with more accuracy [209]. The producer-scrounger model is simplified for accuracy and ease of computation by assuming only one producer with several scroungers and rangers at each search point [208]. In [174], a new modified group search optimiser (MGSO) is presented for improving the scrounger and ranger operators of GSO for solving the non-convex economic dispatch problem. Performance evaluation was tested with systems of 3, 13 and 40 systems.

(6) Biogeography-Based Optimisation

Developed by Dan Simon in 2008 [210], motivated by natural process and geography, Biogeography-Based Optimisation (BBO) is a global optimisation algorithm that studies the distribution of species of organisms and ecosystems in geographic space and through geological time. Bio-geographical models describe the evolution of species (speciation), migration of species (immigration and emigration) between islands, and the extinction of species. This inspiration allows for information sharing between candidate solutions [200, 210]. A candidate solution is an island, and their habitat characteristic features are called suitability index variables (SIV). The fitness of each solution is called habitat suitability index (HSI). Islands with high HSI are said to be friendly to life, and they tend to share their features with those with low HSI by emigrating solution features to other habitats. Those with low HSI accept a lot of new features from the high HSI solutions by immigration from other habitats. This process improves the solutions and evolves a solution to the optimisation problem. Migration and mutation are the two main operators of BBO. Mutation increases the diversity of the population to evolve better solutions. A remarkable feature of BBO is that the original population is not discarded after each generation, but modified by migration. The immigration and emigration rates are determined by the fitness of each solution. BBO has successfully been applied to solving both SELD and DELD problems in [171, 211, 212], tested with 3, 6, 15 and 40 problem cases.

(7) Simulated Annealing

Simulated Annealing (SA) is based on thermodynamics and metallurgy, inspired by the formation of crystals in solids during cooling, in order to increase the sizes of the crystals and reduce unwanted defects. Developed differently by Kirkpatrick *et al*, in 1983 [213], and V. Cerny in 1985 [214], the method was introduced to solve complex optimisation problems. Optimal solution results from a careful choice of cost function, appropriate mutation mechanism, solution space and cooling process. General/implementation simplicity, ability to deal with arbitrary cost functions and to refine optimal solutions necessitates its applicability. However, its major draw back is repeated annealing [26, 73], leading to high computing time requirement. This can be overcome through parallel processing in a multi-processor system. Applications to power system are in the area of unit commitment, maintenance scheduling and planning of transmission expansion. The potential and effectiveness of SA (over other contemporary approaches in literature) in solving the SELD with non-smooth cost function was investigated in [175], and tested with 3, 13 and 40 generating systems; and adapted to the DELD problems in [64].

(8) Harmony Search

The Harmony Search (HS) algorithm, proposed by Geem *et al*, in 2001 [215], is a new meta-heuristic nature-inspired algorithm which mimics the improvisation process of music players to find the perfect state of harmony. HS has successfully been applied to solving Mechanical and Civil Engineering optimisation problems [216]. Each musician is a decision variable, note is a value, while best harmony is a global optimum. The SELD problem was solved using HS in [101, 216]. Steps to HS algorithm implementation are as follows:

(i) Defining the problem and specifying the parameters. The parameters are: the harmony memory size/number of solution vectors in the memory, harmony memory considering rate, pitch adjusting rate, and the number of improvisations (stopping criterion);

(ii) Initialising the harmony memory, generated uniformly within a given range (lower and upper bounds of each decision variable);

(iii) Generating a new harmony, called ‘improvisation’, using memory consideration, pitch adjustment, and random selection;

(iv) Updating the harmony memory – the generated harmony vector replaces the worst harmony in the harmony memory if its fitness is better than the worst harmony;

(v) If the stopping criterion is met (number of iterations is reached), the computation is terminated, otherwise, steps (iii) and (iv) are repeated.

A modified HS was used to solve the SELD problem in for both smooth and non-smooth cost functions [173], and tested with systems of 6, 15, 18 and 40 generators; whose results compared better than those of GA and PSO.

These computational approaches have the ability of finding global optimal or near-optimal solutions to power systems optimisation problems, though, most of them suffer from long computation time requirements and a large number of problem-specific parameters. A greater percentage of power systems optimisation research is based on EAs. We therefore present a survey and review of EA related work in a subsequent section in this chapter.

2.6.3 Hybrid Methods

Practical modern and real life power systems problems are characterised by non-convex cost functions and non-linearity of generators outputs, such as ramp rate limits, prohibited zones and complicated constraints. These characteristics render the power systems problems very challenging for any optimisation algorithm, as it may not be able to find a sufficiently good solution in good computational time for larger systems [51]. As in many other fields, this challenge has led to the exploration of hybrid methods in this application domain. A hybrid method integrates two or more optimisation techniques in attempt to combine their strengths and overcome any inherent weaknesses in either of them. For example, in [58], a multiple tabu search (MTS) algorithm was proposed for a constrained practical power problem. To improve the performance of conventional TS, the MTS introduced additional mechanisms such as adaptive searches, multiple searches and a restarting process. The algorithm was tested for 6 and 15 generating units, and compared with conventional GA and SA, where it exhibited superior performance. Though TS is good in solving complex power problems, the only problem is that it is a problem-dependent algorithm and it exhibits great difficulty in defining good strategies for memory structure that work across different problems. In [79], particle swarm optimisation (PSO) was combined with simulated annealing (SA) to create an innovative approach capable of generating high quality solutions with greater stability in convergence characteristics and reduced calculation time. This is as a result of PSO's major problem of premature convergence, where the particles readily fly into local optima, and SA has a high probabilistic jumping feature for helping to overcome the PSO's major weakness. The work proposed a new coding scheme, which involved normalising the current generator's output based on the previous output, and then rearranging the ramp boundaries to consider the limits. In [172], an improved/hybrid HS (with PSO) was used to solve the DELD problem, and tested with systems of 5, 10 and 30 generators. The power system problems application areas of the most common hybrid AI-based approaches are [26]:

- GA/FL: Economic load dispatch, power system control, and power stability;
- GA/GP: Power system identification, controller design;
- ANN/FL: Power generation and distribution, load forecasting, fault diagnosis, and scheduling of generator maintenance;
- FL/Expert systems: Power system planning; fault diagnosis;
- GA/PSO: Load flow, power distribution;
- GA/SA: Generator maintenance scheduling;
- GA/ANN/FL/Expert systems: Load forecasting, planning generation expansion, and stabilization of power systems;
- GA/FL/SA/Expert systems: Scheduling of generator maintenance and reactive power planning.

2.7 Evolutionary Algorithm (EA)

In solving optimisation problems, EAs rely on the concepts of selection pressure and diversity, proceeding in broad and simple terms by the application of selective pressure to a diversified population [63]. This leads to the strongest among the members of the population dominating, a phenomenon called "survival of the fittest". The central idea in EAs is the maintenance of a population of individuals called chromosomes, representing possible solutions to a problem. With the passage of time, the members of the population interact with each other and move through subsequent iterations called generations. Fitter solutions (the ones closer to the optimal solutions) have the higher probability of passing through to the next generation. Through the three classic operators of selection, crossover and mutation, EA performs a search mechanism on the population, and directs the search to promising regions efficiently. Thus the problem of getting stuck in local optima is avoided. EA is a rapidly growing research area in Artificial Intelligence, and has a wide range of applications in business, science and engineering, including power systems optimisation [14, 63, 83]. A general workflow of a simple EA is given below:

1. *Encode* the problem to be solved as a string of chromosomes;
2. *Generate* an initial population of chromosomes;
3. *Evaluate* the fitness of each of the chromosomes in the population;
4. *Select* parent chromosomes from the population to form breeding pair;
5. Breed the parent chromosomes:
   (i) Perform *crossover* on the parent to produce children;
   (ii) Perform *mutation* on the children;
6. *Evolve* (replace the parent population with the child population to form new population). If *stopping criterion* is reached, output the best chromosome as the required solution, otherwise go back to step 3 above.

2.7.1 Encoding a Problem

An EA works on a population of chromosomes. A chromosome is a string which represents a solution to a particular problem. It is an abstraction of the DNA chromosome in biology, which is conceived as a string of letters in English language alphabets [68]. The exact position in a chromosome is called a *gene*, while the value which is present at that location is called an *allele*. The ways in which the alleles are represented in a chromosome is referred to as *encoding*. The most popular or classical encoding is the binary encoding (bit-string representation with values 0 or 1) [36]. Others include: permutation encoding, value encoding and tree encoding [63, 83]. Tables 2.3, 2.4 and 2.5 are representations of binary, permutation and value encoding.

Chromosome 1	*1011* *allele 1*	*0100* *allele 2*	*0101* *allele 3*
Chromosome 2	*0110* *allele 1*	*1110* *allele 2*	*1000* *allele 3*

Table 2.3: Binary encoding

Chromosome 1	*4* *allele 1*	*2* *allele 2*	*9* *allele 3*	*5* *allele 4*
Chromosome 2	*1* *allele 1*	*3* *allele 2*	*7* *allele 3*	*2* *allele 4*

Table 2.4: Permutation encoding

Chromosome 1	*52.78*	*101.33*	*97.02*	*77.39*
	allele 1	*allele 2*	*allele 3*	*allele 4*
Chromosome 2	*201.01*	*337.96*	*419.36*	*105.44*
	allele 1	*allele 2*	*allele 3*	*allele 4*

Table 2.5: Value Encoding

Permutation encoding is used in ordering problems such as the travelling salesman problem (TSP), task ordering problem, etc. Here, the chromosome is a string of a sequence of numbers. This encoding is useful when individual fitness depends on positions of genes in chromosome. Each number represents a city to be visited, and in the order of visit. In some problems of high complication (including most power system problems), direct value encoding is used. Here, the chromosome is a string of numbers representing the characteristics of the problem domain. In this thesis we invariably use a value encoding, where allele values represent generator real power outputs.

2.7.2 Generation of the Initial Population

The initial population for an EA is a set of candidate solutions to the problem. There are many methods of generating the initial population of chromosomes. The common method is random generation. Here, the allele values are random numbers based on the defined boundaries, such as the lower and upper limits of each generator. While this approach is efficient and provides a population covering the feasible solution space, the entire initial population may also be infeasible. That is, subsequent generations may not be as the previous ones, resulting in good solutions to evolve slowly. The initial population could also be the output of another search algorithm in a situation where a hybrid approach is used.

2.7.3 Fitness Evaluation

To use an EA to solve a specific problem, one of the first steps is to specify a fitness function which calculates the quality of any candidate solution to that specific problem. The fitness evaluation process simply uses this fitness function to compute the quality of each chromosome. In analogy to biology, the chromosome is the genotype, while the solution it represents is the phenotype [68]. The computation takes several factors and objectives into account, including the cost minimisation/maximisation, penalty handling, resources utilisation, run time, etc.

2.7.4 Selection Methods

This is the main driving force of an EA. Selection is the process of choosing individuals (parents) from the initial population to go into the mating pool, in order to generate offspring. This is the basis of new population. An ideal selection method should be simple, computational efficient and suited for parallel implementation [83][184]. An EA uses fitness to discriminate the qualities of solutions represented by chromosomes in any population [68]. The general motive behind all selection methods is to provide a selective pressure (guided by fitness values) in favour of better solutions. Therefore, selection process models the idea of survival of the fittest, and helps to narrow search space to optimal bands. Selection balances exploration against exploitation [16]. Exploration explores new parts of the search space to get into a region of high fitness, while exploitation focuses on finding the best solution in the highly fit region of the search space. These concepts need to be carefully balanced in other to prevent highly-fit individuals from taking over the entire population, thereby wiping out all useful information which may be present in the less-fit individuals. On the other hand, too weak solutions will result in too slow evolution. The balance drawn between these two goals is primarily controlled by selection pressure. Exploration is paramount in the early stage of the evolutionary process, whereas exploitation becomes more important as regions with good solutions are found.

Different selection methods can be used depending on the design of the algorithm. These methods can be categorised into two groups [34]:

1) Ordinal Selection;
2) Proportional Selection.

In ordinal selection, the decision is based on the ranked order of the solutions' fitness values. *Truncation selection, tournament selection* and *rank selection* fall into this category. In truncation selection, *N* fit solutions from the parent population are selected at once. Tournament selection works by selecting the fittest out of two (or more) randomly chosen chromosome from the parent population. The selective pressure of tournament selection can be adjusted by means of the tournament size parameter, which is small relative to the population size [63]. The ratio of tournament size to population size can be used as a measure of selective pressure. A tournament size of 1 is equivalent to selecting an individual at random, while a size equal to the population size is equivalent to selecting the best individual at any given point.

Rank selection aims at preventing an early convergence [84]. The individuals in a population are assigned numerical ranks according to their fitness values, and the expected value depends on its rank, rather than its absolute fitness [85]. This approach prevents the highly-fit individuals from dominating the population at the early stage of the evolution at the expense of less-fit ones. The problem of reducing the diversity of the population is also eliminated. The algorithm for the rank selection method is as follows:

1. *Rank each individual in the population in increasing order of fitness from 1 to* N *;*
2. *Choose the expected value* Max*, of each individual with rank* N*, where* $Max \geq 0$*;*
3. *The expected value of each individual in the population at time is given by:*

$$E(i,t) = Min + (Max - Min)\frac{rank(i,t) - 1}{N - 1} \tag{2.23}$$

Where:

N *= Highest rank (population size);*

Max *= Value of the highest rank;*

Min *= Value of the lowest rank;*

i *= Individual in the population;*

t *= time.*

The highest rank is N, with expected value Max, while the lowest rank is 1, with expected value Min. All other individuals get expected values equally spaced between these two [83]. Solutions always allocate on the basis of rank, and so neither over explore/exploit. Evolution is also more controlled. However, in most cases, the algorithm will be slower in finding highly-fit individuals.

In proportional selection, the selection probability for each individual in the current population is determined, and then sampled to make the breeding pool. *Roulette-wheel selection* (also known as fitness proportionate selection), *Boltzmann selection* and *Sigma scaling selection* fall into this category [63, 83]. In roulette-wheel selection, all individuals have a chance of being selected at any given point. Furthermore, the probability that a given individual will be selected is proportional to its fitness, and equal to its normalised fitness. The sum of all the normalised fitnesses in the population is 1. Because highly-fit individuals have higher normalised fitnesses, they have a higher chance of being selected. Also, because all individuals have some chances of being selected, there is less loss of genetic diversity than if the single best individual of every generation is selected. The problems of the roulette-wheel selection approach are: (1) It handles only maximization problems; (2) Scaling of the fitness becomes very important near the point of convergence; (3) Individuals with very poor fitness values die out quickly, resulting in premature convergence. The "expected value" of an individual, that is, the number of times that individual will be selected to reproduce, is computed as the fitness of the individual divided by the average fitness of the total population:

$$p\,(x\ is\ selected) = \frac{f\,(x)}{\sum_{y \in P} f\,(y)} \tag{2.24}$$

Where:

$f\,(x)$ = Fitness of solution, x

$f\,(y)$ = Average fitness of the total population

P = Population size

The algorithm for the roulette-wheel selection is as follows [83]:

1. *Sum the total expected value of individuals in the population,* T;
2. *Repeat* N *times, where* N *is the number of individuals in the population;*
3. *Choose a random integer,* r *, between* 0 *and* T *;*
4. *Loop through the individuals in the population, summing the expected values, until the sum is greater or equal to* r *.*
5. *The individual whose expected value puts the sum over this limit is the one selected.*

In Boltzmann Selection, temperature is used to control the rate of selection of the individuals. All individuals are initially given a good chance to contribute to the final solution, depending on the temperature chosen. The expected value of an individual is computed as:

$$E(i) = \frac{e^{f(i)/T}}{\left\langle e^{f(j)/T} \right\rangle} \tag{2.25}$$

Where:

$f(i)$ = *Fitness of individual;*

T = *Temperature used to set selection pressure;*

$e^{f(j)/T}$ = *Sum over all solutions in the entire population.*

The temperature starts out high, lowering the selective pressure as the temperature increases. Thus, every individual have some good chance of being selected. The temperature is gradually decreased, leading to an increased selection pressure. As evolution progresses, the value of T decreases, until either the maximum number of generations is reached or the optimal solution is found.

Sigma scaling (or sigma truncation) selection aims to maintain constant selection pressure over the course of the evolution. Here, an individual's expected value depends on its fitness, mean as well as standard deviation of the population [63]. The expected value $E(i,t)$ of and individual is given as:

$$E(i,t) = \begin{cases} 1 + \dfrac{f(i) - \bar{f}(t)}{2\sigma(t)} & if\ \sigma(t) \neq 0 \\ 1 & if\ \sigma(t) = 0 \end{cases} \tag{2.26}$$

Where:

$f(i)$ = *Fitness of individual* i *;*

$\bar{f}(t)$ = *Mean fitness of the population at time* t *;*

σ = *Standard deviation.*

At the earlier stage of the evolution, when the standard deviation of the fitness is high, highly-fit individuals will not be much. But as the algorithm converges and standard deviation becomes lower, the highly-fit individuals increase in number. Sigma scaling normalises the spread of fitness using standard deviation, thereby reducing the risk of premature convergence.

2.7.5 Tournament Selection

Tournament selection is the selection method used in this work. It simply chooses *t* individuals from the population (uniformly at random, with replacement), and the fittest of those individuals (breaking ties randomly) is the one returned as 'selected'. In the context of selecting a set of N parents, this process is simply repeated N times. The parameter *t* is called the tournament size. A standard and commonly used tournament size is two (and this is called binary tournament selection). It has the following work flow, and diagrammatically illustrated in Figure 2.14:

(i) Choose two individuals at random from initial population;

(ii) Pick a random number, r (between 0 and 1);

(iii) If r < k (where k is a user-defined parameter, over 0.5 but less than 1), the fitter of the two individuals is selected as parent to go into the mating pool;

(iv) Else, the less-fit individual is selected;

(v) Return the two individuals to the original population.

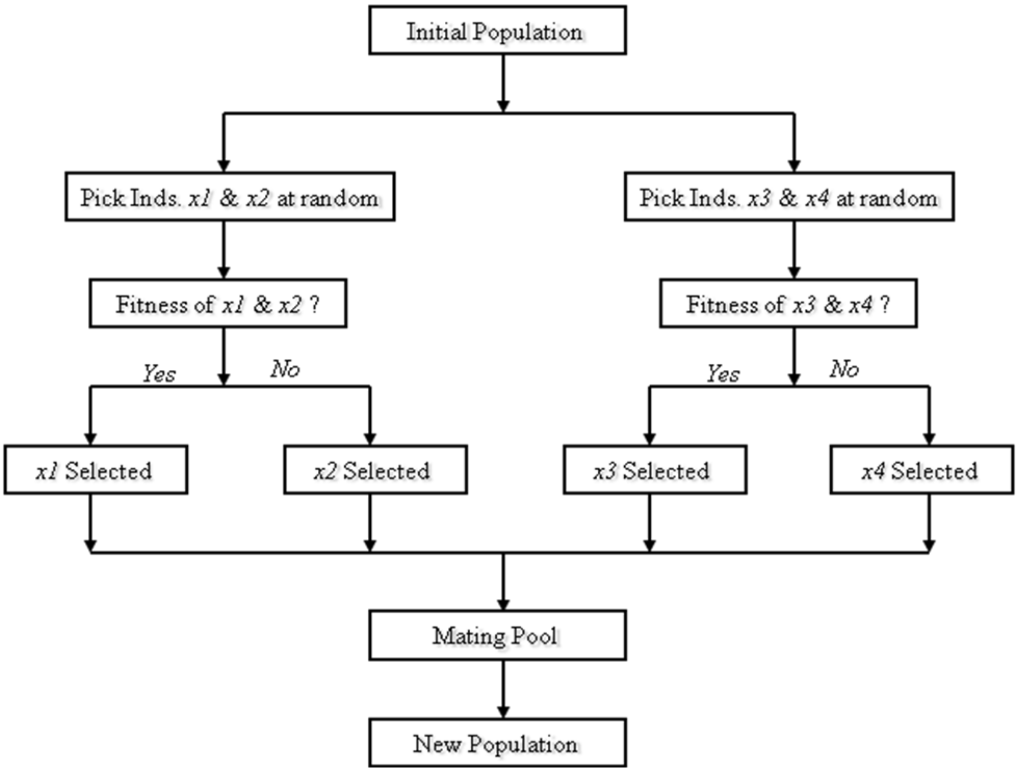

Figure 2.14: Diagrammatical illustration of tournament selection

Tournament selection is known to overcome the weaknesses of other selection methods, including the roulette wheel selection, by preventing the domination of highly-fit solutions, thereby wiping out all useful information which may be present in the less-fit ones. It also eliminates too weak solutions which will result in too slow convergence.

2.7.6 Breeding

This is also known as recombination [68], a process which follows selection where the selected chromosomes (parents) from the current population are recombined to form a successor population (children). This is to stimulate the mixing of genetic composition of the parents when they reproduce. It is expected that more highly-fit chromosomes will result from this process since selection is biased to favour chromosomes with higher fitness. Breeding is achieved by applying *genetic operators*, which generate new candidate solutions by using parts of existing candidate solutions (the selected parents). The canonical operators are *crossover* and *mutation*. Crossover typically combines parts of two parent solutions to form two child solutions; however in general, such a *recombination* operator can generate a child from more than two parents, and generate one or more children. In contrast, a mutation operator always involves a single parent, and the child is a 'mutation' of that parent – that is, it will typically be the same as the parent except for changes in a small number of its alleles. Genetic operators are invariably non-deterministic.

2.7.6.1 Standard Crossover Methods

Standard crossover exchanges the genes (genetic composition) between two parents. Crossover typically takes two parents and produces two children, with the children inheriting a mix of genes from the parents. In most EAs it occurs with a high probability, called the crossover rate. Following selection of parents, in a standard implementation, a random number between 0 and 1 is generated which is compared to the crossover rate. If the crossover rate is lower than the number, no crossover occurs and the parents progress to the next phase unchanged. But if the crossover rate is greater or equal to the number, crossover is done. One-point crossover is the simplest crossover operator. Other alternatives which are generalisation of the one-point crossover are: two-point and multi-point crossover operations. Another form, called uniform crossover [68], selects uniformly between the allele values of parents at each point to form children. Figure 2.15 shows a one-point crossover acting on a binary encoding.

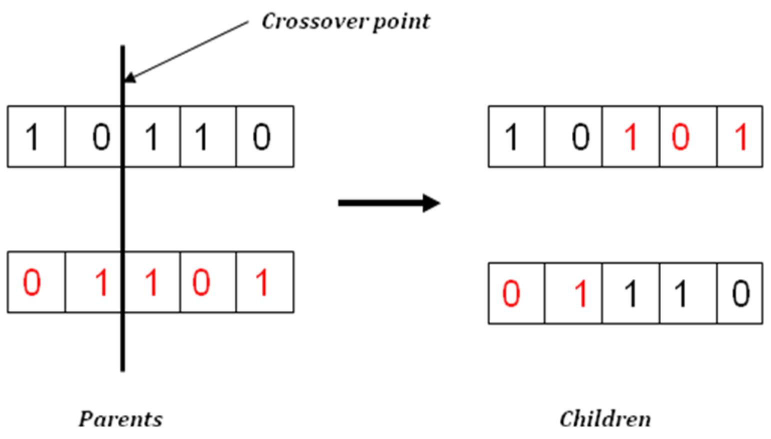

Figure 2.15: One-point crossover for binary encoding

2.7.6.2 Mutation Operators

This is a major source of genetic variation. In a typical EA design, the mutation operator is applied to the resulting children solutions after the crossover, and allows new genetic patterns to be introduced, whether desirable or undesirable. Mutation could be viewed as a transition from a current solution to its neighbourhood solution in local search algorithms, as it randomly changes the value of a part of the solution to another. In other EA designs, mutation is a standalone operator that is applied with its own 'mutation rate'. Either way, EAs needs mutation to avoid genetic stagnation, because standard crossover methods cannot introduce new alleles to the population. Mutation is a background operator which ensures that alleles have the possibility of entering and providing genetic variation in a given population. It usually occurs with a low probability, typically between 0 and 1. The GA process could be very detrimental if the mutation rate is set too high, as it will force the population to adapt to a new environment, and will not necessarily produce optimal individuals due to the instability of the population [63].

In its simplest form of operation, the genes undergoing mutation are 'randomly' selected, but in other cases, they could be 'targeted'. Different types/forms of mutation operators exist for the different encoding methods: Single-gene mutation, *N*-random gene mutation, uniform mutation, boundary mutation and Gaussian mutation, bit-flip mutation, swap mutation, insertion mutation, inversion mutation, scramble mutation, displacement mutation, real-valued mutation, etc [180][185]. Single gene mutation chooses an allele at random, and changes it to a random new value, with a probability, called the "mutation rate". *N*-random gene mutation repeats the single-gene mutation *N* times. Uniform mutation replaces the value of the selected gene with a uniform random value chosen between user-defined upper and lower limits. Non-uniform mutation increases the probability that the amount of mutation will be close to zero with increased number of generation. Boundary mutation randomly selects an allele, and replaces it with either the lower or upper limits. Gaussian mutation adds a unit Gaussian distributed random value to the selected gene.

For binary encoding, a bit-flip is implemented by randomly switching of bits (0 to 1 and 1 to 0). Figure 2.16 shows a one-point bit flipping mutation for a binary encoding.

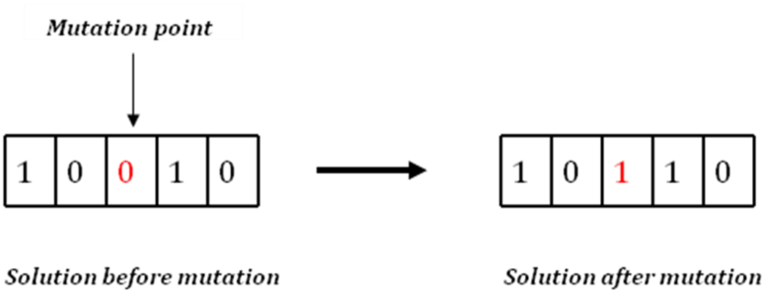

Figure 2.16: One-point, bit flipping mutation for binary encoding

For permutation encoding, swap, insertion, inversion, scramble and displacement mutations are commonly used. Swap mutation generates a new chromosome by swapping two alleles randomly from the current chromosome, as shown in Figure 2.17, where the alleles in the third and fifth gene positions are swapped with each other:

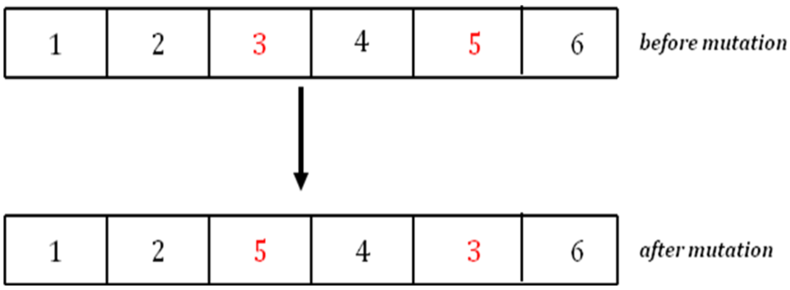

Figure 2.17: Swap mutation for permutation encoding

Insertion mutation selects an allele at a random gene position and inserts, also randomly, into another gene position within the chromosome. Figure 2.18 shows an insertion mutation for permutation encoding.

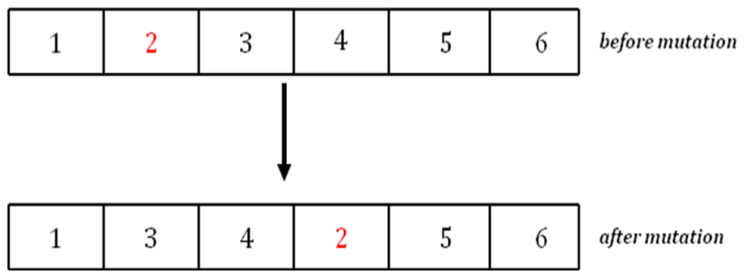

Figure 2.18: Insertion mutation for permutation encoding

Inversion mutation randomly selects two gene positions, and inverts (reverses) the alleles in those positions, as shown in Figure 2.19, where the alleles between the second and fifth positions (inclusive) are inverted.

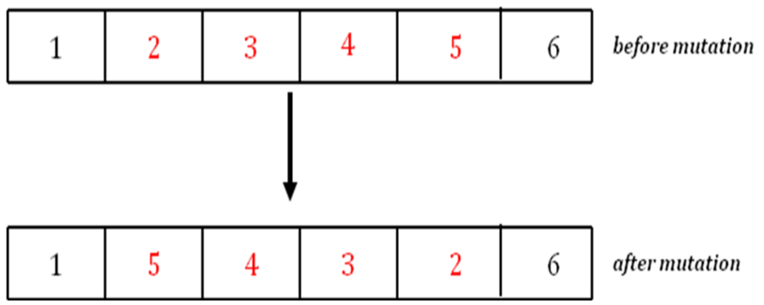

Figure 2.19: Inversion mutation for permutation encoding

Scramble mutation randomly selects two gene positions, and scrambles (randomly re-arranges) the alleles in those positions. This is illustrated in Figure 2.20.

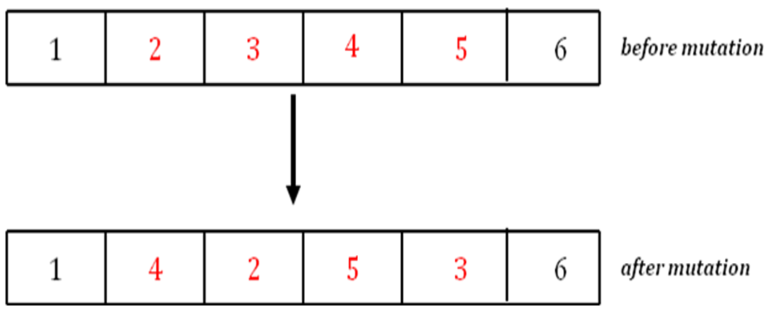

Figure 2.20: Scramble mutation for permutation encoding

Displacement mutation randomly selects two gene positions, and moves the alleles between them as a block to another random position. Figure 2.21 illustrates a displacement mutation for permutation encoding.

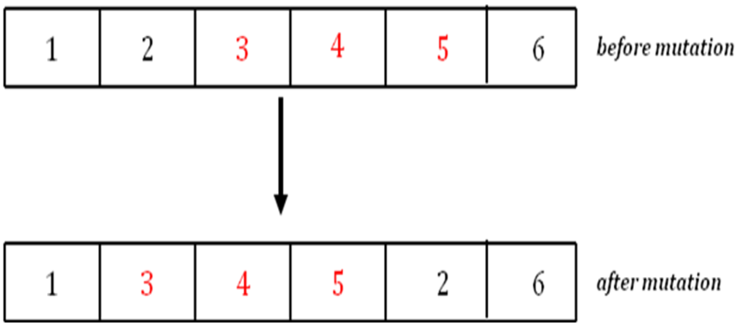

Figure 2.21: Displacement mutation for permutation encoding

For value encoding, a real-valued mutation operator randomly selects an allele (a number) from the parent solution, changes its magnitude defined by the programmer, and returns the result in the child solution as shown in Figure 2.22 below, where the magnitude of the allele in the third gene position is changed from **12.99** to **13.57.**

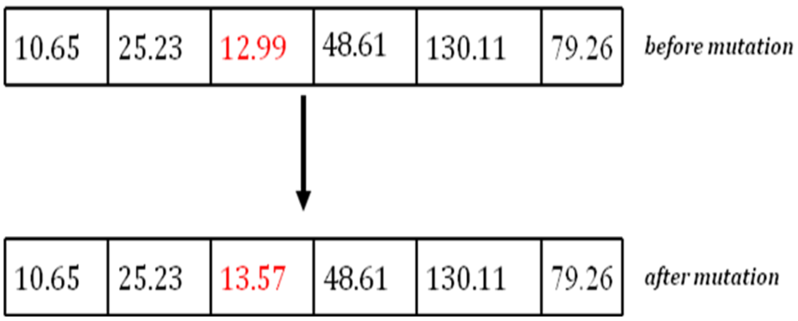

Fig 2.22: Single gene real-valued mutation for value encoding

2.7.7 Evolution

The resulting child population after mutation replaces the old population, forming the successor population. The selection and breeding processes are iterated, leading to a succession of 'generations' of solutions. At each next generation, the successor population becomes the source (parent) population. The EA goes through a number of generations until one or more of the stopping criteria (reaching a fixed number of iterations or finding a sufficiently good solution) are met, where the best chromosome in the resulting population is returned as the solution to the problem. To prevent losing the best individuals of the population from one generation to the next, elitism is used. This technique copies the top *N%* of the present population on to the next generation, where *N* = elitism rate.

2.7.8 Tuning of Evolutionary Parameters

When building an EA, there are various design decisions that need to be made. The encoding type, population size, crossover and mutation rates, etc need to be carefully chosen as their values have a combined effect on the performance of the EA. This is called "tuning" of genetic parameters [86]. There is a "parameter space" that consists of all the possible sets of EA parameters (for standard EAs, these are: population size, crossover rate and mutation rate). An element of this space is a triple defined as *(pop_size, cross_rate, mut_rate)*; e.g. (100, 0.8, 0.5) [63]. This varies widely between different problems and encoding. It is very difficult to arrive at specific values that work for all EAs. De Jong in 1975, determined systematically the effects of the different parameters on the performance of most EAs [63], and concluded that the optimal parameters were: population size of 50 to 150 individuals, one-point crossover of rate 0.6 (and above) and a mutation rate in the region of 0.001 [186]. Grefenstette in 1986, evolved an optimal set of parameters for De Jong's EA, and recommended: (30, 0.95, 0.01) [87, 186]. Based on the work of Grefenstette, it was therefore agreed to begin an EA search by considering a population size of 30 individuals, crossover rate of 0.6 and a mutation rate of 0.01 [86]. The tuning of evolutionary parameters for the current work is based on this recommendation.

For static mutation probability (or rates), and in addition to the above, recommendations were also made through experience as well as trial and error. Schaffer (1989) [186], recommended a region of [0.001, 0.005], Bäck (1992) [186], derived the following expression: $1.75/N * L^{½}$, where N is the population size and L is the chromosome length; while Muhlenbein (1992) [186], suggested that the optimal value of the mutation rate should be: $1 / L$. While these values were widely used and provided a good basis for much earlier work, however, as the range of problems expanded, it became clear that no single set of parameters is optimal over a wide range of problems. Recent researches are progressing towards the area of adaptive mutation techniques [186], tailored to the problem requirements and domain. There are two levels of adaptations in mutation – top level and bottom level [187]. The top-level adapts the mutation and crossover ratio in a given EA run [188, 189]. In the COBRA (COst Based operator Rate Adaptation), developed in [189], the EA swaps a given k fixed probabilities periodically between k operators, giving highest probability to the operator that has high fitness value. The bottom-level performs a deterministic self-adaptive mutation probability uniformly or non-uniformly over each gene position [187, 190]. The self-adaptive mutation, developed in [190], was achieved by adding a probability vector for each individual. The operator first mutates the mutation probability with itself, and the resulting probability was used to mutate a targeted gene.

2.7.9 EA Applications

Though EAs have been used in solving a variety of optimisation problems, it cannot be recommended in solving every problem. For some scientific/analytical functions involving few variables, traditional calculus-based methods are much better and faster. EAs are however very effective at solving problems that are believed to be NP-hard – i.e. problems where the only known algorithms that can guarantee a solution have exponential time complexity. A good example is the travelling salesman problem, where the salesman is required to visit all locations exactly once and minimising the total distance travelled, that is, determining the shortest route for the tour. For n locations there are in the region of $n!$ possible routes for the salesman. As the number of locations grows, the number of possible routes grows exponentially, and so does the time required to find the guaranteed best route. Using an algorithm that guarantees to find the best route, for example, it could take about 160 years to find the solution to a 20-location travelling salesman problem. The economic load dispatch problem is very similar to this, in the sense that the guaranteed optimal solution can only be found from evaluating all possible combinations of the generator outputs, and the complexity increases exponentially with the number of generating units. As an 'approximate' algorithm, EAs, like other approximate algorithms, aim to find a good solution in reasonable time; as such, EAs are arguably the most successful class of optimisation algorithms, with an ever-growing base of empirical evidence showing the success in finding sufficiently good solutions in reasonable time to an extremely wide range of different problems. In particular, EAs have the following advantages that aid its use in solving optimisation problems [14]:

1. Ability to encode variables so that the optimisation is done with the encoded variables (in our case, the real power output of generators);
2. Ability to deal with huge numbers of variables simultaneously, searching the entire parameter space efficiently;
3. Ability to optimise variables with arbitrarily complex fitness functions;
4. Suitability for parallel implementations ;

EAs have proved successful in a very wide range of learning and optimisation areas, including [93]:

1. Path Finding – Ordering, routing, trend spotting, clustering, etc.
2. Engineering – Mechanical design (weight, cost), structural design (beam sizes), electrical design (robotics, electronics, power systems), process control, computer network design, parameter modelling, helicopter pilot modelling, etc.
3. Research and Development – Function optimisation, pharmaceutical drugs design, chemotherapy treatment, curve and surface fitting, etc.
4. Management – Timetabling, job-shop scheduling/distribution, project management, task assignment, courier routing, etc.
5. Financial Markets – Portfolio balancing, budget forecasting, investment/stock market analysis, payment scheduling, etc.
6. Game playing – Board games, prisoner's dilemma, etc.
7. Evolution of natural processes – Fuzzy logic systems, pattern and speech recognition using artificial neural networks, etc.

2.7.10 Multi-Objective Evolutionary Algorithm (MOEA)

The first attempt to solve a MOOP was made by Rosenberg in [19], but due to unsuccessful implementation, the formulation was later restated as a single-objective problem and solved with a GA. David Schaffer proposed the first MOEA in 1984 [88], an approach called the vector evaluated genetic algorithm (VEGA). It consists of a genetic algorithm with modified selection mechanism. A number of sub-populations were generated at each generation by means of proportional selection depending on the fitness functions. The sub-populations were later shuffled to generate new populations, on which GA would be applied. Following the success of VEGA, several MOEAs were developed, as listed in [88, 89, 90]. The algorithms differ in their fitness assignment procedure and diversification approaches.

2.7.11 Approaches to MOEA Design

There are four main approaches involved in MOEAs. They include: weighted formula, lexicographic ordering, goal programming, and Pareto approaches [18, 63, 91].

The weighted formula approach involves the aggregation of all the objectives using different weighting coefficients, transforming the multi-objective optimisation problem into a scalar optimisation problem [18]. It is represented in the form:

$$Min \sum_{i=1}^{k} w_i f_i(\bar{x}) \tag{2.27}$$

Where: w_i is a non-negative weighting coefficient for each objective. The sum of all the weighting coefficients is usually equal to "1". Conceptually, the approach is very simple in usage and implementation, which probably explains its popularity. It has a high computational efficiency and can be applied in generating good non-dominated solutions. The major set-back is that the weight coefficients are arbitrary numbers which most often are not justified.

In the lexicographic ordering approach, the objectives are ranked in order of their importance. Solutions are obtained by minimising the objective functions, starting with the most important and proceeding according to the assigned priority [18]. In a mathematical formulation with k objectives, let $f_1(x)$ and $f_k(x)$ represent the most and least important objective functions, respectively. The first problem is formulated as:

$$Minimise \qquad f_1(x) \tag{2.28}$$

Its solution is x^*, and $f_1^* = f_1(x^*)$ is obtained. The i^{th} problem is given by:

$$Minimise\; f_i(x),\; i = 1, 2, ..., i-1 \tag{2.29}$$

Similarly, $f_i^* = f_i(x^*)$ is obtained. The solution produced at the end, x_k^* is the desired solution x^* of the problem. This approach avoids the problem of mixing criteria that are non-commensurable in the same formula, as each criterion is handled by the approach separately. However, because of the randomness of the process, the approach tends to favour certain objectives over others. This causes the population to converge to some parts of the Pareto front [19].

In the goal programming approach, targets or goals are assigned for each objective to be achieved. The values of these goals are incorporated into the problems in the forms constraints or penalties. The aim of the objective functions is to minimise the deviations from the targets. The simplest form of this approach is formulated in [18] as:

$$Minimise \sum_{i=1}^{k} | f_i(\underline{x}) - T_i |, \ \underline{x} \in F \qquad (2.30)$$

Where T_i denotes the target or goal set by the decision maker for the *ith* objective function $f_i(\underline{x})$, and F represents the feasible region.

In the Pareto approach, instead of transforming the MOOP into a single-objective optimisation problem and then solving it using a single-objective search method, a multi-objective search algorithm is used. The aim is to find sets of solutions in the populations that are non-dominated with respect to the entire population. In typical versions of a Pareto approach, the non-dominated solutions are assigned the highest ranking numbers, and the ranking of any dominated solution depends on how many other solutions in the population dominate it . By definition, a solution x_1 is said to dominate a solution x_2 if x_1 is better than x_2 with respect to at least one of the objectives being optimised, and x_1 is not worse than x_2 with respect to all the objectives.

The concept of Pareto optimality in a two objectives problem is shown in Figure 2.23, where *A* and *B* represent non-dominated solutions on the Pareto front [92]. None of them is preferred to the other. Solution *A* has a worse value of f_2 than solution *B*, but a better value of f_1, while solution *B* has a worse value of f_1 and a better value of f_2. Therefore, neither solution *A* nor solution *B* is dominated by each other.

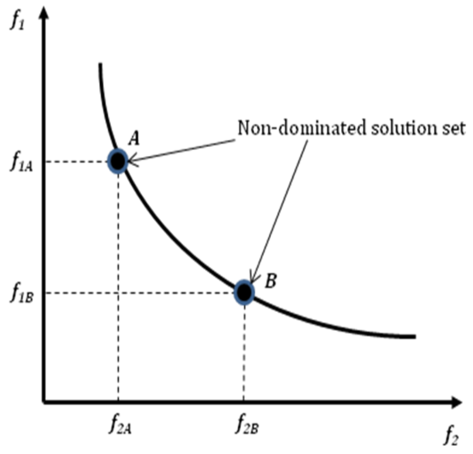

Figure 2.23: Pareto-optimal surface for a two objective problem

2.7.12 Simple Local Search Algorithms

Although EAs are arguably the most widely used algorithms to solve complex optimisation problems (including the ELD), several simpler alternatives are available which tend to be in the class 'local search', and which may be suited to a range of problems. In particular, local search algorithms are often hybridized with EAs, in attempt to better balance the exploratory powers of an EA with the exploitatory skill of local search. Such algorithms include: random search (rarely useful as a standalone algorithm, but can be useful in combination with an EA), greedy search, and hill-climbing [93].

2.7.12.1 Random Search

In a random search algorithm, a fixed number of chromosomes are randomly generated and evaluated and a track of the best solutions kept. The exploration of solutions within the search space is done in an unconventional manner. The following pseudo-code describes a random search algorithm.

*Generate an initial chromosome, **best**, at random.*
While stopping criteria are not met
  *Generate a new chromosome, **next**, at random*
  *If **next** is fitter than **best***
    best = next
  End if
End while
*Output **best** as the solution*

2.7.12.2 Greedy Search

A greedy search algorithm solves a problem by constructing a complete solution to the problem iteratively, using a series of steps. The most favourable option available to the designer is used at each step to extend the current solution to the next phase. The driving force for the popularity of this algorithm is simplicity. Although a best possible move is made at each step/iteration, but however, it does not guarantee that the complete solution is optimal.

2.7.12.3 Hill Climber

Like all local search methods, the hill climber applies an iterative improvement technique to a chromosome (the current) in the search space. Depending on implementation, there are two forms of hill climber: *random mutation hill climber* and *steepest ascent hill climber*.

In random mutation hill climber (RMHC), new chromosome (next) is generated by mutating *current* at a randomly selected gene within the solution string. If *next* is better in fitness than *current*, it becomes the *current*, and the algorithm "climbs the hill towards higher fitness" by a step; otherwise, the current solution is retained. The iteration terminates if there is no further improvement in fitness or time/patient has expired [16]. The steepest ascent hill climber (SAHC) is a modification of RMHC. The algorithm initially considers all possible mutations within the solution string as *next* solution (all possible neighbours to the current solution). The *next* solution that returns the highest fitness value is selected for comparison with *current* solution. If the returned *next* is better than *current*, it becomes the *current*; otherwise no local improvement is possible. That is, the algorithm has reached an optimum [16]. The pseudo-code of RMHC is described as follows:

> *Initialise* ***best*** *chromosome*
> *Select a* ***current*** *chromosome at random*
> *While stopping criteria are not met*
>   *Generate* ***next****, by mutating current at a random gene*
>   *If* ***next*** *is fitter than* ***current***
>   *current = next*

```
            if current is fitter than best
                best = current
            End if
        End if
    End while
    Output best as the solution
```

For a better understanding of a hill-climbing algorithm, consider that the space of all possible solutions to a given problem is represented as a three-dimensional contour landscape; a given set of coordinates represents one particular solution. The better solutions are higher in altitude, forming *hills* and *peaks*, the worse solutions are lower, forming *valleys*. The hill climber then starts from a given point on the landscape and *climbs* the hill. Hill-climbing is a greedy algorithm, as it always makes the best choice available at each step hoping that the overall best result can be achieved this way [93].

Hill-climbing algorithm is very simple and easy implement, once problem representation and the evaluation functions are known. However, it has a number of weaknesses [16]:

1. It terminates at a locally optimum solution;
2. There is no information regarding the deviation of the local optimum solution from the global optimum;
3. The obtained optimum solution depends on the configuration of the initial solution.

2.8 Review of Related Work on EA and Power System Optimisation

EAs have been an actively used optimisation tool in solving real-world engineering problems including power generation, and several variants of it have been developed to suit the particular problem at hand. In this section, we review the extent of work that has been done in using EAs to solve ELD problems. Starting with 'single approach' applications, the review progresses to hybrid methods involving EAs.

The authors in [36] presented a GA with simple binary encoding, single-point crossover, bit-flip mutation and roulette-wheel selection with fitness ranking. The work describes the use of a generic/off-the-shelf EA in solving an optimisation problem, with the algorithm extracting the generator outputs from randomly generated bit-strings and fed into a cost function to compute the fuel cost of three generators. However, the published results were far from acceptable values, the technique provided a good platform.

In [54], DE was used to solve the ELD problem, tested with IEEE 30-bus 6 generators system, neglecting and considering losses. The approach introduced the concept of penalty handling to improve computational resources. The algorithm realised good results in comparison with other reported techniques such as gradient projection, successive linear programming, and GA, but not as comparable as the Newton's method.

In [53], a comparative investigation was made of conventional GA (CGA) and micro-GA (μGA), and tested with the IEEE 6-bus and 31-bus Nigerian grid system. While satisfactory results were realised in both approaches over reported classical methods and a Hopefield neural network approach, CGA showed he major drawback of long computation time, while the μGA (characterized by a very small population size) was found to converge quickly.

In [44], an improved GA (IGA) was integrated with multiplier updating (MU) and was applied to an ELD with valve-point effects and multiple fuels. Performance efficiency of the IGA_MU was investigated by separate applications involving a 10 units test system considering multiple fuels only, a 13 units system considering valve-point effects only, and a 10 units system considering both multiple fuels and valve-point effects. Essential features of the algorithm are: ease of implementation, effectiveness and robustness, automatic adjustment of randomly assigned penalty factors to their real values and application to large-scale system. Results showed IGA_MU's applicability to real-world power problems.

In [46], application of GA to solve both smooth and non-smooth ELD problems were extended to larger test system, considering valve-point effects and transmission losses, but relaxing other equality and inequality constraints. Simulation results were presented for 6 and 40 generating units, which indicate improvements in total fuel cost savings in comparison with other artificial intelligence methods.

In [30], an approach was proposed for solving the combined economic and emission dispatch (CEED) problem using a binary coded GA and PSO. The CEED was converted into a single objective function using modified price penalty factor approach, and tested with 3 generating units.

In [31], a GA was hybridised with fuzzy cardinal priority ranking, where a Pareto optimal solution for the multi-objective generation and emission dispatch problem involving different combinations of fuel cost, oxides of nitrogen, sulphur and carbon, was solved using a non-dominated sorting genetic algorithm (NSGA-II). The approach uses a crowding distance technique to add diversity to the converging solutions, elitist strategy to preserve the best solution in a current population, and NSGA II to provide solutions very close to Pareto-optimal. The result is a single best compromise solution of all the required objectives (reduction of fuel cost and gaseous emissions). It was tested with a system of 6 generators.

In [47], a hybrid method of GA and Lambda iteration was used to economically determine the output of power generators with prohibited operating zones. The approach involved two steps. The first step uses lambda iteration method without considering prohibited operating zones, while the second step uses GA process any units that are caught within prohibited operating zones, by setting their limits to either the lower or upper limits of the prohibited zones.

In [78], a GA was combined with DE and SQP local search to overcome premature convergence and speed up the search process for ELD with valve-point loading effects. The algorithm consisted of two parts: the first part uses a GA to find a near-optimal global search region, while the second part uses DE to explore the search region and SQP to exploit (find tune) the solutions in order to locate local optimum solutions. This was tested with systems of 13 and 40 units.

In [80], a GA was hybridised with Pattern Search and SQP to overcome the drawback of Pattern Search and SQP methods that need to be supplied with initial/suitable starting point by the user. Relying on these good guesses at the initial points makes the methods more likely to get trapped in local optima. Here, the GA generates the initial good solution automatically. Pattern search was used with SQP to refine or fine tune the search. The algorithm reduced total execution time, and was tested with 3, 13 and 40 units.

The EA-based (and by extension, the AI-based) techniques for solving power system optimisation as reviewed so far have covered problems involving smooth and non-smooth functions, encoding type, genetic operators, line losses consideration, constraints (including generators limits, power balance, ramp-rates and prohibited operating zones), computational resources, valve-point loading effects, multiple fuels, number of generating units, environmental consideration, etc. We described and evaluated a new EA approach for solving the optimal flow problem for ELD in the electricity generation industry in [34]. The method combined a standard EA with smart mutation and hill-climbing techniques, and considered benchmark instances of the ELD problem involving minimum/maximum generation limits, power balance, ramp-rates and prohibited operating zones. Violation of either of these constraints introduces the concept of penalties, and these in turn provide the basis for the smart mutation operator. Our smart EA was compared with a basic EA and reported results for other recent algorithms, on three benchmark cases involving 6, 15 and 20 generating units. On the larger two of these problems we find better solutions than have so far been reported in the literature. In later chapters, the review will extend to dynamic problems and bidding context in a deregulated power market. However in the next section we focus on literature in the EA area that pertains to the 'smart mutation' approach explored in this thesis.

2.9 Review of Smart Mutation

There are two broad approaches to realising solutions of optimisation problems using EAs. The first option is 'off-the-shelf' methods (e.g. use standard approaches, such as a binary GA, with standard settings) [36]. Here, methods are designed to encode the problem into binary strings, or to whichever default encoding is used in the off-the-shelf algorithm, which are later decoded into final solutions. A variety of implementations could then be made depending on the nature of the problem being solved, but the emphasis here is on fitting the application specifics into the framework provided by the off-the-shelf optimisation algorithm. The second option involves adapting a standard EA by developing appropriate operators (crossover and mutation) that are designed and adapted with the specific application in mind, which may work in different ways to standard approaches. For example, crossover may make use of application-specific knowledge concerning aspects of solutions that are particularly good or bad in combination. This option also includes the possibility of changing various aspects of the standard algorithm, driven by what seems sensible or insightful in the context of the application domain. Relatively limited work has been done in this context which can be seen as generally applicable across a class of problem domains, however it is of interest in this light to consider a number of works below [178, 179, 180, 181, 182, 183], in which the common theme is the exploitation of problem-specific information generated when calculating a chromosome’s fitness..

The focus of [178] is on SAT (satisfiability problems), which are essentially constraint satisfaction problems. The fitness function calculates a weighted sum of penalties associated with the violation of constraints. The idea of the SAW (stepwise adaptation of weights) approach was to continually change the weight values for the constraints, giving higher values to constraints that were difficult to satisfy. This is related to the idea of smart mutation, in the sense that it is an alternative way to use 'violation' information to guide the search. In [179], timetabling problems were solved using a chromosome structure in which each gene position was a specific event (e.g. an examination), and the value of the gene indicated the timeslot for that event. In this scenario, given the information that can be accumulated when calculating fitness, it was possible for mutations to be targeted towards genes (events) that were involved in penalty violations (clashes with other events); it was also possible to gather information relating to good possible new timeslots for those events. Various smart mutation operators were tested to exploit these ideas. A swap mutation operator for permutation-based representation by generating a new chromosome was developed in [180]. This was achieved by randomly swapping two genes from a current one (aiming at reducing overall fitness), applied to a job-shop scheduling problem. In [181], a position mutation was used to change the amplitude of a parametric movement to support a global search at an early stage of the optimisation, and a local search thereafter. In [182], a GP with smart mutation was developed to discover and evolve agents for extracting regions-of-interest (ROIs) in remotely sensed images. The mutator works by deleting internal nodes that decrease the overall fitness and keep promising ones. Kateb et al [183], developed a sputnik (an elitist mutator), which relies on selection based on a continuous learning of past effects on fitness functions, aiming to enhance the algorithm by guiding the search towards a faster trade-off to save time and number of fitness evaluations. While improving the EA’s performance and obeying all operational constraints, the smart mutation operator developed in this thesis targets mutation on genes according to their respective contributions to the cost function and penalties violations.

2.10 Summary

Power system studies remain highly active research area, with research interests in the field growing on a daily basis because of the high importance of electricity. This chapter explored key concepts related to the research, including: use of optimisation as a problem solving tool, electrical energy and electrical power, (identifying power generation as the primary domain of the work), concept of optimal power flow leading to economic load dispatch (ELD) problems. It presented the various formulations of ELD problems, including: smooth and non-smooth fuel cost functions, multiple fuels, ramp rate limits and prohibited operating zones constraints; and chronicled the historical developments in the solution approaches of ELD problems, consisting of mathematical programming methods, artificial intelligence methods and hybrid methods, with a review of related work in those approaches. Despite the advances in these solution approaches, there are still potential areas for further investigative study. The introduction of hybrid methods to solving ELD problems is a novel approach that is yet to be fully and exhaustively harnessed.

Two major categories of ELD problems (SELD and DELD) were identified, which constitute distinct problem areas in the research.

This chapter also presented a brief survey of evolutionary algorithms, the optimisation approach used in the research. It described the design and implementation details of a standard EA, justified its suitability to solving ELD problems, and reviewed recent and related work in power systems optimisation. Lastly, it discussed the concepts of multi-objective EAs, non-evolutionary search algorithms and a brief review of aspects related to smart mutation.

Chapter 3

A New Evolutionary Algorithm for the Static Economic Load Dispatch Problem

SELD problems handle a single load optimisation period (e.g. of one hour duration), in which the variables (generator outputs) do not vary with time. In this chapter, we describe and evaluate improved EA approaches for realising optimal solutions for SELD problems in the electricity generation industry. The methods combine a standard EA with "smart mutation". The key aspect of the method that improves performance is the smart mutation operator, which targets mutation on genes according to their respective contributions to the cost function. We consider benchmark instances of the economic load dispatch problem, which involve minimum/maximum generation limits, power balance, ramp rates and prohibited operating zones. Violation of either of these constraints introduces the concept of penalties, and these in turn provide the basis for the smart mutation operator. Our "Smart" EA (SEA) was compared with a Basic EA (BEA), and with reported results for other recent algorithms, on three benchmark cases involving 6, 15 and 20 generating units.

3.1 Design Methodology

Optimisation techniques for scientific and engineering systems involve the process of solving a set of non-linear equations describing the optimal or steady-state operation of the systems. The ELD problem is formulated as minimising a scalar objective function through the optimal operation of a vector of control parameters. This is mathematically illustrated in (3.1), (3.2) and (3.3); and diagrammatically represented in Figure 3.1.

Minimise: $C\,(x, u)$ (3.1)

Subject to:

$G\,(x, u) = 0$ (3.2)

$H\,(x, u) \leq 0$ (3.3)

Where:

C = Cost function

x = vector of dependent variables (generating cost);

u = vector of control variables (generator outputs);

G(x, u) = Set of non-linear equality constraints (power balance);

H(x, u) = Set of inequality constraints (limits in generators outputs).

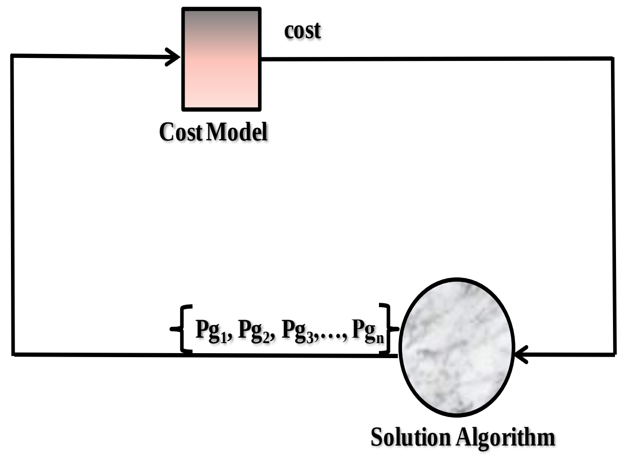

Figure 3.1: Diagrammatic representation of the design approach

The approach consists of two main components: the controlling device and the device to be controlled. A typical ELD problem is one that involves the optimal set of generating units. This minimises the operation cost (mainly fuel cost). The device to be controlled is the generating cost model, while the solution algorithm (EA) is the controlling device. A vector of generators output is fed into the cost model, which produces a scalar cost of generating those outputs. The cost is passed on to the solution algorithm to be minimised, an iterative process which continues until either a cost lower than acceptable minimum is found or number of maximum iterations is completed.

The approach we took in this paper was to build into the EA a straightforward way of adapting the search process to the search landscape, by targeting the mutation operation towards genes that had the most negative influence on the cost function. Such “smart” mutation has been used successfully in a variety of domains in which it is possible to relate components of the cost function to particular components of the genome. Figure 3.2 shows a flow diagram for the SELD optimisation using EA.

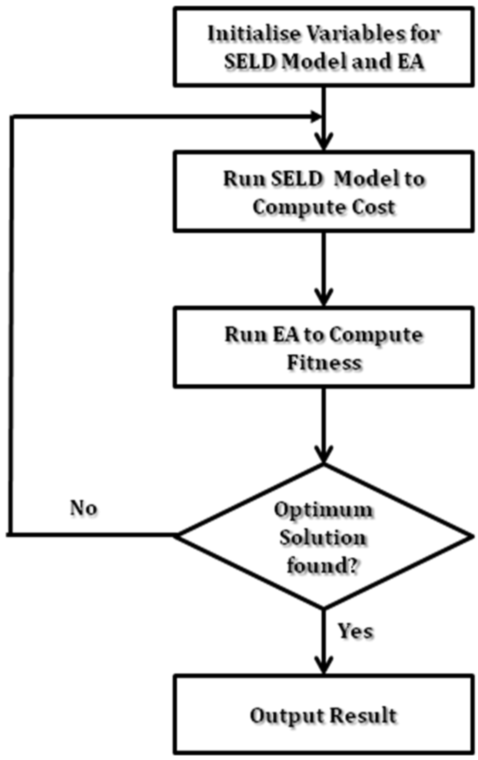

Figure 3.2: Flow diagram for SELD optimisation

3.2 Problem Formulation

The SELD is an optimisation task whose goal is to find the optimal combination of online power generators that will minimise the total fuel cost to meet the total system's load demand while satisfying various equality and inequality constraints. This is done over an appropriate short-term period, usually one hour. The constraints in a practical generator include minimum and maximum generation limits, power balance, ramp rate limits and prohibited operating zones. For a thermal generating station, the unit fuel cost is shown in the quadratic form, and the goal is to minimise the total fuel cost as in (2.11) and (2.12), subject to the generating limit constraint of (2.14), power balance constraint of (2.15), ramp-rate limits constraints of (2.21), and prohibited operating zones constraints of (2.22).

3.3 The Optimisation Approach

A key task in an optimisation process is constraints handling [33, 54]. Most algorithms were initially conceived to solve unconstrained problems. Therefore, application of the appropriate algorithm to solving constrained problems such as the present work involves various techniques of handling constraints to keep the control variables in feasible regions where all constraints are satisfied. A technique for handling these constraints constitutes the "smartness" within our proposed EA approach to this problem.

3.3.1 Basic (Conventional) Evolutionary Algorithm

EAs are popular heuristic methods for realising solutions of unconstrained as well as constrained optimisation problems. Loosely reflecting Darwin's evolutionary theory [63], an EA uses selection, crossover and mutation processes to create an environment where population of individuals (chromosomes) compete with one another, and only the fittest move from the current generation to the subsequent ones. Realisation of solutions to optimisation problems using conventional EAs, herein referred to as Basic EA (BEA) involves the following steps:

1. *Generate initial population of solutions;*
2. *Fitness evaluation of each member of the population;*
3. *Form breeding pair by selecting parent solutions;*
4. *Recombination – perform crossover and mutation on selected parent solutions;*
5. *Replace parent population with the new (child) population*
6. *If stopping criterion is not reached, go back to step 2.*

A BEA initialises the population by choosing each solution at random. An evaluation determines the fitness value of each solution (called chromosome) of the population. Each chromosome contains a number of genes. A value encoding method is used consisting of double numbers chromosome genes, representing the generating units within a plant. Selection chooses parent solutions, based on their fitness values to go into breeding. Crossover exchanges some of the genes between breeding parents. This occurs with a probability called *crossover probability*, usually greater than 50%. Mutation is applied to the resulting solution which changes the value of one or more genes, thereby introducing genetic variation between the parents and children. Mutation also occurs with a probability called *mutation rate*, which is usually small, resulting in typically one gene changed per chromosome. The resulting (child) population replaces the parent population. This process continue until a stopping criterion is reached – either a maximum number of functional evaluation is completed, or an acceptable cost value is found.

3.3.2 Constraints Handling

The bulk of optimisation task done in an evolutionary algorithm is constraint handling. The random initialisation of chromosomes sets the genes (outputs of generating units) of each chromosome in the initial population to random double numbers representing the generators' outputs, between their minimum and maximum generating limits. During fitness evaluation (computation of the cost function) of each chromosome, various checks are performed to ensure that the units' outputs obey all operational constraints – remain within the generation limits, maintain power balance, ramp-rates limits and avoid the prohibited operating zones. Violation of any of the constraints constitutes a penalty, which augments the initial objective cost function of (2.11) to form the fitness function shown in (3.4).

$$Min\, C_T = \begin{cases} C_T \\ C_T + total\_penalty \end{cases} \tag{3.4}$$

This helps guide the search process towards repairing that solution. We describe here how the four operational constraints identified in section 3.2 are handled.

(i) ***Generating Limit:*** A random small positive number less than one (scaling factor), ensures that the value of genes (generating units' outputs) are within the legal minimum and maximum limits, according to (3.5):

$$Pg_i = Pg_i^{\min} + Math.random() * (Pg_i^{\max} - Pg_i^{\min}) \tag{3.5}$$

(ii) ***Power Balance:*** If the equality constraint of (2.15) is not satisfied, a penalty factor, q_1, is used to normalise and maintain an overall power balance, using (3.6), ensuring that the terms in the bracket equal to zero:

$$Penalty\_pb = q_1 \left(\sum_{i=1}^{N} Pg_i - P_D - P_L \right) \tag{3.6}$$

(iii) ***Ramp-rates Limit:*** The power output of a unit at the current time depends on the output at the previous time, ramp up value (*UR*) and ramp down value (*DR*) of the generator. The power output of a unit due to the ramp-rates limit is defined as follows:

$$Pg_{rr\lim} = \begin{cases} Pg_{i,t-1} - DR_i, & Pg_{i,t} < Pg_{i,t-1} - DR_i, \\ Pg_{i,t-1} + UR_i, & Pg_{i,t} > Pg_{i,t-1} + UR_i, \\ Pg_{i,t}, & otherwise \end{cases} \tag{3.7}$$

If the inequality constraint of (2.21) is not satisfied, a penalty factor, q_2, is assigned to the affected units outside the feasible regions, using (3.8):

$$Penalty\_Pg_{rr\lim} = q_2\left(\sum_{i=1}^{N} Pg_i - Pg_{rr\lim}\right) \tag{3.8}$$

(iv) ***Prohibited Operating Zones:*** The prohibited operating zone penalty function counts the number of units that fall within such prohibited zones, according to the following rules:

$$PZ_{i,k} = \begin{cases} 1, & if\ Pg_i\ violates\ prohibited\ zone \\ 0, & otherwise \end{cases} \tag{3.9}$$

If the prohibited zones are not violated, the ELD is solved in a straightforward process, otherwise, a number '1' is assigned to such occurrence, and the constraint is processed using (3.10):

$$Penalty\_pz = q_3 \sum_{i=1}^{N} PZ_{i,k} \tag{3.10}$$

Where: q_3 = penalty factor, k = number of prohibited zones in i, N = number of generating units.

3.3.3 The Smart Evolutionary Algorithm

The approach of capturing gene-specific contributions to costs, and using this information to help target the mutation operator, is at the heart of what we call a 'smart evolutionary algorithm' (SEA), and we use this approach in the current and later chapters on a series of specific optimisation problems. The method basically combines *a* *standard evolutionary algorithm* with *a smart mutation algorithm*. In the standard evolutionary algorithm of section 3.3.1, single mutation per chromosome is used. Occurring at a low probability called 'mutation rate', it chooses an allele at random and changes it to a new value. It works as a background of the EA in introducing new genetic materials in the population by randomly modifying some building blocks, maintains population diversity, and helps solutions escape local optima. This mutation scheme has severally been used and shown in the literature to produce near-optimal results [191], including a real-life virology application [192]. The basic pseudo-code for SEA is given as follows:

START

DEFINE parameters and INPUT data

- *Cost coefficients of the generators (a, b, c)*
- *B-matrix coefficients*
- *Load demands (P_D)*
- *Generating limits (Pg_{min} and Pg_{max})*
- *Ramp-rate limits (Pg_{rrlim})*
- *Prohibited zones (PZ)*

INITIALIZE a random population of chromosomes

FOR i = 1 to N (N = No of Generating Units)

$$Pg_i = Pg_{i\_\min} + Math.Random * (Pg_{i\_\max} - Pg_{i\_\min})$$

END FOR

EVALUATE Objective function (For each chromosome in the population)

FOR i = 1 to N

$$f(Pg_i) = a_i Pg_i^2 + b_i Pg_i + c_i$$

Compute power loss

Check for constraints violation in power balance

Check for constraints violation in generation and ramp-rates limit

Check for constraints violation in prohibited operating zones

Violation of a constraint leads to penalty treatment, defined as:

$$Min\ F_c = \begin{cases} f(Pg_i) \\ f(Pg_i) + penalty(Pg_i) \end{cases}$$

IF no constraint is violated,

Penalty = 0

Fitness function = Objective function

ELSE

Fitness function = Objective function + Penalty

END IF

END FOR

WHILE (stopping criteria is not reached)

From the entire population

SELECT parent pairs for breeding

CROSSOVER the genes of selected parents

Create an array to return children
Crossover parents to create children
Return the children in an array
*Perform **SMART MUTATION***
EVALUATE
REPLACE parent population with children population
Perform ELITISM (Keep a percentage of best individuals)
FOR i = 1 to Population size
Sort Chromosomes according to their fitness (highest to lowest)
Calculate Number of Elites (based on elitism rate)
Copy Elites onto next Generation
END FOR
END WHILE
OUTPUT (The optimum/best compromising solution vector)
FOR i = 1 to N

$$Pg = [Pg_{1,}\ Pg_{2,}\ Pg_{3,} \ldots Pg_N]$$

END FOR
Output Total Generation = Sum of all Generators' Output
Output Total cost = Sum of Costs of generation
STOP.

3.3.4 The Smart Mutation

Of the four constraints identified in section 3.3.2, generating limit, ramp-rates limit and prohibited operating zone constraints are local to each unit; while power balance constraint is global, applying to all the generating units. The smart mutation operator focuses mutation on units contributing most to cost and violates either or both of the local constraints – ramp-rates limit and prohibited operating zones (where available in a given problem case). The use of a scaling factor (a randomly generated small positive number between 0 and 1) ensures that the outputs of all the generation units are within feasible/legal regions; hence the local generating limits constraint is not violated. One of the ways to investigate the overall performance of an algorithm is by evaluating its constraints handling capabilities (described in section 3.3.2), which also contributes to the 'smartness' of the algorithm. The following is a pseudo code of the smart mutator.

START
INITIALIZE costs
Calculate the costs produced by all the units
SET cost of the first unit (i = 1) as the highest cost
FOR i = 2 to N (where N = number of generating units)
IF (cost[i] > highest cost)
Highest cost = cost[i]

Highest position = i

END IF

Pg_i *= value of unit at highest position*

Mutate this Pg_i *by subtracting a small random deviation from the unit:*

Pg_i *(new) =* Pg_i *– (Math.random () * * Pg_i*)*

(Where: Math.random () is a random number between 0 and 1, e.g. 0.2)

Replace Pg_i *with* Pg_i *(new)*

//Check for violation of local constraints

//Ramp-rates limits violation:

IF Generation decreases

$Pg_i = (Pg_{i-1} – DR)$

IF Generation increases

$Pg_i = (Pg_{i-1} + UR)$

(Where: Pg_i *= current value;* Pg_{i-1} *= previous value; UR = up-ramp rate limit)*

ELSE

$Pg_i = Pg_{i-1}$

END IF

END IF

IF (Pg_i *< Max (lower limit, (*Pg_{i-1} *– DR))) OR (*Pg_i *> Min (upper limit, (*Pg_{i-1} *+ DR)))*

(Where: lower limit and upper limit = legal lower and upper limits of the unit)

Pg_i *violates ramp-rates limit*

Mutate this Pg_i *by subtracting a small random deviation from the unit:*

Pg_i *(new) =* Pg_i *– (Math.random () ** Pg_i*)*

END IF

Replace Pg_i *with* Pg_i *(new)*

//Prohibited operating zones violation

FOR k = 1 to z_i *(where:* z_i *= number of prohibited zones of unit, i)*

IF ($Pg_i > Pg^l_{i,k}$*) AND (*$Pg_i < Pg^u_{i,k}$*)*

(Where: $Pg^l_{i,k}$ *and* $Pg^u_{i,k}$ *are the lower and upper bounds of the* k^{th} *prohibited zone)*

Mutate this Pg_i *by subtracting a small random deviation from the unit*

Pg_i *(new) =* Pg_i *– (Math.random () ** Pg_i*)*

Assign the number "1" for every Pg_i *that violates prohibited operating zone*

(i.e. counting every occurrence)

END IF

END FOR

Replace Pg_i *with* Pg_i *(new)*

//Ensure that all the units' outputs are within feasible limits (Generating limit constraint):

$Pg_i = Pg_{i\_min} + (scaling\_factor * (Pg_{i\_max} – Pg_{i\_min}))$

END FOR

STOP

The total number of units selected for mutation in addition to the highest cost producing one depends on the number of local constraints violated. Therefore a minimum of one and maximum of three genes are involved. Perhaps in some problem cases, either or both ramp-rates limit and prohibited operating zones constraints may not be present. In some other cases, the unit with the highest cost may also violate a constraint, while in some other cases, no constraint may be violated. But where violation exists, their numerical (penalty) values – (3.8) and (3.10) for ramp-rates limit and number of prohibited operating zones violated, respectively; alongside the global load balance penalty value (3.6) augment the problem's objective function to form a generalised fitness function.

The magnitude of the constraints as well as the cost produced by each of the units contributes to the overall fitness. As the penalties values gradually reduce to 0, total cost equals total fitness, which is the optimal value. By targeting the unit with the highest cost, the mutation operator attempts to minimise the total cost of producing an optimal power. We found out that reducing the values of those units reduces over generation of power, and consequently minimises power loss. Mutating the units that violate local constraints has a tendency of forcing them into the feasible regions. Besides, the small random deviation subtracted from the units ensures that the changes introduced are not too significant, thereby distorting the generating unit.

3.3.5 SEA Variants

Several variants of the smart mutator are possible. In this thesis, we investigated and implemented the following three smart evolutionary algorithms: SEA1, SEA2 and SEA3; resulting from three distinct variants of the smart mutator. We define and describe briefly the implementation differences between the three variations.

3.3.5.1 SEA1

This SEA variant involves the use of tournament selection based on the penalty values (costs and numerical values of the constraints) to decide which gene to mutate. It works in direct analogy to the tournament method of selecting parents from an initial population to go into breeding in order to generate an offspring during an evolutionary process (see section 2.7.5). But while the EA selection method chooses individuals or chromosomes (the entire generating units) from the population (uniformly at random, with replacement), and the fittest of those individuals is the one returned as selected, the SEA1 chooses genes from the chromosome, by 'targeting' those with the highest operating cost and violate constraint(s). The less-fit units are selected in this case, having the highest cost of generation, and falling outside the feasible range respectively. This is the basis of the new chromosome, a potential solution to the problem. The number of units selected (tournament size) as well as the size of the problem case has combined effect on the selective pressure. A tournament size of 1, 2 or 3, out of 6 units has a higher selective pressure, than about same tournament size out of 20 or 40 units.

3.3.5.2 SEA2

In this variant, there is a mutation probability, called 'smart mutation probability', different from the normal 'mutation rate' of an EA. The smart mutation probability is used to bias the mutation operation, such that when the probability is met, smart mutation is done; otherwise, a 'single-gene, uniform random mutation' is done (that is, the mutated unit is randomly selected, and not targeted on the basis of the penalty values). This gene-specific mutation probability in SEA2 is a random number between '0' and '1'. In contrast with the 'mutation rate' which usually occurs with a much lower probability, and aims to maintain diversity in the entire population of individuals; the 'smart mutation probability' is a fixed, higher-level probability operator that targets a particular gene, based on the criteria of merit (high fitness value, constraint violation, etc); uniformly or non-uniformly, deterministically and adaptive [187, 190].

3.3.5.3 SEA3

This EA variant of the smart mutator is an extension of SEA2, but instead of having a fixed value of the smart mutation probability, it is 'computed' (as the ratio of generation number over the maximum generation). Therefore, the mutation probability value starts at '0', and gradually moves to '1.0' in a linear fashion as we get to the maximum number of generations. This is a simple linear adaptation that works by linearly increasing the smart mutation probability from beginning to end of an optimisation run.

3.4 Experiments and Simulation Results

We implemented a Basic EA (BEA) and three Smart EAs (SEA1, SEA2 and SEA3), from the three smart mutation variants, and used them to carry out detailed experimentations to investigate the efficiency, validity and robustness of the algorithms in three different problem areas involving 6, 15 and 20 generators.

3.4.1 Test Case I: 6 Generators

In [54], the conventional/SELD problem was solved using Differential Evolution (DE), allocating power output to 6 thermal generators, taking into account the effects of transmission losses. Using the data from this paper, we carried out several initial experiments aimed at selecting appropriate values for the genetic parameters - crossover rate, population size, tournament size and mutation rate. Total load demand was set at 283.40MW; the detailed parameters are given in Table 3.1, and the loss coefficient B-matrix given in (3.11).

Units	Pg_{min}	Pg_{max}	***a***	***b***	***c***
1	50	200	0.00375	2.00	0
2	20	80	0.01750	1.75	0
3	15	50	0.06250	1.00	0
4	10	35	0.00834	3.25	0
5	10	30	0.02500	3.00	0
6	12	40	0.02500	3.00	0

Table 3.1: Generators' data for the 6-unit problem from [54]

$$\begin{bmatrix} 0.0002 & 0.00001 & 0.000015 & 0.000005 & 0.0 & 0.00003 \\ 0.00001 & 0.0003 & -0.00002 & 0.000001 & 0.000012 & 0.00001 \\ 0.000015 & -0.00002 & 0.0001 & -0.00001 & 0.00001 & 0.000008 \\ 0.000005 & 0.000001 & -0.00001 & 0.00015 & 0.000006 & 0.00005 \\ 0.0 & 0.000012 & 0.00001 & 0.000006 & 0.00025 & 0.00002 \\ -0.00003 & 0.00001 & 0.000008 & 0.00005 & 0.00002 & 0.00021 \end{bmatrix} \tag{3.11}$$

B_0 and B_{00} were neglected. Based on available data in this problem case, the main constraints which constitute the source of penalties are generating limits and power balance constraints. Augmenting the penalty function to the original objective cost function of (2.11) yields the generalised fitness function given by (3.12).

$$C_T = \sum_{i=1}^{N} C_i + q_1 \sum_{i=1}^{N} (Pg_i - P_D - P_L)^2 \tag{3.12}$$

Where: q_1 is a penalty factor which normalises the power balance, assigning a high cost of penalty to affected ones far from the feasible region [54, 101, 108]. The rule defined in (3.5) ensures that outputs of the generating units are within the legal minimum and maximum limits. Starting with SEA1, Tables 3.2, 3.3 and 3.4 summarise the results of different values for crossover rate, population size and tournament rate, averaged over 30 runs. As in SEA1, SEA2 operates a single mutation per solution of rate 0.01 in addition to the smart mutation probability of 0.6, tuned in Table 3.5, averaged over 30 runs. Arising from our findings on the tuning of EA parameters in section 2.7.8, we kept the mutation rate as low as 0.01, being a background operator, to provide a reasonably good diversity in the population, and avoid distortion of the building blocks. The resources allocations for crossover rate, population size, tournament size, and smart mutation probability (for SEA2) in the best solutions out of the 30 runs are respectively shown in Tables 3.6, 3.7, 3.8 and 3.9. This is in terms of realising the two main goals of the dispatch – lower cost of generation, and meeting load demand.

Crossover Rate	***0.6***	***0.7***	***0.8***	***0.9***
Av Cost	769.59	**758.19**	779.28	774.97
Std Dev	15.41	12.25	12.78	18.04

Table 3.2: Summary of results for different crossover rates, averaged over 30 runs of SEA1 algorithm on the 6-unit problem [54]

Pop Size	***10***	***20***	***30***	***40***	***50***	***100***	***150***
Av Cost	758.19	755.66	756.01	757.55	748.07	**738.73**	740.21
Std Dev	12.25	15.30	18.66	12.82	11.02	9.28	12.36

Table 3.3: Summary of results for different population sizes, averaged over 30 runs of SEA1 algorithm on the 6-unit problem [54]

Tournament Size	***2***	***4***	***6***	***8***	***10***
Av Cost	**738.73**	742.22	741.01	741.64	744.64
Std Dev	9.28	11.00	10.45	13.28	12.77

Table 3.4: Summary of results for different tournament sizes, averaged over 30 runs of SEA1 algorithm on the 6-unit problem [54]

Smart Mutation Probability	***0.1***	***0.2***	***0.3***	***0.4***	***0.5***	***0.6***	***0.7***	***0.8***	***0.9***
Av Cost	757.83	759.09	758.23	761.37	752.58	**749.54**	765.32	770.89	765.94
Std Dev	15.07	15.62	14.56	13.92	15.33	**13.34**	13.24	14.51	16.88

Table 3.5: Summary of results for different smart mutation probabilities, averaged over 30 runs of SEA2 algorithm on the 6-unit problem [54]

Units	*Crossover Rate*			
	0.6	***0.7***	***0.8***	***0.9***
1	173.15	168.89	120.10	95.48
2	20.77	24.02	68.39	56.82
3	19.41	48.30	31.12	77.99
4	19.48	10.91	13.31	30.84
5	21.57	15.56	21.22	25.67
6	29.21	16.58	30.03	26.79
Total Gen	283.61	284.29	284.18	283.58
Total Cost	776.22	**715.62**	765.32	743.40
Loss	0.21	0.89	0.78	0.18

Table 3.6: Resources allocation for different crossover rates in the best of 30 runs of SEA1 algorithm on the 6-unit problem [54]

Units	*Population Size*						
	10	***20***	***30***	***40***	***50***	***100***	***150***
1	168.89	111.19	159.80	154.80	135.25	151.31	161.21
2	24.02	76.03	25.95	34.44	33.69	24.69	27.39
3	48.30	48.64	45.48	44.51	46.69	48.89	45.61
4	10.91	15.58	17.44	27.63	33.44	23.45	19.87
5	15.56	11.33	12.57	13.63	15.36	15.69	17.66
6	16.58	21.08	22.92	13.62	20.83	19.85	15.25
Total Gen	284.29	283.84	284.14	288.62	284.66	283.88	286.98
Total Cost	715.62	730.56	713.66	724.54	720.60	**709.60**	719.72
Loss	0.89	0.44	0.74	5.22	1.26	0.48	3.58

Table 3.7: Resources allocation for different population sizes in the best of 30 runs of SEA1 algorithm on the 6-unit problem [54]

Units	*Tournament Size*				
	2	***4***	***6***	***8***	***10***
1	151.31	164.27	122.69	141.82	120.72
2	24.69	26.66	43.06	40.04	65.25
3	48.89	46.14	49.57	49.55	49.44
4	23.45	18.27	30.23	17.93	20.40
5	15.69	15.43	13.37	15.45	16.07
6	19.85	14.58	24.56	20.91	12.57
Total Gen	283.88	285.35	283.42	285.69	284.44
Total Cost	**709.60**	711.74	713.65	719.00	715.55
Loss	0.48	1.95	0.02	2.29	1.04

Table 3.8: Resources allocation for different tournament sizes in the best of 30 runs of *SEA1* algorithm on the 6-unit problem [54]

Units	*Smart Mutation Probability*								
	0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1	128.96	137.99	151.71	152.54	133.95	168.99	131.30	158.78	167.57
2	51.11	29.72	22.55	34.07	47.37	30.54	55.50	50.44	35.77
3	49.49	49.95	44.18	34.97	41.67	40.75	27.92	22.47	24.03
4	12.73	11.54	20.18	22.78	17.48	11.68	19.62	17.17	29.20
5	26.51	25.51	16.84	19.01	21.62	15.76	23.38	18.55	14.39
6	17.44	29.86	28.33	20.55	21.36	15.71	25.92	16.08	13.92
Total Gen	286.23	284.58	283.79	283.92	283.44	283.44	283.63	283.50	284.88
Total Cost	719.98	723.70	726.12	731.53	721.24	711.86	726.39	737.76	740.02
Loss	2.83	1.18	0.39	0.52	0.04	0.04	0.23	0.10	1.48

Table 3.9: Resources allocation for different smart mutation probabilities in the best of 30 runs of SEA2 algorithm on the 6-unit problem [54]

From these initial experiments, tuned values for population size, tournament size, crossover rate, and smart mutation probabilities were respectively determined from the lowest costs realised with the parameters' values, given in Table 3.10, alongside with a fixed number of generations (of 100, to ensure the same number of fitness evaluations, penalty factor, q_1= 500000, scaling factor, α = 0.1, as in [54]) and elitism rate, which were used for the main experiment involving SEA1, SEA2 and SEA3, with results shown in Table 3.11.

Parameters	***Values***
Population size	100
Tournament size	2
Crossover rate	0.7
Mutation rate	0.01
Mutation probability (in SEA2)	0.6
No of Generations	100
Elitism Rate	10%
No. or runs	30

Table 3.10: Experimental parameters and values for the 6-unit problem

	SEA1	***SEA2***	***SEA3***
Ave Cost	738.73	749.54	744.49
Std Dev	13.74	13.34	10.57
Min Cost	709.60	711.86	710.13
Max Cost	759.20	766.97	758.97

Table 3.11: Summary of results, averaged over 30 runs in each of SEA1, SEA2 and SEA3 approaches on the 6-unit problem [54]

To further demonstrate the efficiency of the SEAs, we consider the distribution pattern of the best solutions in each of the 30 runs as shown in Figure 3.3, which alongside Table 3.11 shows that the range of variation of the costs from each independent run is relatively small, and equally distributed between the minimum and maximum costs.

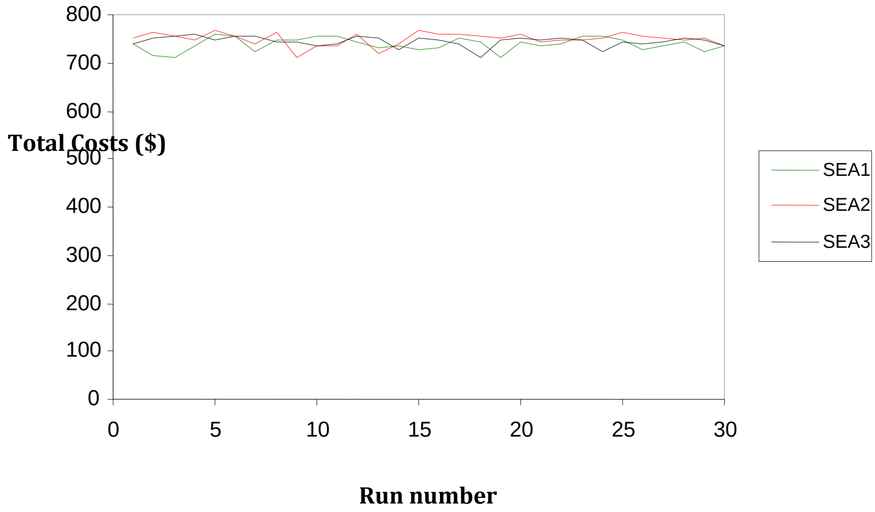

Figure 3.3: Distribution of generation costs for *SEA1*, *SEA2* and *SEA3*, averaged over 30 runs on the 6-unit problem [54]

The results of SEA1, SEA2 and SEA3 were compared with our BEA [34], and previously published results from other algorithms that have been tested on the same problem case, namely: Differential Evolution [54], Genetic Algorithm [36][54], Successive Linear Programming [36][94] and the Quasi-Newton Method [36]. Table 3.12 summarises the comparison results based on the resources allocation to the units from the best of 30 independent runs of algorithms BEA, SEA1, SEA2 and SEA3 (the number of runs for DE, SLP, GA and QN were not reported). This shows superior performance of the three SEAs in terms of both lower generation costs and lower power losses. All the three smart mutation variants performed very well on this problem, having low total cost of generation ($709.60/h, $711.86/h and $710.13/h) respectively for SEA1, SEA2 and SEA3; and reduced power losses of 0.48MW, 0.048MW and 1.0MW respectively. The algorithms converged in less than 30 generation (out of 100), probably because of the relatively small size of the problem case. Although the population size of the SEAs was set to 100 following the parameters tuning (giving them higher number of fitness evaluations), while DE was set to 26 [54], and GA to 48 [36], the results of Table 3.3 shows that during the tuning, with population size of 20, SEA1 still had a better average result of $755.66/h, than both DE and GA ($803.07/h and $803.69/h respectively). The experimental conditions of SLP and QN which were used to compare with DE and GA were not reported in literature. It is also to be noted that the experimental parameters and conditions for BEA, although same as those of SEA1, SEA2 and SEA3, but had less tuning than SEA1, SEA2 and SEA3.

Units	*SEA1*	*SEA2*	*SEA3*	*BEA [34]*	*DE [54]*	*SLP [54]*	*GA [54]*	*QN [54]*
1	151.31	168.99	130.44	171.58	177.51	175.25	179.37	170.24
2	24.69	30.54	41.12	49.26	48.61	48.34	44.24	44.95
3	48.89	40.75	49.63	22.63	20.91	21.21	24.61	28.90
4	23.45	11.68	27.55	21.20	21.64	23.60	19.90	17.48
5	15.69	15.76	19.45	12.73	12.47	12.25	10.71	12.17
6	19.85	15.71	61.21	14.17	12.02	12.33	14.09	18.47
Total Gen	283.88	283.44	284.40	292.11	293.16	292.98	292.92	292.21
Total Dem	283.40	283.40	283.40	283.40	283.40	283.40	283.40	283.40
Loss	0.48	0.04	1.00	8.71	9.76	9.58	9.52	8.81
Total Cost	709.60	711.86	710.13	801.30	803.07	803.08	803.69	807.78

Table 3.12: Resources allocation in the best of 30 runs of *SEA1*, *SEA2* and *SEA3*, and comparison with other approaches on the 6-unit problem

We investigate the overall performance by comparing the penalties handling capabilities of BEA, SEA1, SEA2 and SEA3 as shown in the costs and penalties convergence characteristics of Figures 3.4, 3.5, 3.6 and 3.7. The vertical (fitness) axis of these curves refers to the values of the total cost and penalty in the average of 30 runs in each of the four algorithms. Starting with randomly generated populations at generation 0, the values of the costs and penalties gradually converge smoothly to the respective optima as generation increases in each of the four algorithms. There is a great similarity in the trend of cost curves of BEA and SEA1 (Figures 3.4 and 3.5), but the major difference is in the penalty curves. It can therefore be concluded that BEA is not very efficient in reducing penalty violations, unlike SEA1, SEA2 and SEA3. This contributes to the ‘smartness’ of the algorithms. As the value of the total penalties gradually reduce to 0.0, total cost equals total fitness, which is the optimal value. Moreover, the best results from BEA have a considerably higher total cost than SEA1, SEA2 and SEA3.

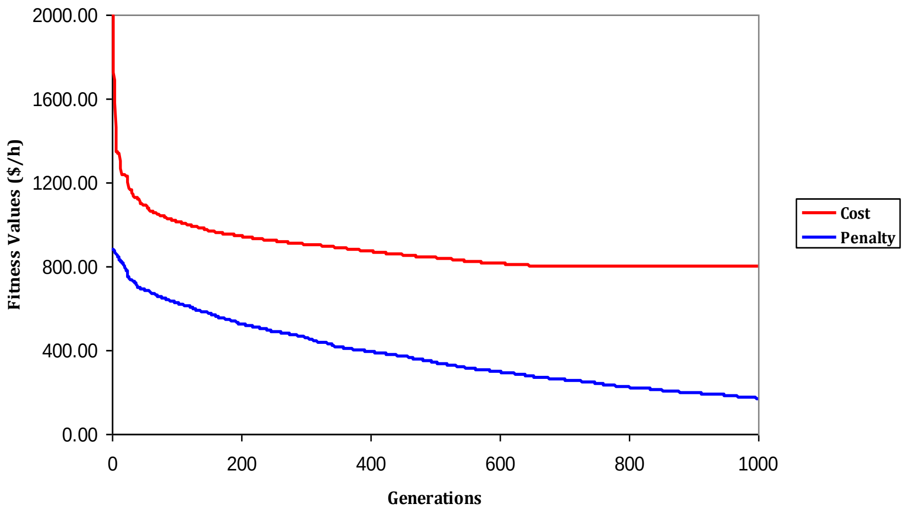

Figure 3.4: Cost and penalty convergence characteristics for an average of 30 runs of BEA on the 6-unit problem

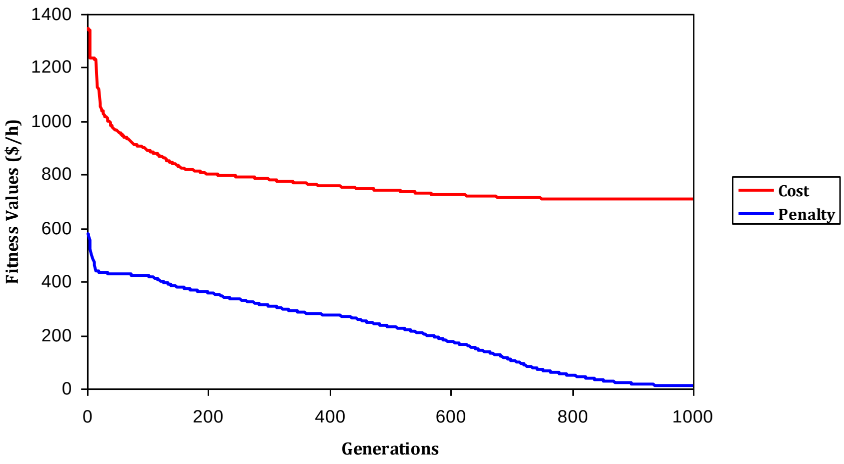

Figure 3.5: Cost and penalty convergence characteristics for an average of 30 runs of SEA1 on the 6-unit problem

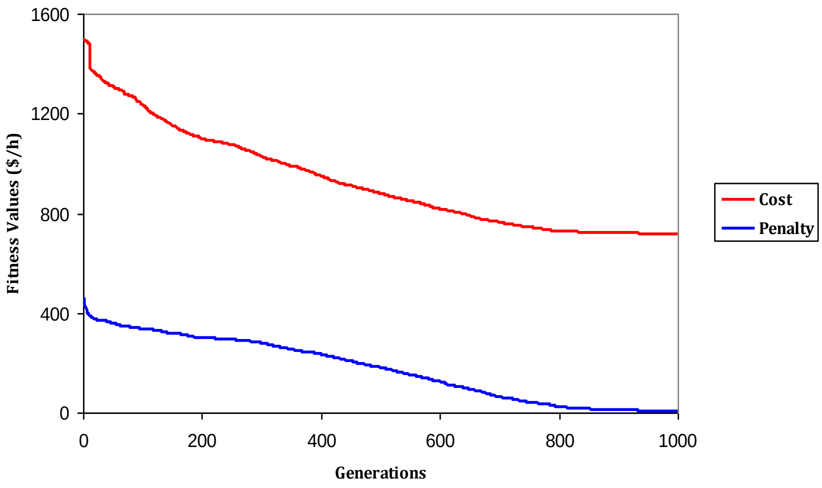

Figure 3.6: Cost and penalty convergence characteristics for an average of 30 runs of SEA2 on the 6-unit problem

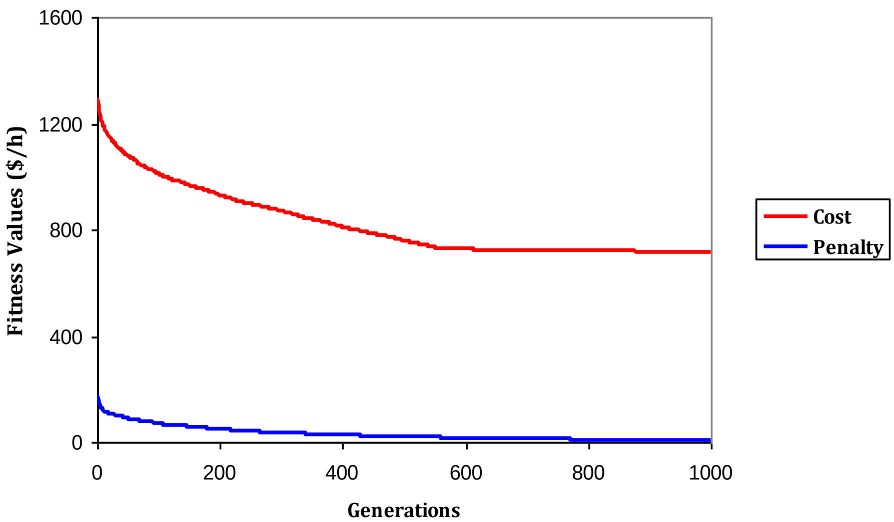

Figure 3.7: Cost and penalty convergence characteristics for an average of 30 runs of SEA3 on the 6-unit problem

SEA3 shows the most active performance in reducing penalty violations, but not the best overall performance on the present test problem. This may be because the problem is relatively small, and the evidence suggests that SEA3 may show its value on larger scale test problems. However, the generation cost of $709.60/h from the best resources allocation of SEA1 (Table 3.12) is the lowest we have seen in the literature to date for this problem, while meeting load demand.

3.4.2 Test Case II: 15 Generators

In [33, 65], the SELD problem was solved using improved Particle Swarm Optimisation (PSO) based on Gaussian probability distribution/chaotic sequences and a simple but efficient Evolutionary Strategy (ES) respectively, to determine the optimal combination of power outputs of 15 generators that will minimise the total fuel cost, satisfying power demand, ramp-rates limits and prohibited operating zones violation. The total load demand was 2630MW. Table 3.13 shows the parameters for generating unit capacity and cost coefficients, while data for ramp rates and prohibited zones are shown in Table 3.14. The B-matrices loss coefficient is taken from [33]. In this problem case, the constraints which constitute the source of penalties are generating limits, power balance, ramp-rates and prohibited operating zones violation constraints. Augmenting the penalty function to the original cost function of (2.11) yields the generalised fitness function given by (3.13). The rule defined in (3.5) ensures that outputs of the generating units are within the legal minimum and maximum limits.

Units	Pg_{min}	Pg_{max}	a	b	c
1	150	455	0.000299	10.1	671
2	150	455	0.000183	10.2	574
3	20	130	0.001126	8.8	374
4	20	130	0.001126	8.8	374
5	150	470	0.000205	10.4	461
6	135	460	0.000301	10.1	630
7	135	465	0.000364	9.8	548
8	60	300	0.000338	11.2	227
9	25	162	0.000807	11.2	173
10	25	160	0.001203	10.7	175
11	20	80	0.003586	10.2	186
12	20	80	0.005513	9.9	230
13	25	85	0.000371	13.1	225
14	15	55	0.001929	12.1	309
15	15	55	0.004447	12.4	323

Table 3.13: Generators' data (limits and cost coefficients) for the 15-unit problem

Units	P_0	*UR*	*DR*	*Prohibited Zones*		
				Zone 1	*Zone 2*	*Zone 3*
1	400	80	120			
2	300	80	120	[185 255]	[305 335]	[420 450]
3	105	130	130			
4	100	130	130			
5	90	80	120	[180 200]	[305 335]	[390 420]
6	400	80	120	[230 255]	[365 395]	[430 455]
7	350	80	120			
8	95	65	100			
9	105	60	100			
10	110	60	100			
11	60	80	80			
12	40	80	80	[30 40]	[55 65]	
13	30	80	80			
14	20	55	55			
15	20	55	55			

Table 3.14: Generators' data (ramp-rates and prohibited zones) for the 15-unit problem

$$C_T = \sum_{i=1}^{N} C_i + q_1 \sum_{i=1}^{N} (Pg_i - P_D - P_L)^2 + q_2 \left(\sum_{i=1}^{N} Pg_i - Pg_{rr\lim} \right)^2 + q_3 \sum_{i=1}^{N} PZ_{i,k} \tag{3.13}$$

Where: Pg_{rrlim} and $PZ_{i,k}$ are as defined in (3.7) and (3.9) respectively; q_1, q_2 and q_3 are penalty factors for the power balance, ramp-rates and prohibited operating zones violations respectively. The values of q_1, q_3 were determined by experiments, while q_2 was set at 0.01 for uniformity with the specifications of [33, 65]. Using the data from these papers, we carried out several rigorous initial experiments to select appropriate values for the scaling factor (α) and penalty factors (q_1, q_2 and q_3). Starting with SEA1, Tables 3.15, 3.16, 3.17, 3.18 and 3.19 summarise the results of tuning both α and q_1, all averaged over 30 runs, from where the values: 0.2 and 50,000 respectively were selected; while Table 3.20 summarises the results of tuning the q_3 for q_1 = 50,000 and α = 0.2, all average over 30 runs, from where a choice of 0.4 was made.

α	***0.1***	***0.2***	***0.3***	***0.4***	***0.5***	***0.6***
Av Cost	32,957.44	33,142.36	33,013.07	33,186.51	33,661.90	33,390.89
Std Dev	760.61	601.55	463.21	514.72	1145.81	950.63

Table 3.15: Summary of tuning scaling factor (α), for power balance penalty factor (q_1) = 5, averaged over 30 runs of SEA1 on the 15-unit problem [33, 65]

α	***0.1***	***0.2***	***0.3***	***0.4***	***0.5***	***0.6***
Av Cost	33,293.02	32,625.95	33,039.55	32,627.15	32,640.26	32,723.77
Std Dev	768.36	670.68	650.67	470.13	876.77	609.79

Table 3.16: Summary of tuning scaling factor (α), for power balance penalty factor (q_1) = 50, averaged over 30 runs of SEA1 on the 15-unit problem [33, 65]

α	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**
Av Cost	32,943.18	32,654.30	32,734.35	32,725.23	32,711.46	32,659.72
Std Dev	780.91	486.55	581.99	533.66	532.83	600.90

Table 3.17: Summary of tuning scaling factor (α), for power balance penalty factor (q_1) = 500, averaged over 30 runs of SEA1 on the 15-unit problem [33, 65]

α	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**
Av Cost	32,673.29	32,768.52	32,705.95	32,668.49	32,637.27	33,195.28
Std Dev	589.00	663.34	567.20	598.27	430.72	737.99

Table 3.18: Summary of tuning scaling factor (α), for power balance penalty factor (q_1) = 5000, averaged over 30 runs of SEA1 on the 15-unit problem [33, 65]

α	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**
Av Cost	32,786.10	**32,583.07**	32,607.42	33,198.10	32,623.41	32,609.63
Std Dev	696.77	334.57	396.58	475.78	674.77	617.60

Table 3.19: Summary of tuning scaling factor (α), for power balance penalty factor (q_1) = 50,000. The results are averaged over 30 runs of SEA1 on the 15-unit problem [33, 65]

	q_3				
	0.2	**0.4**	**0.6**	**0.8**	**1.0**
Av Cost	32,569.98	**32,554.51**	32,761.29	32,728.54	32,713.77
Std Dev	575.86	562.64	567.26	421.53	703.60
	q_3				
	1.2	**1.4**	**1.6**	**1.8**	**2.0**
Av Cost	32,731.88	32,861.85	32,854.77	32,812.04	32,853.41
Std Dev	632.39	414.88	585.86	445.42	661.31

Table 3.20: Summary of tuning prohibited operating zones penalty factor (q_3), for power balance penalty factor (q_1) = 50,000 and scaling factor (α) = 0.2, averaged over 30 runs of SEA1 on the 15-unit problem [33, 65]

The tuned values for α, q_1, q_3; tournament size, crossover rate, and smart mutation probability from the experiments of 6 units, the same population size and number of generations from [33, 65] (for equal number of fitness evaluations) are as shown in Table 3.21, which were used for the main experiment involving SEA1, SEA2 and SEA3, with results shown in Table 3.22 and Figure 3.8, averaged over 30 runs.

Parameters	***Values***
Population size	50
Tournament size	2
Crossover rate	0.7
Mutation rate	0.01
Mutation probability (in SEA2)	0.6
No of Generations	100
Elitism Rate	10%
A	0.2
q_1	50,000
q_3	0.4

Table 3.21: Experimental parameters and values for the 15-unit problem

	SEA1	SEA2	SEA3	BEA [1]	PSO [2]	GA [2]	ES [3]
Av Cost	32,552.37	32,669.92	32,549.90	33005.02	35122.79	33228	32620
Std Dev	240.26	279.75	178.59	263.95	1918.62	-	-
Max Cost	33,004.98	33,330.68	32,841.26	33572.28	38044.42	33337	32710
Min Cost	32,146.66	32,213.09	32,208.12	32774.19	33135.30	33113	32568

Table 3.22: Summary of simulation results for SEA1, SEA2 and SEA3, averaged over 30 runs, and comparison with other approaches on the 15-unit problem

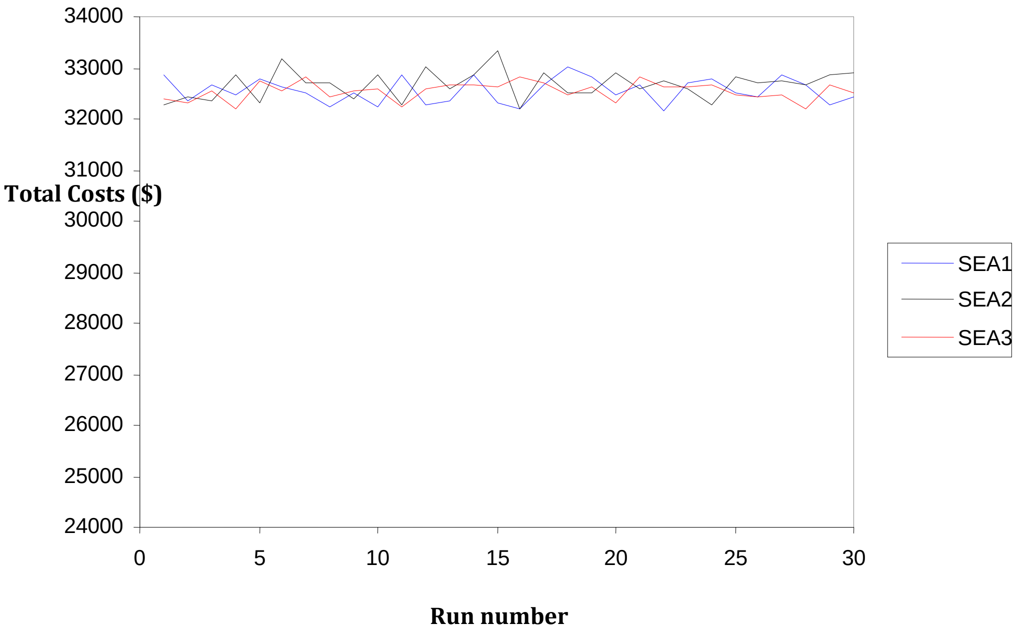

Figure 3.8: Distribution of generation costs for SEA1, SEA2 and SEA3, averaged over 30 runs on the 15-unit problem [33, 65]

Units	SEA1	SEA2	SEA3	BEA [34]	PSO [33]	GA [33]	ES [65]
1	396.96	330.12	295.19	402.08	440.50	415.31	455.00
2	266.38	445.92	310.76	424.99	179.60	359.72	380.00
3	125.31	88.68	107.58	92.11	21.05	104.43	130.00
4	117.85	63.22	85.97	119.10	87.14	74.99	150.00
5	308.33	373.19	446.78	401.60	360.77	380.28	168.92
6	322.02	457.90	238.81	356.65	395.83	426.79	459.34
7	460.06	452.62	403.37	157.84	432.01	341.32	430.00
8	281.33	60.60	293.29	216.38	168.92	124.79	97.42
9	88.18	36.60	59.25	81.58	162.00	133.14	30.61
10	53.71	110.61	118.03	114.77	138.43	89.26	142.56
11	47.77	78.91	71.95	73.29	52.63	60.06	80.00
12	26.81	53.16	60.35	75.85	66.89	50.00	85.00
13	66.30	25.71	76.63	69.13	62.75	38.77	15.00
14	25.99	25.20	42.08	23.93	47.56	41.94	15.00
15	32.78	36.68	20.02	37.22	27.61	22.64	15.00
Total Gen	2630.79	2639.13	2630.05	2646.51	2643.68	2668.40	2653.85
Total Dem	2630.00	2630.00	2630.00	2630.00	2630.00	2630.00	2630.00
Loss	0.79	9.13	0.05	16.51	13.68	38.40	23.85
Total Cost	32,771.63	32,651.91	32,556.90	32,561.06	33,135.30	33,113.00	32,568.54

Table 3.23: Resources allocation in the best of 30 runs of SEA1, SEA2 and SEA3, and comparison with other approaches on the 15-unit problem

Again, the distribution curve of the best solutions in each of the 30 runs, which is shown in Figure 3.4, shows a relatively small range of variation, with the costs equally distributed between the minimum and maximum costs. Although, SEA3 has the lowest average cost of $32,549.90/h, but a solution from SEA1 has the lowest minimum cost of $32,146.66/h. A standard T-test (one tailed) with significance level $p < 0.1$ (confidence level 90%) was applied. The results show no significant difference statistically between the three approaches for this problem. Simulation results of SEA1, SEA2 and SEA3 were compared with BEA [34], PSO [33], GA [33], and ES [65], with the same data. Tables 3.22 and 3.23 summarise the comparison results, including the best resources allocation. Table 3.22 shows that the results of SEA1, SEA2 and SEA3 were better than those of the other approaches. However, we noted that the generation cost from the minimum cost of PSO reported in [33] works out to the value shown in Table 3.23, but is wrongly calculated and/or reported in [33]. Although, there is not much difference in the generation costs of BEA, SEA1, SEA2, SEA3 and ES approaches, the much reduced power loss in SEA1, SEA2 and SEA3 is evident of the smart behaviour of the three approaches. Besides having lower generation costs, they did not over produce beyond the customers demand. The cost of the over generated electricity lost in ES is $292.69/h, against $202.58/h, $9.69/h, $112.03/h and $0.61/h respectively for BEA, SEA1, SEA2 and SEA3. The generation cost and power loss of $32,556.90/h and 0.05MW from the best resources allocation of SEA3 (Table 3.23) are the lowest we have seen in the literature to date for this problem, while meeting load demand.

3.4.3 Test Case III: 20 Generators

In [33, 56], the SELD problem was solved using improved Particle Swarm Optimisation (PSO), Lamda-Iteration Method (LIM) and Hopfield Neural Network (HNN) to determine the optimal combination of the power outputs of 20 generators that will minimise the total fuel cost, satisfying power demand and considering power losses. This problem case investigates the performance of SEA in larger problem case, with the same overall fitness function as in the 6-unit problem case. The total load demand was 2500 MW. Table 3.24 shows the parameters for generating unit capacity and cost coefficients, while the B-matrix loss coefficient is available from [33].

Units	***P** min*	***P** max*	***a***	***b***	***c***
1	150	600	0.00068	18.19	1000
2	50	200	0.00071	19.26	970
3	50	200	0.00650	19.80	600
4	50	200	0.00500	19.10	700
5	50	160	0.00738	18.10	420
6	20	100	0.00612	19.26	360
7	25	125	0.00790	17.14	490
8	50	150	0.00813	18.92	660
9	50	200	0.00522	18.27	765
10	30	150	0.00573	18.92	770
11	100	300	0.00480	16.69	800
12	150	500	0.00310	16.76	970
13	40	160	0.00850	17.36	900
14	20	130	0.00511	18.70	700
15	25	185	0.00398	18.70	450
16	20	80	0.07120	14.26	370
17	30	85	0.00890	19.14	480
18	30	120	0.00713	18.92	680
19	40	120	0.00622	18.47	700
20	30	100	0.00773	19.79	850

Table 3.24: Generators' data for the 20-unit problem

We performed several initial experiments to select appropriate values for scaling factor and load balance penalty factors. Starting with SEA1, and using the previously tuned values of the parameters from the experiments of 3.4.1, Tables 3.25 and 3.26 summarise the results of tuning q_1 and α, from where the values: 50,000 and 0.2 were selected respectively, average over 30 runs. Their respective resources allocations for the best of 30 runs are shown in Tables 3.27 and 3.28. This is in terms of lower cost of generation and meeting load demand.

	q_1						
	5	***10***	***50***	***500***	***5,000***	***50,000***	***500,000***
Ave Cost	61552.11	61705.44	61782.95	61745.74	61600.49	**61108.99**	61710.03
Std Dev	667.18	497.58	771.90	795.51	647.75	547.51	740.68

Table 3.25: Summary of results for tuning power balance penalty factor (q_1), averaged over 30 runs of SEA1, on the 20-unit problem

	A				
	0.1	***0.2***	***0.3***	***0.4***	***0.5***
Ave Cost	61569.85	**61108.99**	61470.32	61487.66	61664.60
Std Dev	807.29	547.51	570.23	672.05	679.05
	A				
	0.6	***0.7***	***0.8***	***0.9***	***1.0***
Ave Cost	61255.18	61371.29	61743.99	61406.48	61900.04
Std Dev	901.48	625.59	501.21	541.59	1097.01

Table 3.26: Summary of results for tuning scaling factor (α), averaged over 30 runs of SEA1 on the 20-unit problem

Units	q_1						
	5	***10***	***50***	***500***	***5,000***	***50,000***	***500,000***
1	541.87	318.62	488.71	390.97	307.89	212.08	386.69
2	191.59	155.34	65.41	88.84	125.65	75.33	101.72
3	100.73	80.44	112.53	69.68	108.83	159.53	70.46
4	151.07	198.61	96.77	73.51	101.43	153.53	78.34
5	107.58	140.25	76.50	80.94	64.88	145.30	126.46
6	39.25	98.89	53.54	21.59	36.07	31.89	59.64
7	120.19	107.13	91.99	104.19	76.19	41.08	101.40
8	68.31	81.04	149.31	146.89	129.49	142.45	61.10
9	146.19	162.19	130.33	122.25	119.11	74.34	127.23
10	82.06	33.78	72.61	130.62	148.31	117.10	52.56
11	131.99	207.13	221.71	207.83	277.63	281.14	226.03
12	197.05	210.50	388.51	431.27	469.72	413.08	451.76
13	83.52	154.80	82.43	152.44	109.93	96.47	125.94
14	125.21	110.62	98.31	43.36	96.66	127.99	80.99
15	116.68	117.95	159.69	142.64	118.09	79.41	172.67
16	49.59	48.05	24.31	35.50	28.30	40.04	45.38
17	46.94	46.62	33.79	52.43	32.63	53.01	43.51
18	61.74	76.90	35.79	87.36	35.63	110.82	44.70
19	53.56	72.57	71.09	60.75	60.41	70.10	95.52
20	86.08	83.21	52.49	70.59	55.11	75.77	54.99
Total Gen	2501.19	2504.64	2505.83	2505.65	2501.96	2500.22	2507.09
Total Dem	2500.00	2500.00	2500.00	2500.00	2500.00	2500.00	2500.00
Total Cost	60801.21	60999.65	60692.22	60711.04	60720.35	**60522.37**	60580.86
Loss	1.19	4.64	5.83	5.65	1.96	**0.22**	7.09

Table 3.27: Resources allocation in the best of 30 runs of SEA1, for different power balance penalty factor (q_1), on the 20-unit problem

The tuned values α, q_1, and those of tournament size, crossover rate, and smart mutation probability from experiments of 3.4.1; including the same population size and number of generations from [33, 56] (for uniformity of equal number of fitness evaluations) are as shown in Table 3.29, which were used for the main experiment, with results shown in Table 3.30 and Figure 3.9, average over 30 runs.

Units	***α***				
	0.1	***0.2***	***0.3***	***0.4***	***0.5***
1	306.73	212.08	364.17	253.32	530.27
2	151.69	75.33	176.68	131.13	152.06
3	116.50	159.53	67.03	148.42	92.85
4	174.69	153.53	167.47	108.47	102.09
5	77.89	145.30	70.26	58.49	50.27
6	73.81	31.89	48.84	57.68	72.90
7	65.13	41.08	31.10	120.67	68.04
8	79.19	142.45	90.30	99.26	137.26
9	129.16	74.34	149.84	150.52	102.40
10	105.37	117.10	37.09	104.92	77.89
11	235.31	281.14	238.77	273.32	150.59
12	439.05	413.08	468.90	296.66	299.14
13	73.28	96.47	119.42	130.16	72.58
14	125.38	127.99	52.87	32.20	121.37
15	61.55	79.41	90.60	122.51	165.91
16	57.53	40.04	72.16	74.00	53.86
17	39.44	53.01	58.67	81.92	63.02
18	68.59	110.82	35.65	99.23	43.81
19	52.38	70.10	96.70	81.77	86.11
20	20.10	75.77	64.24	79.32	60.26
Total Gen	2501.78	2500.22	2500.75	2503.97	2502.65
Total Dem	2500.00	2500.00	2500.00	2500.00	2500.00
Total Cost	60837.01	**60522.37**	60732.54	61054.07	60772.39
Loss	1.78	**0.22**	0.75	3.97	2.65

Units	***α***				
	0.6	***0.7***	***0.8***	***0.9***	***1.0***
1	582.80	438.73	533.40	504.56	388.85
2	96.93	184.64	112.56	171.65	171.84
3	54.50	52.01	124.79	83.91	176.84
4	146.56	79.89	103.17	119.05	176.10
5	142.65	144.39	94.62	130.40	81.16
6	43.12	73.90	52.25	20.85	28.91
7	105.69	88.29	91.98	36.96	93.04
8	50.54	109.39	85.64	63.52	107.64
9	136.80	60.98	53.21	198.31	166.47
10	62.43	145.21	148.29	44.47	92.87
11	158.88	178.21	121.28	109.13	138.71
12	354.77	339.44	410.90	417.83	254.69
13	139.27	153.47	53.70	56.18	97.55
14	83.78	73.97	96.33	82.55	121.59
15	46.42	25.25	116.21	143.47	98.93
16	29.39	28.85	47.11	39.10	45.91
17	38.52	52.90	59.62	65.90	75.61
18	118.97	100.20	66.54	50.19	76.74
19	47.14	89.50	78.88	83.79	52.61
20	65.61	89.74	57.07	78.20	53.99
Total Gen	2504.77	2508.97	2507.54	2500.03	2500.05
Total Dem	2500.00	2500.00	2500.00	2500.00	2500.00
Total Cost	60576.81	60848.93	60850.22	60718.27	61028.95
Loss	4.77	8.97	7.54	0.03	0.05

Table 3.28: Resources allocation in the best of 30 runs of SEA1, for different scaling factor (α), on the 20-unit problem, for power balance penalty factor (q_1) = 50,000

Parameters	*Values*
Population size	30
Tournament size	2
Crossover rate	0.7
Mutation rate	0.01
Mutation probability (in SEA2)	0.6
No of Generations	100
Elitism Rate	10%
A	0.2
q_1	50,000

Table 3.29: Experimental parameters and values for the 20-unit problem

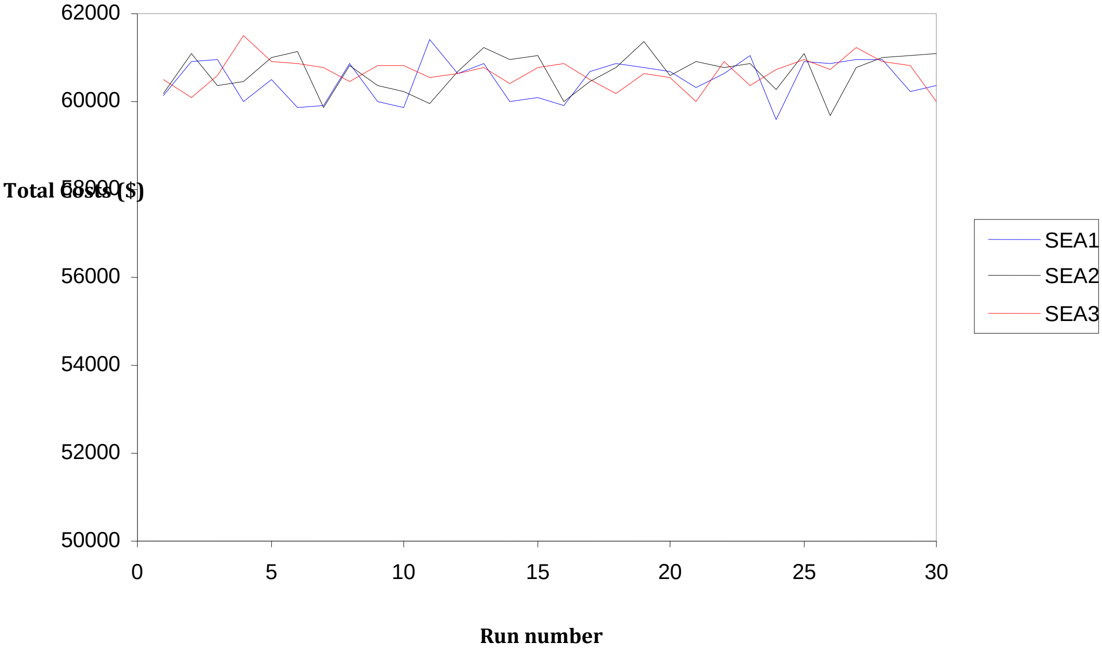

Figure 3.9: Distribution of generation costs for SEA1, SEA2 and SEA3, averaged over 30 runs on the 20-unit problem [33, 56]

Simulation of SEA1, SEA2 and SEA3 were compared with BEA [34], particle swarm optimisation (PSO) [33], Lambda-iteration method (LIM) [56] and Hopefield neural network (HNN) [56], all with the same set of data.

	SEA1	*SEA2*	*SEA3*	*BEA [34]*	*PSO [33]*	*LIM [56]*	*HNN [56]*
Av Cost	60,492.99	60,671.28	60,659.80	61,654.76	61,171.84	-	-
Std Dev	464.65	442.11	331.24	515.57	532.44	-	-
Max Cost	61,422.78	61,359.72	61,498.31	62,648.52	63,184.63	-	-
Min Cost	59,588.38	59,687.80	60,003.02	60,727.19	60,760.25	62,456.64	62,456.63

Table 3.30: Summary of results, averaged over 30 runs for SEA1, SEA2 and SEA3, and comparison with other approaches on the 20-unit problem

Units	*SEA1*	*SEA2*	*SEA3*	*BEA [34]*	*PSO [33]*	*LIM [56]*	*HNN [56]*
1	289.17	482.25	598.68	520.89	536.32	512.7805	512.7804
2	143.69	186.31	90.90	115.14	106.56	169.1033	169.1035
3	139.32	77.08	63.61	122.24	98.71	126.8898	126.8897
4	53.29	88.32	87.35	94.36	117.32	102.8657	102.8656
5	110.31	100.97	113.77	86.18	67.08	113.6836	113.6836
6	93.32	23.15	46.52	60.39	51.47	73.5710	73.5709
7	121.79	97.58	118.37	56.63	47.73	115.2878	115.2876
8	116.38	122.65	120.57	87.13	82.43	116.3994	116.3994
9	89.77	154.67	74.99	49.45	52.09	100.4062	100.4063
10	125.11	111.42	88.70	114.46	106.51	106.0267	106.0267
11	255.72	159.23	215.88	200.39	197.94	150.2394	150.2395
12	315.95	367.03	233.96	410.74	488.33	292.7648	292.7647
13	65.91	99.12	153.89	117.98	99.95	119.1154	119.1155
14	112.47	49.61	84.57	65.63	79.89	30.8340	30.8342
15	80.27	85.92	132.17	125.34	101.53	115.8057	115.8056
16	38.32	52.92	36.65	33.21	25.84	36.2545	36.2545
17	58.66	69.92	43.61	82.43	70.02	66.8590	66.8590
18	94.51	52.46	41.88	60.43	53.95	87.9720	87.9720
19	117.02	46.76	94.70	71.28	65.43	100.8033	100.8033
20	79.06	73.70	59.23	34.01	36.26	54.3050	54.3050
Total Gen	2500.02	2501.06	**2500.00**	2508.31	2512.33	2591.9671	2591.97
Total Dem	2500.00	2500.00	**2500.00**	2500.00	2500.00	2500.00	2500.00
Total Cost	60846.10	60609.72	**60483.14**	60727.19	60760.25	62456.64	62456.63
Loss	0.02	1.06	**0.00**	8.31	12.33	91.97	91.97

Table 3.31: Resources allocation in the best of 30 runs of SEA1, SEA2 and SEA3, and comparison with other approaches on the 20-unit problem

Tables 3.30 and 3.31 summarise the comparison results. The distribution curve of the best solutions in each of the 30 runs as shown in Figure 3.8, alongside Table 3.30 shows that the range of variation of the total costs from each run is also relatively smallest in SEA3. From Table 3.30, SEA1 has both lowest average cost and lowest minimum cost. Generally, the results of SEA1, SEA2 and SEA3 were better than those of the other approaches. Again, the minimum cost of PSO reported in [33] works out to the value shown in Table 3.32, but is wrongly calculated and/or reported in [33]. In Table 4.31, SEA3 has the lowest generation cost of \$60483/h; against \$60609.72/h, \$60727.19/h, \$60760.25/h, \$60846.10/h, \$62456.63/h and \$62456.64/h respectively for SEA2, BEA, PSO, SEA1, LIM and HNN. No power was lost in SEA3, as it exactly generated the customers' load demand of 2500MW; against 0.02MW, 1.06MW, 8.3139MW, 12.33MW, in SEA1, SEA2, BEA, PSO respectively, and 91.97MW in both LIM and HNN. Standard T-test (one tailed) with significant level $p < 0.1$ was applied to see which method is better. We confirmed that SEA1 is superior to SEA3 with 94% confidence, SEA1 is superior to SEA2 with 93% confidence, and SEA1 is superior to BEA with 95% confidence. As noted in Table 3.31, the generation cost of \$60,483/h from the best resources allocation/dispatch of SEA3 is the lowest we have seen in the literature to date for this problem case, while meeting the load demand.

3.7 Summary

The chapter described and evaluated a novel smart EA used to solve SELD problems. We considered practical instances of SELD problems involving generation limits, power balance, ramp-rates and prohibited operating zones constraints and described the mechanisms for handling any of the violated constraints. We developed three versions of the smart mutation operator and performed rigorous genetic tuning of genetic parameters to select values for further experimental runs in the three versions on three benchmark cases involving 6, 15 and 20 generating units; the ones commonly explored in recent literature, with a focus on the larger problem cases. We compared results with those reported for a range of recent alternative algorithms, where they exhibited superior performances. In the 6-unit problem case, simulation results of SEA1, SEA2 and SEA3 were compared with BEA [34], DE [54], GA [54], SLP [94] and QN [36], with the same set of data. On the basis of the best resources allocation to the units, they were all better in terms of both lower generation costs and lower power losses. On the basis of constraints handling capabilities using cost and penalty curves, SEA3 proved the optimisation approach. In the 15-unit problem case, simulation results of SEA1, SEA2 and SEA3 were compared with BEA [34], PSO [33], GA [33], and ES [65], with the same data. From the results, there is not much difference in the generation cost using BEA, SEA1, SEA2, SEA3 and ES approaches, but they did not over produce electricity. In the 20-unit problem case, generally, the results of SEA1, SEA2 and SEA3 were better than those of BEA [34], PSO [33], LIM [56] and HNN [56], all with the same set of data. Both lowest average cost and lowest minimum cost were realised using SEA1, but the best resources allocation of these approaches (Table 3.31) reveals a lowest generation cost being realised using SEA3; and no power was lost. From the above results, SEA1 and SEA3 appear to be the best optimising approaches in this problem.

Chapter 4

A New Evolutionary Algorithm for the Dynamic Economic Load Dispatch Problem

In dynamic environments, optimisation problems change over time. They are also called time-dependent problems or dynamic time-linkage problems, where decisions made at a given time may affect output obtained in a later time [1, 95]. It is expected of algorithms solving dynamic optimisation problems to both locate optimal solutions of the given problem and keep track of such solutions as they change with time. We show in this chapter how the SELD has been extended by others to the dynamic context, and then we extend our approach to solve it. From the experiments that provided superior results on the SELD using three versions of the smart mutation operator, we adapt the algorithm for the dynamic case, and investigate three optimisation approaches. This chapter compares the results of our algorithms on two published dynamic ELD test cases, systems of 5 and 10 generators with identified methods from the literature. These are the major test cases in the literature for which we have comparative results for other algorithms. We also describe, present and evaluate results of 18 variants of the dynamic approach.

4.1 From SELD to DELD Optimisation

DELD optimisation is an extension of SELD optimisation in the context of electrical power generation [4, 5]. The goal is to determine the optimal power generation schedule of on-line generators in a real time basis. In the static problems, the purpose is to optimise the settings for each unit in a generating station so as to supply sufficient power to meet a given overall predicted demand for minimal cost. In the dynamic problems, predicted demand exists for each of a number of successive periods (e.g. hourly periods), and the static problems are solved for each period. The generated power of each unit is determined with respect to predicted load demand over the dispatch period. Traditionally, SELD minimises the total generation cost among the committed units satisfying all constraints. But in practical systems involving ramp-rate limits, operational decisions at a given hour affect the decision at a later hour. Due to the change in load conditions arising from these limits, the power generation has to be altered to meet the demand [2]. This is a major limitation of SELD optimisation. To solve this problem, DELD takes into consideration the dynamic costs involved in changing from one output level to another [5]. Until recently, DELD has been treated as a series of static problems. It is therefore, now, an accurate formulation of the ELD problem on a real time basis, but also a difficult and complex optimisation task.

While optimum static dispatch problems have received considerable attention since 1920 or even earlier, the dynamic dispatch problem was first introduced in 1972 by Bechert and Kwatny [6, 96]. Initially formulated as optimal control dynamic dispatch (OCDD), it modelled power system generation by means of state equations where the state variables are the outputs of the generators and the control inputs are the generators' ramp rates [40, 51]. In a later approach, the optimisation was carried out with respect to the dispatchable powers of the committed generation units, formulated as a minimisation problem of the total cost over the dispatch period under some constraints [51]. This formulation is known as dynamic economic load dispatch (DELD). In this chapter, we extend the SELD formulation to a dynamic context, implement three dynamic approaches using the earlier developed smart mutation operator, and test on two cases.

4.2 Steps in DELD Optimisation

In the most general terms, the following steps are taken to formulate a DELD problem in any given scenario.

1) Select an appropriate objective function representing the problem:
    a. Minimisation of total cost (main objective);
    b. Minimisation of emissions (oxides of carbon, nitrogen and sulphur). This may be considered as a constraint, with the resulting problem referred to as emission constrained economic load dispatch (ECELD), or in conjunction with cost minimisation.
    c. Maximisation of profits (revenue minus cost). This is used in deregulated electricity market, detailed in Chapter 5.

2) Determine the constraints under which the problem will be solved – equality, inequality and dynamic.
    a. Load balance;
    b. Ramp-rate limits;
    c. Generation capacity;
    d. Security constraints;
    e. Emission constraints.
3) Choose a suitable optimisation method which gives an optimal solution within acceptable computational time. The choice of the optimisation method depends on several considerations, such as:
    a. Type of objective function;
        i. Linear/non-linear;
        ii. Smooth/non-smooth;
        iii. Convex/non-convex.
    b. Constraints;
    c. Complexity of problem.

    For DELD with smooth, convex and incrementally increasing cost function, mathematical programming (gradient-based) methods are used. For non-smooth or non-convex cost functions (problems with valve-point effects, ramp-rate limits and prohibited operating zones), artificial intelligent methods are used.

4.3 Previous DELD Solution Approaches

The development of DELD formulations and approaches is a dynamic research area, due to the dynamic nature of power systems and large variations of load demands. Solution is achieved by means of discretisation of the entire dispatch period into a number of small intervals over which the load is assumed to be constant and the system in a temporal steady state [40]. Various methods have been applied to solve the DELD in the past decades with varying degree of success, in terms of yielding an optimal solution, and possibility of getting stuck at local optima. Recent solutions to DELD problems with non-smooth and non-convex cost functions include: General algebraic modelling system [5], Maclaurin Series-Based Lagrangian Method [6], Sequential Approach with Matrix Framework [96], Fuzzy-Optimisation Approach [97], Genetic Algorithm [98, 99], Hopfield Neural Network/Quadratic Programming [100], Swarm Intelligence-Based Harmony Search Algorithm [101], Particle Swarm Optimisation [4, 102, 103, 104, 105, 106], Evolutionary Programming/Sequential Quadratic

Programming [107], Differential Evolution [108, 109], Simulated Annealing [64] Fuzzy Logic/Simulated Annealing [110], Clonal Selection Algorithm [111].

The major common difficulty with all these methods is choosing control parameters. The approach in [6] is simple and easy to implement, with a fast convergence rate, low computational time demands, and produces a unique solution, unlike other stochastic search methods. However, it is mainly applicable to low-dimensional and relatively simple functions. For large power systems, the sequential approach with matrix framework suffers from the curse of dimensionality due to the matrix size and coupled with the sequentiality of the algorithm. In [99], a hybrid approach to the DELD problem was explored, where calculus of variations and a GA jointly optimise the variables, with the GA focussing on penalty weighting parameters. However, a problem involving only 3 generating units was considered and they were not sure of its applicability to larger systems.

Artificially intelligent methods such as GA, ANN, FL, EP, DE, SA, and PSO have attracted great interest in the recent past for realising optimal solutions, and applied to solve DELD problems [96, 106, 107]. However, they are population-based search methods, with random control parameters, and where the number of the search variables are large and highly correlated, realising global optimal solutions becomes a problem due to the large dimensionality of the dynamic dispatch [96, 106].

4.4 DELD Problem Formulation and Constraints

The main objective of a DELD problem formulation [103,105] is to simultaneously minimise the generation cost and meet the consumers' load demand over a given period of time while satisfying three major constraints – load demand balance, generation limit and ramp rates. The SELD objective function of (2.11) is modified to DELD, as shown:

$$Min\, C_T = \sum_{t=1}^{T}\sum_{i=1}^{N} C_{i,i} \tag{4.1}$$

Where, C_T is the total operating cost over the whole dispatch period, T is the number of intervals in time in the scheduled horizon, N is the number of generating units, and $C_{i,t}$ $(Pg_{i,t})$ is the fuel cost of i^{th} generating unit at time, t, for real power output (Pg). We use the cost function with valve-point loadings effect, represented as the sum of the smooth quadratic function and the absolute value of the sinusoidal function given in (2.19), with the power balance, generation limit and ramp-rates limit constraints modified dynamically as shown in (4.2) to (4.10).

Power Balance:

$$\sum_{i=1}^{N} Pg_{i,t} - P_{D,t} - P_{L,t} = 0 \tag{4.2}$$

Where, t = 1, 2, 3… T, $P_{D,t}$ is the total system power demand at time, t, $P_{L,t}$ is the transmission power losses at time interval, t, which is given by:

$$P_{L,t} = \sum_{i=1}^{N}\sum_{j=1}^{N} Pg_{i,t} B_{ij} Pg_{j,t} + \sum_{i=1}^{N} B_{0i} Pg_{i,t} + B_{00} \tag{4.3}$$

Generation Limit:

$$Pg_i^{\min} \leq Pg_{i,t} \leq Pg_i^{\max} \tag{4.4}$$

Where, $Pg_i^{\min}$ and $Pg_i^{\max}$ are the lower and upper limits of the i^{th} generating unit respectively, for i = 1, 2, 3… N and t = 1, 2, 3… T.

Ramp-Rates Limit:

The decisions at the current time period will affect the decisions at a later time period due to variation in power demands over time. There exist three possible cases in actual operation of the units [65, 78]: steady state condition, increasing generation and decreasing generation conditions, as shown in Figure 4.1 (a), (b) and (c) respectively.

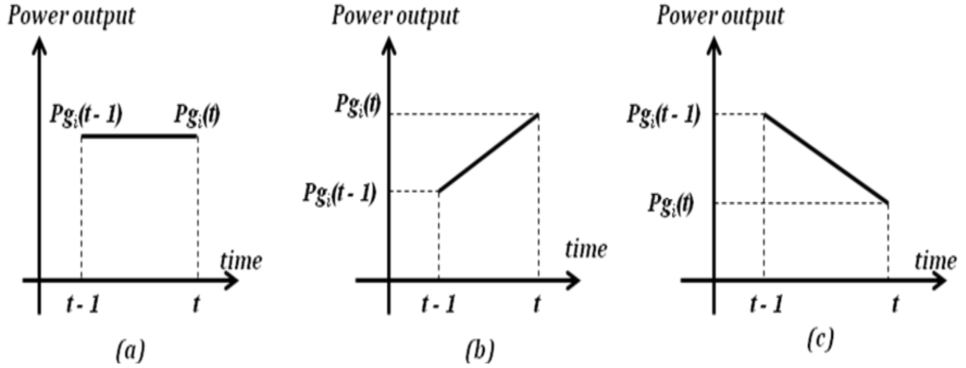

Figure 4.1: Three possible situations of an on-line generator due to ramp-rates

For steady state:

$$Pg_{i,t} = Pg_{i,t-1} \tag{4.5}$$

If Generation increases:

$$Pg_{i,t} - Pg_{i,t-1} \leq UR_i \tag{4.6}$$

If Generation decreases:

$$Pg_{i,t-1} - Pg_{i,t} \leq DR_i \tag{4.7}$$

Where, $Pg_{i,\,t-1}$ is the power generation of unit i at the previous time period, UR_i and DR_i are ramp-up and ramp-down limits of generating unit i, respectively, for: $i = 1, 2, 3 \ldots N$ *and* $t = 1, 2, 3 \ldots T$. Therefore the constraint of (4.4) due to dynamic ramp-rates is modified as:

$$\max(Pg_{i,t}^{\min}, Pg_{i,t-1} - DR_i) \leq P_{i,t} \leq \min(Pg_{i,t}^{\max}, Pg_{i,t-t} + UR_i) \tag{4.8}$$

Where:

$$Pg_{i,t}^{\min} = \max(Pg_{i,t}^{\min}, Pg_{i,t-1} - DR_i) \tag{4.9}$$

$$Pg_{i,t}^{\max} = \min(Pg_{i,t}^{\max}, Pg_{i,t-1} + UR_i) \tag{4.10}$$

4.5 SEA for the DELD Problem

We investigate three dynamic optimisation approaches used in solving DELD problems with valve-point effects and ramp-rate dynamic constraint for a 24-hour dispatch period.

4.5.1 Problem Representation

Assuming there are N generators and T dispatch periods, the control variables can be represented by the following array, with N x T as dimension:

$$Pg^j = \begin{bmatrix} Pg_{11} & Pg_{12} & \dots & Pg_{1T} \\ Pg_{21} & Pg_{22} & \dots & Pg_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ Pg_{N1} & Pg_{N2} & & Pg_{NT} \end{bmatrix} \tag{4.11}$$

Where Pg^j is the j^{th} element in the solution space, Pg_{iT} is the output of generator i at time T.

4.5.2 Dynamic Optimisation Approaches

In this formulation, we took SEAs that have recently provided superior results on the three benchmark problems of SELD, adapted them for the dynamic case, and investigated three dynamic optimisation methods (M1, M2 and M3):

1. M1 is a baseline dynamic optimisation method; it solves the dynamic form of the problem, but by solving the static problems in sequence;
2. M2 involves solving the static problems together, treated as a single multi-part problem with suitably adjusted constraints;
3. M3 extends M2, with the final population of previous periods being used to initialise the populations of subsequent periods.

4.5.3 A Solution Algorithm Procedure for the DELD Problem

Starting with initial feasible solution vectors of (4.12), we describe here the flow of work for the implementation of the dynamic approaches.

$$Pg = [(Pg_{11,} Pg_{21,} Pg_{31,} \dots Pg_{N1}) \cdots (Pg_{1T,} Pg_{2T,} \dots Pg_{NT})] \tag{4.12}$$

i. Read system data.
ii. Initialise randomly at the first period, the population of chromosome within the generating units' range of operation according to the specifications of M1, M2 and M3.
iii. Uniform gene distribution ensures that the values of the generators' outputs are within the legal minimum and maximum limits defined in (3.5). This is modified dynamically as:

$$Pg_{i,t} = \alpha * (Pg_{i,t}^{\max} - Pg_{i,t}^{\min}) + Pg_{i,t}^{\min} \tag{4.13}$$

Where: α is a scaling factor (user-defined small positive number less than one).

iv. With additional dynamic constraints involved (load balance and ramp rate limits) across the dispatch periods, handling constraints violation becomes a harder task, and consequently smart mutation is not as straightforward as in SELD. There exists also a complexity in realising an optimum solution in this formulation due to the presence of valve-point loading effects, which creates additional ruggedness in the cost curve and is likely to increase the density of local optima.

v. In a given dispatch period, the initial population of chromosomes contains feasible genes (outputs for each generating unit) which must be within the minimum and maximum generation limits according to (4.13).

vi. Checks are made to ensure that violations in power balance and ramp-rate limits constraints are handled. Violation of either or both of them constitutes the penalty in this case. The mechanisms of handling these constraints were described in section 3.3.2.

vii. Compute costs of individual genes using the objective function of (4.1).

viii. The genes with the highest cost, including those that violate load balance and/or ramp-rates constraints are subject of the smart mutation.

ix. The penalties augment the objective function to form the generalised fitness function of equation (4.14), used in a similar problem case of [101, 108].

$$C_T = \sum_{t=1}^{T}\sum_{i=1}^{N} C_{i,t} + q_1\left(\sum_{t=1}^{T}\sum_{i=1}^{N} Pg_{i,t} - P_{D,t} - P_{L,t}\right)^2 + q_2\left(\sum_{t=1}^{T}\sum_{i=1}^{N} Pg_{i,t} - P_{rr\lim}\right)^2 \tag{4.14}$$

Where: q_1 and q_2 are penalty terms which reflect the violation of the power balance and ramp-rates constraints respectively, assigning a high cost of penalty to affected ones far from the feasible region.

x. Output the best compromising solution vector for each of the dispatch periods:

$$Pg = [(Pg_{11,}\ Pg_{21,}\ Pg_{31,} \dots Pg_{N1}) \cdots (Pg_{1T,}\ Pg_{2T,} \dots Pg_{NT})] \tag{4.15}$$

4.6 Experimental Design and Simulation Results

The performances of M1, M2 and M3 were tested in two different problem cases involving 5 and 10 generators. The results of parameter tunings for the evolutionary runs and penalty parameters q_1 and q_2 in (4.14), are as described in the experiments of chapter 3. A total of 18 experiments were carried out, making use of SEA1, SEA2 and SEA3 (from chapter 3) for each of M1, M2 and M3 in both 5 and 10 generators.

4.6.1 Case I: 5 Generators

The generators' data and load demand in each hour are shown in Tables 4.1 and 4.2 respectively, which were taken from [6, 102, 108, 109]. The loss coefficients matrix is given in (4.16). The dispatch period is an arbitrary 24 hours and losses were considered. Simulation of the algorithm was made for 30 runs, with the hourly costs for SEA1, SEA2 and SEA3 in each of M1, M2 and M3 given in Table 4.3, while Table 4.4 compares the results with other approaches in the literature using the same set of data.

Unit	***a_i $/h***	***b_i $/MWh***	***c_i $/(MW)^{2}h***	***e_i $/h***	***f_i 1/MW***	***P_i^{min} MW***	***P_i^{max} MW***	***UR***	***DR***
1	25	2.0	0.0080	100	0.042	10	75	30	30
2	60	1.8	0.0030	140	0.040	20	125	30	30
3	100	2.1	0.0012	160	0.038	30	175	40	40
4	120	2.0	0.0010	180	0.037	40	250	50	50
5	40	1.8	0.0015	200	0.035	50	300	50	50

Table 4.1: Generators' data for the 5-unit problem, taken from [6, 102, 108, 109]

$$B = \begin{bmatrix} 0.000049 & 0.000014 & 0.000015 & 0.000015 & 0.000020 \\ 0.000014 & 0.000045 & 0.000016 & 0.000020 & 0.000018 \\ 0.000015 & 0.000016 & 0.000039 & 0.000010 & 0.000012 \\ 0.000015 & 0.000020 & 0.000010 & 0.000040 & 0.000014 \\ 0.000020 & 0.000018 & 0.000012 & 0.000014 & 0.000035 \end{bmatrix} \quad per\ MW \tag{4.16}$$

Period (Hour)	***Load (MW)***	***Period (Hour)***	***Load (MW)***	***Period (Hour)***	***Load (MW)***	***Period (Hour)***	***Load (MW)***
1	410	7	626	13	704	19	654
2	435	8	654	14	690	20	704
3	475	9	690	15	654	21	680
4	530	10	704	16	580	22	605
5	558	11	720	17	558	23	527
6	608	12	740	18	608	24	463

Table 4.2: Hourly load demand for the 5-unit problem, taken from [6, 102, 108, 109]

Hour	M1			M2			M3		
	SEA1	SEA2	SEA3	SEA1	SEA2	SEA3	SEA1	SEA2	SEA3
1	1646.82	1590.39	1480.86	1617.40	1590.78	1611.09	1542.27	1587.03	1508.49
2	1756.98	1594.84	1528.35	1646.99	1667.11	1658.81	1615.11	1685.58	1579.03
3	1673.56	1624.81	1604.98	1665.97	1657.17	1789.49	1650.18	1673.40	1663.93
4	1847.47	1683.50	1662.49	1768.70	1752.72	1848.14	1830.90	1766.13	1726.54
5	1860.56	1789.59	1757.44	1808.95	1936.41	1835.29	1842.68	1915.13	1782.90
6	2067.11	1925.39	1943.08	2001.13	1914.91	1901.05	1909.81	1929.46	1849.05
7	2055.72	1962.25	1933.60	2093.54	1895.70	1930.68	1962.25	1888.83	1838.46
8	2066.91	2010.32	2003.64	2059.04	1906.76	1920.18	2010.32	1904.95	1911.67
9	2150.78	2090.26	2096.37	2114.96	1993.82	1899.48	2090.26	1962.76	1892.80
10	2255.42	2173.67	2121.94	2130.19	2033.66	1932.01	2165.05	2017.54	1950.94
11	2272.31	2171.65	2140.43	2150.46	2197.13	1836.63	2120.43	1997.92	1790.95
12	2297.44	2225.73	2280.74	2084.66	2124.66	1844.66	2112.56	2027.72	1827.45
13	2217.35	2187.57	2130.41	2208.62	2146.13	1867.34	2108.30	2039.44	1850.53
14	2244.22	2112.98	2103.93	2193.17	2187.55	1916.75	2073.03	1989.88	1922.89
15	2131.26	2145.01	1978.04	2192.03	2114.59	1907.77	1972.87	1991.86	1893.28
16	1985.26	1908.59	1924.83	2054.26	1920.37	1911.12	1891.49	1894.66	1867.16
17	1895.88	1823.78	1744.80	1972.13	1915.78	1917.19	1734.05	1935.75	1831.04
18	1983.59	1936.27	1923.54	2029.02	1898.46	1976.09	1917.25	1853.46	1874.35
19	2100.15	2014.84	1979.08	1902.70	1995.21	1910.49	1939.36	1886.44	1896.29
20	2266.59	2181.62	2134.24	1913.84	2030.30	1918.76	1920.51	1985.91	1914.25
21	1953.88	1930.19	1902.16	1872.82	1999.92	1873.74	1905.38	1910.93	1832.89
22	1954.74	1950.85	1975.22	1845.43	1713.15	1919.60	1867.29	1847.69	1901.64
23	1860.02	1827.82	1773.09	1785.91	1709.48	1879.93	1712.58	1802.90	1782.82
24	1701.37	1731.69	1583.11	1873.91	1670.39	1878.00	1582.90	1762.05	1701.60
Total	**48,245.39**	**46,493.59**	**45,706.37**	**46,985.80**	**45,972.22**	**44,884.35**	**45,476.82**	**45,257.42**	**43,590.76**

Table 4.3: Hourly costs of the 9 dynamic approaches, averaged over 30 runs on the 5-unit problem

Approach	Min Cost ($/hr)	Av Cost ($/hr)	Max Cost ($/hr)
PSO [6]	50,124.00	-	-
PSO [4]	49,970.43	50216.59	51803.30
MSL [6]	49,216.81	-	-
SA [64]	47,356.00	-	-
EAPSO [102]	43,784.00	43,794.00	44,041.00
Clonal [111]	43,446.22	-	-
DE [101]	43,213.00	43,813.00	44,247.00
HHS[101]	43,154.86	-	-
M1_SEA1	46,985.74	48,245.39	49,417.14
M1_SEA2	45,414.51	46,493.59	49,711.25
M1_SEA3	44,810.20	45,706.37	46,655.38
M2_SEA1	45,680.25	46,985.80	48,104.08
M2_SEA2	42,125.08	45,972.22	47,379.97
M2_SEA3	**38,638.89**	**44,884.35**	48,437.36
M3_SEA1	44,325.67	45,476.82	46,451.82
M3_SEA2	41,704.25	45,257.42	48,280.88
M3_SEA3	**40,837.70**	**43,590.76**	46,395.08

Table 4.4: Summary of costs of the 9 dynamic approaches, averaged over 30 runs and comparison with other approaches on the 5-unit problem

Table 4.4 reveals that M2_SEA3 and M3_SEA3 seem to be the best two dynamic optimisation approaches for this problem, with the former having the lowest minimum cost and the latter having the lowest average cost over 30 runs among the 9 approaches. To further explore the differences between the two approaches, we make a plot of the variation of their generation costs across the 24 hours dispatch period, as shown in Figure 4.2 (averaged over 30 runs).

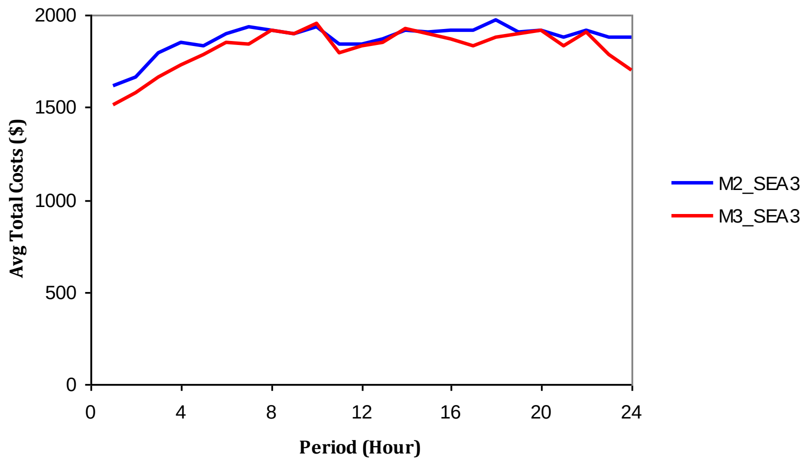

Figure 4.2: Variation of costs of the best two approaches (M2_SEA3 and M3_SEA3), averaged over 30 runs, in the entire dispatch, on the 5-unit problem

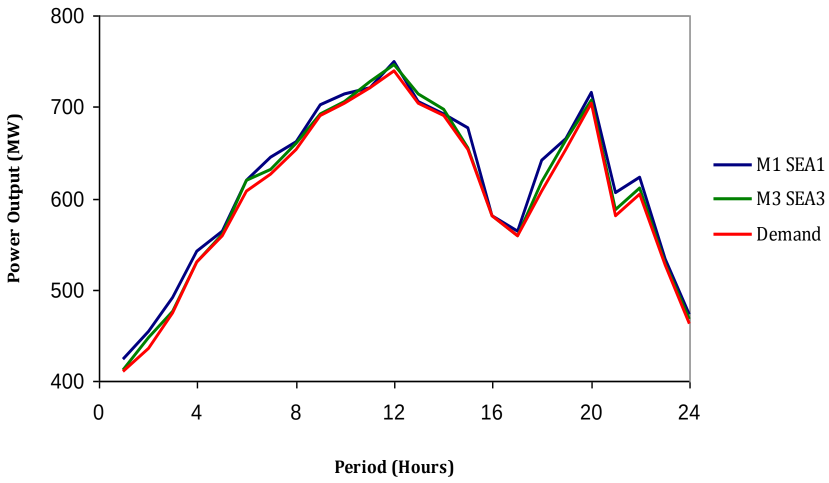

Figure 4.3: Load curve comparing the best approach (M3_SEA3) and worst approach (M1 SEA1), averaged over 30 runs, with load demand on the 5-unit problem

From the curve, it is clear that M3_SEA3 outperforms M2_SEA3. Standard T-test (one-tailed) with confidence level $p < 0.1$ (90% confidence) was applied to see if this difference was statistically significant. With 97% confidence, we confirmed by experimental results that M3_SEA3 is superior to M2_SEA3 ($p < 0.03$). Figure 4.3 is a load curve distribution comparing the best and worst approaches (M3_SEA3 and M1_SEA1) respectively, with the load demand across the entire dispatch. Being a smaller problem case, all the 9 approaches performed relatively well, confirming the performance efficiency of the SEAs. Table 4.5 shows the resources allocation in the best of 30 runs of M3_SEA3.

Hour	*Unit 1*	*Unit 2*	*Unit 3*	*Unit 4*	*Unit 5*	*Total Gen*	*Power Dem*	*Loss*
1	55.95	48.53	102.50	101.75	103.91	412.65	410	2.65
2	11.68	85.48	38.58	187.98	123.53	447.25	435	12.25
3	45.66	88.52	45.97	220.10	75.19	475.44	475	0.44
4	40.29	119.52	121.15	72.43	177.34	530.73	530	0.73
5	58.37	36.13	129.48	141.09	194.85	559.93	558	1.93
6	40.16	116.94	72.47	127.07	263.27	619.92	608	11.92
7	23.99	66.46	154.50	204.16	182.51	631.63	626	5.63
8	46.61	111.47	123.59	151.18	227.65	660.51	654	6.51
9	28.99	61.37	143.64	217.51	240.21	691.71	690	1.71
10	64.02	58.59	115.99	235.61	230.82	705.03	704	1.03
11	12.67	88.14	157.76	198.53	269.80	726.90	720	6.90
12	47.36	67.41	164.07	201.92	264.84	745.60	740	5.60
13	60.73	75.93	142.12	180.72	253.79	713.28	704	9.28
14	53.89	60.59	150.14	156.54	275.85	697.02	690	7.02
15	51.94	107.42	118.20	91.55	286.47	655.58	654	1.58
16	27.48	86.36	62.23	129.23	275.41	580.71	580	0.71
17	69.00	102.91	93.12	65.13	228.24	558.40	558	0.40
18	34.09	85.21	86.36	187.22	224.51	617.38	608	9.38
19	37.20	99.94	93.13	181.54	250.74	662.55	654	8.55
20	70.59	97.61	102.99	207.10	228.39	706.68	704	2.68
21	29.48	94.44	44.96	132.89	285.47	587.24	580	7.24
22	24.29	101.17	142.06	86.52	256.22	610.26	605	5.20
23	63.41	32.84	55.05	134.67	241.62	527.60	527	0.60
24	67.26	72.62	127.17	148.95	51.09	467.08	463	4.08

Table 4.5: Resources allocation in the best of 30 runs of M3_SEA3, on the 5-unit problem

4.6.2 Case II: 10 Generators

The generators' data and load demand in each hour are shown in Tables 4.6 and 4.7 respectively, which were taken from [64, 102, 107, 108, 109]. The dispatch period is 24 hours, with transmission losses ignored for ease of comparison of results with those of other approaches reported in literature.

Unit	a_i $/h	b_i $/MWh	c_i $/(MW)^{2}h	e_i $/h	f_i 1/MW	P_i^{min} MW	P_i^{max} MW	*UR*	*DR*
1	958.20	21.60	0.00043	450	0.041	150	470	80	80
2	131.60	21.05	0.00063	600	0.036	135	460	80	80
3	604.97	20.81	0.00039	320	0.028	73	340	80	80
4	471.60	23.90	0.00070	260	0.052	60	300	50	50
5	480.29	21.62	0.00079	280	0.063	73	243	50	50
6	601.75	17.87	0.00056	310	0.048	57	160	50	50
7	502.70	16.51	0.00211	300	0.086	20	130	30	30
8	639.40	23.23	0.00480	340	0.082	47	120	30	30
9	455.60	19.58	0.01091	270	0.098	20	80	30	30
10	692.40	22.54	0.00951	380	0.094	55	55	30	30

Table 4.6: Generators' data for the 10-unit DELD problem, taken from [64, 102, 107, 108, 109]

Period (Hour)	*Load (MW)*	*Period (Hour)*	*Load (MW)*	*Period (Hour)*	*Load (MW)*	*Period (Hour)*	*Load (MW)*
1	1036	7	1702	13	2072	19	1776
2	1110	8	1776	14	1924	20	2072
3	1258	9	1924	15	1776	21	1924
4	1406	10	2072	16	1554	22	1628
5	1480	11	2146	17	1428	23	1332
6	1628	12	2220	18	1628	24	1184

Table 4.7: Hourly load demand for the 10-unit DELD problem, taken from [64, 102, 107, 108, 109]

Table 4.8 shows the hourly costs (averaged over 30 runs) for SEA1, SEA2 and SEA3 in each of M1, M2 and M3 throughout the entire dispatch period, while Table 4.9 compares the results with other approaches in the literature using the same set of data in terms of total minimum, average and maximum costs.

Hour	***M1***			***M2***			***M3***		
	SEA1	***SEA2***	***SEA3***	***SEA1***	***SEA2***	***SEA3***	***SEA1***	***SEA2***	***SEA3***
1	38313.31	39447.60	41574.58	37978.41	36083.14	32690.97	38772.73	34796.03	31651.14
2	37833.66	41988.94	41389.04	42475.48	37035.72	32482.32	37247.33	35422.78	32846.53
3	39415.11	42466.12	41667.97	42264.87	38227.26	33128.10	43202.17	39002.20	33042.93
4	38348.55	41762.94	41697.76	44254.34	37168.62	35559.17	43224.84	42183.15	34291.91
5	46739.43	43303.06	43288.42	42427.66	35848.01	37250.11	51973.64	41984.53	37207.67
6	45113.18	43604.58	42986.19	45029.90	41895.95	44609.36	42489.33	44285.52	43035.50
7	40002.55	43790.85	43256.59	40195.71	41477.43	43819.79	41115.17	43247.89	44537.43
8	44843.25	42568.29	42022.86	49444.02	40029.04	43484.71	36637.04	39568.58	43058.17
9	46233.03	44030.63	43013.12	47704.65	47371.31	42199.89	37420.92	43374.70	42055.26
10	50229.33	44517.13	43916.68	41945.22	49310.68	51361.63	36158.64	49869.40	51640.30
11	49902.08	44624.73	43810.38	38092.73	50804.44	50962.26	46018.19	49215.63	51154.60
12	49592.99	44498.43	43803.11	36142.29	51006.72	50947.07	47630.11	49105.43	52197.39
13	50929.33	44807.00	43883.21	43492.44	49520.47	51486.81	43677.24	41263.10	51573.64
14	40490.97	42843.23	42302.61	43761.07	47495.42	42280.53	43314.59	38577.28	42221.92
15	39802.55	42545.08	41951.01	47698.34	40057.86	43340.43	44862.99	38854.36	43091.51
16	38999.82	42408.81	41596.97	46354.95	39667.70	41075.00	41665.71	42872.50	39006.07
17	39228.40	42231.53	42431.22	44650.54	36081.35	37216.78	42619.52	42431.22	37474.33
18	46213.18	43887.67	43093.84	42317.32	41962.62	44673.83	42405.33	43093.84	42925.24
19	38820.45	42094.32	41421.31	39986.82	40162.38	43618.04	42575.22	41421.31	43358.17
20	50629.33	44696.57	44116.68	44534.58	49477.35	51461.63	43847.04	44116.68	51806.97
21	45133.03	44115.38	42849.77	44691.47	47537.98	42566.24	43059.67	42849.77	42155.26
22	43413.18	43751.03	42439.76	40200.68	42229.29	44491.83	42940.73	42439.76	43068.84
23	38846.08	42448.25	41661.37	44076.25	37959.39	36116.77	42844.65	43484.71	34097.31
24	39121.64	34859.84	41650.29	45265.22	35908.52	35385.16	40551.13	41075.00	32334.55
Total	**1038149.4**	**1027292.0**	**1021824.8**	**1034984.9**	**1014319.1**	**1012208.4**	**1016281.0**	**1014592.4**	**999832.6**

Table 4.8: Hourly costs of the 9 dynamic approaches, averaged over 30 runs on the 10-unit problem

Approach	***Minimum Cost ($/hr)***	***Average Cost ($/hr)***	***Maximum Cost ($/hr)***
PSO [4]	1,052,655.81	1,055,963.30	1,046,633.42
SQP [101]	1.051,163.00	-	-
EP [101]	1,048,638.00	-	-
EP-SQP [107]	1,031,746.00	1,035,748.00	-
DGPSO [104]	1,028,835.00	1,030,183.00	-
DE [109]	1,023,432.00	1,026,475.00	1,027,634.00
DE [108]	1,019,786.00	-	-
HHS [101]	1,019,019.11	-	-
M1_SEA 1	1,027,664.37	1,038,149.43	1,046,674.90
M1_SEA 2	**803,166.34**	1,027,291.99	1,217,239.92
M1_SEA 3	**734,893.10**	1,021,824.75	1,234,955.97
M2_SEA 1	930,464.83	1,034,984.93	1,142,581.07
M2_SEA 2	983,428.21	1,014,319.09	1,055,054.96
M2_SEA 3	996,122.31	1,012,208.44	1,030,564.23
M3_SEA 1	917,565.27	1,016,280.95	1,090,724.60
M3_SEA 2	**817,541.23**	1,014,592.37	1,088,462.06
M3_SEA 3	977,838.55	**999,832.63**	1,021,407.08

Table 4.9: Summary of results, over 30 runs and comparison with other approaches on the 10-unit problem

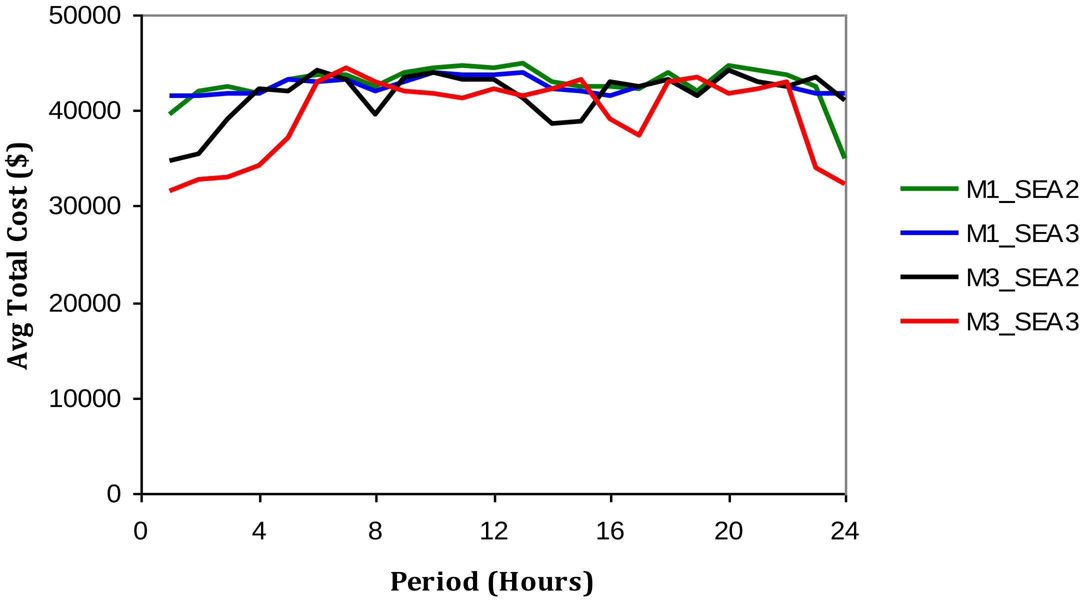

Figure 4.4: Variation of costs of the best 4 dynamic approaches, averaged over 30 runs, in the entire dispatch on the 10-unit problem

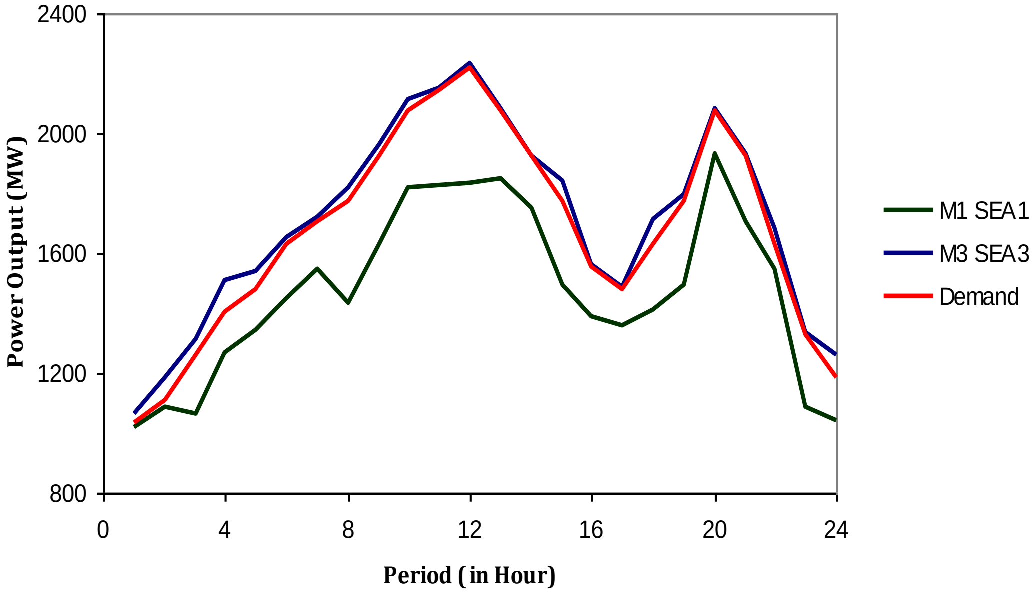

Figure 4.5 Load curve comparing the best (M3_SEA3) and worst (M1_SEA1) approaches, over 30 runs, with load demand on the 10-unit problem

Hour	*Unit 1*	*Unit 2*	*Unit 3*	*Unit 4*	*Unit 5*	*Unit 6*	*Unit 7*	*Unit 8*	*Unit 9*	*Unit 10*	*Total Gen*	*Power Dem*
1	125.2	150.2	160.1	120.3	122.2	117.2	107.2	85.1	20.4	55	1063.0	1036
2	226.3	149.9	164.9	121.8	122.1	120.3	119.1	85.1	20.3	55	1184.9	1110
3	300.4	135.5	184.2	114.6	173.1	129.4	118.1	85.1	20.6	55	1315.9	1258
4	380.0	232.9	188.5	169.7	133.9	118.0	119.1	85.1	25.1	55	1507.5	1406
5	370.8	321.6	85.3	235.0	117.5	117.8	117.1	81.1	40.1	55	1541.3	1480
6	387.4	326.5	195.1	192.1	110.6	117.4	124.2	115.2	26.9	55	1650.5	1628
7	450.8	326.6	200.6	192.4	114.1	118.8	128.3	110.1	26.2	55	1722.9	1702
8	455.8	328.1	307.8	193.1	118.5	115.8	129.1	85.1	27.3	55	1815.6	1776
9	455.2	451.8	307.2	231.2	117.7	113.2	115.2	85.1	30.2	55	1961.9	1924
10	459.8	452.5	337.2	233.2	217.8	119.8	117.2	85.1	39.0	55	2116.6	2072
11	459.6	446.9	338.1	227.9	223.2	133.6	129.7	85.1	49.3	55	2148.4	2146
12	459.1	443.0	339.9	279.8	234.7	159.1	129.1	85.1	49.1	55	2233.9	2220
13	460.8	447.9	333.7	229.7	234.0	124.8	129.0	47.0	20.0	55	2091.9	2072
14	454.8	348.8	279.8	229.4	237.7	124.8	129.5	47.4	20.0	55	1927.1	1924
15	380.4	396.7	224.9	225.0	234.2	127.4	127.2	47.5	20.1	55	1838.4	1776
16	303.9	316.6	317.4	130.5	116.6	120.9	129.9	47.0	20.8	55	1558.5	1554
17	307.0	229.9	343.8	124.9	115.2	120.0	118.8	47.5	26.0	55	1488.1	1480
18	379.2	396.2	286.8	123.1	146.7	129.2	117.5	48.2	29.0	55	1711.0	1628
19	379.4	396.9	285.8	129.0	230.1	129.4	119.2	47.2	26.1	55	1798.1	1776
20	456.0	458.0	330.7	127.0	226.4	123.0	119.1	87.7	97.1	55	2080.0	2072
21	457.1	396.9	315.3	123.1	227.7	123.1	122.0	87.2	28.3	55	1935.7	1924
22	399.0	324.1	306.6	80.2	175.6	119.0	119.4	85.4	21.0	55	1685.4	1628
23	303.1	236.1	198.9	67.1	122.7	125.1	115.1	86.9	28.2	55	1338.1	1332
24	227.3	222.0	197.5	67.1	207.7	123.3	89.1	47.0	21.0	55	1256.8	1184

Table 4.10: Resources allocation in the best of 30 runs of M3_SEA3, on the 10-unit problem

Table 4.9 reveals comparatively lower costs from M1_SEA2, M1_SEA3, M3_SEA2 and M3_SEA3 out of the 9 approaches for this problem in terms of total minimum and average costs. Figure 4.4 shows the variation of their average total costs across the 24 hours dispatch period (over 30 runs), while Figure 4.5 compares the load curve distribution of the best and worst approaches across the entire dispatch period. Again, from the two figures, M3_SEA3 seems to be the best optimiser for this problem, with lower cost (Figure 4.4), and its load trend (Figure 4.5). Results of a standard T-test (one-tailed) with confidence level $p < 0.1$ indicated that M3_SEA3 outperforms M3_SEA2 (with 93% confidence). We also confirmed that M3_SEA3 is superior to M1_SEA2 with 95% confidence, and M3_SEA3 is significantly better than M1_SEA2 with confidence level 97%. Overall, the results tentatively indicate that M3_SEA3 is generally the choice to be recommended. Table 4.10 shows the resources allocation in the best of 30 runs of M3_SEA3.

4.7 Summary

This chapter extended the SELD formulation to a dynamic context. It takes into consideration the dynamic costs involved in changing from one output level to another and is therefore a more applicable formulation of the ELD problem as it is faced by generating stations worldwide. The various steps involved in DELD formulation were shown, and a review of related work on DELD problems was made. The chapter described the DELD problems formulation involving dynamic constraints, and the computational algorithm used. Guided by the smart mutation approaches that provided superior results on in the case of the SELD, we adapted these for the dynamic case, and investigated three optimisation approaches. Finally, we compared the results of two problems test cases, involving systems of 5 and 10 generating units, with identified methods from the recent literature.

In the DELD context, we assessed three approaches: treating the DELD simply as a series of static problems, treating the DELD as a single many-parameter problem, and a basic dynamic optimisation approach, in which the final population of one part of the DELD became the initial population for the next. The two test cases used are the ones commonly explored in recent literature. In conjunction with the three versions of the smart operator we used for the SELD, our results suggest that the third method, which exploits the dynamic nature of the problem, was capable of superior performance to the other two approaches. Comparisons with all approaches so far in the literature that have addressed these problems suggest that this method is superior to previous work. In both test cases, our average and minimum best costs are better than those of the published approaches whenever the comparative figure is obtainable.

Chapter 5

A New Evolutionary Algorithm for the Dynamic Economic Load Dispatch Problem in a Deregulated Electricity Market

Unlike a regulated power system where utility companies share their generating resources to minimise the total cost of supplying the demanded load in a given interval, they compete with each other in a deregulated market to increase their profits [7]. In the deregulated case, the dispatch problem considers generators and financial transactions. The deregulation aims to move electrical power industry from government ownership to private ownership, which results in consumers making their choices of electricity suppliers.

Scenarios in the electrical power industry fall under two broad categories. In the first category, there is sufficient generation capacity to meet demand. Deregulation here creates a healthy competition between the marketers in order to minimise arbitrary price hikes, with improved services. In the second category, the output is not enough to meet the demand. Here, there are a number of sector reforms that precede deregulation, which creates opportunities for more investment funding to provide more infrastructure, including building of more power plants. Generally, deregulation of the power sector helps to increase the efficiency of the generation, transmission and distribution of electrical power, with lower offer prices and higher quality, increased reliability and security.

In most researches involving deregulated environments, DELD transactions were carried out in static context. Here, transactions in the successive dispatch periods/intervals were presented as exclusive events, that is, they were functions of the state of the system at that particular period. But in reality, as evident from the findings of the previous chapter, transactions in one dispatch period will be subject to various decisions in the preceding periods due to incorporation of dynamic constraints such as ramp rates, minimum up/down times, minimum/maximum generating capacity, fuel constraints, etc. The effects of these constraints were analysed in [7], where transition states were calculated using successive dynamic programming, applying Newton's method to compute the optimal states for a given set of transactions within a utility. Deregulation shifts the goal of ELD in the power market from the traditional cost minimisation to maximisation of social profit, an approach termed "profit-based dynamic economic load dispatch" (PBDELD) [9, 10, 11, 12, 13].

This chapter investigates the use of smart EAs to maximise social profit in a deregulated electricity environment. It reviews related work in the literature and describes the solution optimisation procedure, involving three bidding strategies – low, medium and high bidding strategies. It compares results of the best two versions of the smart mutation operator with previous results in the literature involving two test cases, and also defines and show results using a new, larger test case.

5.1 Deregulation in the Global Electricity Market

The global electricity market has for many decades experienced reforms, leading to liberalisation, privatisation and full deregulation. The rate of these reforms however, varies from country to country, with USA and Chile regarded as pioneers of deregulation [112]. In 1978, the Public Utility Regulatory Policies Act of USA and Wholesale Market Pool of Chile were set up. The former required utility firms to buy electricity from "qualifying facilities" of co-generators and smaller power plants, while the latter ensured that generating companies sell their power to retailers [113]. Table 5.1 shows a summary of introduction of reforms and restructuring, leading to liberalisation and privatisation, which subsequently culminated to electricity market deregulation in selected countries.

S/N	*Countries*	*Established Legislation*	*Year*
1	USA	Public Utility Regulatory Policies Act	1978
2	Chile	Wholesale Market Pool	1978
3	United Kingdom	Electricity Act	1989
4	Norway	Energy Act	1990
5	New Zealand	Energy Act and Company Act	1992
6	Australia	Electricity Industry Act	1994
7	Spain	Electricity Act	1994
8	Finland	Electricity Market Act	1995
9	Japan	Electric Utility Law	1995
10	Poland	Energy Act	1997
11	Canada	Energy Competition Act	1998
12	Netherlands	The Electricity Act	1998
13	Ireland	Electricity Regulation Act	1999
14	Czech	The Energy Act	2000
15	Hungary	Electric Power Act	2001
16	Iceland	Electricity Act	2003
17	India	Electricity Act	2003
18	Nigeria	Electric Power Sector Reform Act	2005

Table 5.1: Introduction of deregulation in selected countries

The model of deregulation in Britain, USA and Norway has particularly attracted much discussion attention and growing interest [112, 113]. An ideal deregulated power market controller consists of two entities: the power exchange, and the independent system operator. The power exchange is responsible for the spot trade in the day-ahead market whose main task is to deal with the DELD problem, while the independent system operator (ISO) regulates the activities of the participating companies and stakeholders, including management of traffic, reserve, network security, etc. However, in most cases, these two entities are hybridised into the ISO only (as used in this work), which performs both of these functions.

5.2 Optimal Transactions Dispatch in a Deregulated Market

To achieve a deregulated electrical power sector, there must be the presence of generation companies (GENCOs), transmission companies (TRANSCOs) and distribution companies (DISCOs). Before restructuring, utility companies have the obligation of meeting the predicted load demand of their customers. But with deregulation, GENCOs might decide to produce less than the predicted load in order to maximise their profits [8]. In this scenario, each participating utility submits its bid in form of an offered price for each transaction to the market controller, and receives a scheduled transaction for that particular dispatch period. It is the responsibility of each utility to determine the time, price and the power output that must be transacted within the network in order to maximise its profit. The market controller defines the prevailing market prices for maximising the overall profit. This function is carried out by the independent system operator (ISO). Figure 5.1 shows a system of three utilities and available generators for transactions.

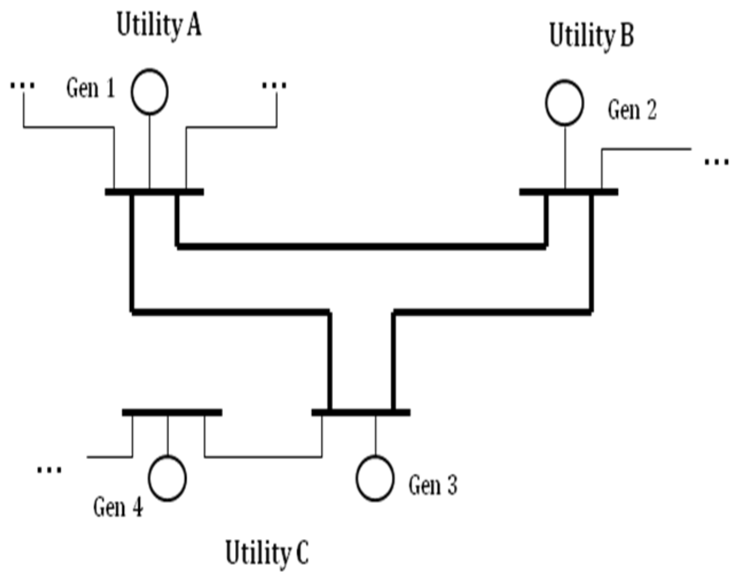

Figure 5.1: A power system showing 3 utilities and 4 generators

Depending on the resultant market structure, there are two cases involved in the formulation of the dispatch problems [51]. The first case is the sole responsibility of individual GENCOs. Here, the ELD schedule of the committed generating units is based on a compromise between the forecasted power/load demand and prices. Therefore, in solving the problem, GENCOs aim not only to minimise total generation cost, but to maximise their profits, with the power price forming a major decision-taking yardstick. This is referred to as Price-Based Dynamic Economic Load Dispatch (PBDELD), to emphasise the importance of energy price in the dispatch transaction [8, 10, 114, 115, 116, 117, 118]. In the second case, the increased competition also increases the number of trading/bidding stakeholders. Both the GENCOs and their customers submit their bids in advance of each bidding period on a daily or hourly basis to ISO, who matches the bids and conducts the DELD, with an aim to maximise the social profit during the trading time-period, while maintaining system reliability and security. This formulation is referred to as Bid-Based Dynamic Economic Load Dispatch (BBDELD) [119, 120, 121, 122, 123]. It moves the dispatch operations from traditional cost-based to a bid-based mechanism.

5.2.1 Price-Based DELD

In this formulation, each GENCO runs its own dispatch based on the forecasted power demands. It might decide to sell its power at less than the forecasted level to maximise profits. Figure 5.2 depicts a typical offered and required price curve in a scheduling transaction *T*, at each dispatch period for each generating unit.

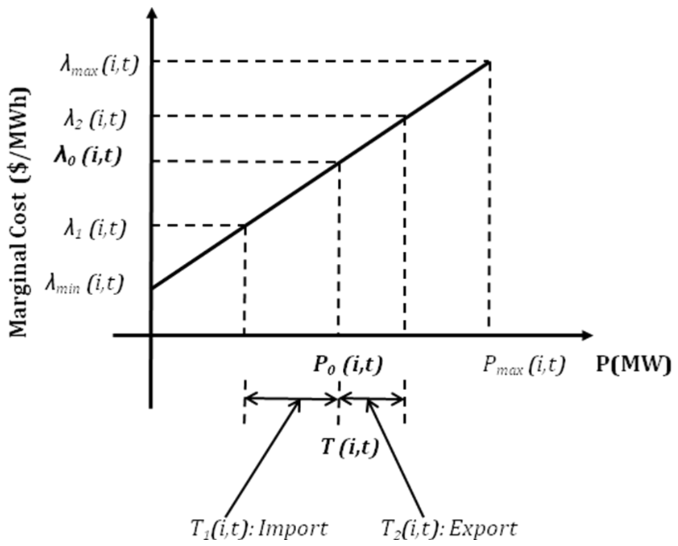

Figure 5.2: Offered and required price curve [7]

The local generation is P_0, and the corresponding incremental cost is λ_0. If the prevailing market price forecasted by ISO is as high as λ_{max}, the transacting utility might decide to increase generation to P_{max}. On the other hand, if the price is below λ_0, the utility will consider importing power. As λ_1 tends to λ_{min}, the utility might replace all the local power generation with imported power, while the entire generated power is available for export as λ_2 tends to λ_{max}. Given a set of spot prices (market prices at each dispatch period), the objective in each utility is defined as follows:

$$Minimise \sum_{t=1}^{T} \sum_{i=1}^{Ng} \left[C_{i,t}(Pg_{i,t}) - \rho_t T_{i,t} \right] \tag{5.1}$$

Where: ρ_t is the spot price at hour, t, defined and/or computed as:

$$\rho_t = \frac{\partial C_{i,t}(Pg_{i,t})}{\partial Pg_{i,t}} + \sum_{\ell \in \Psi} \mu_\ell F_{\ell,t} \tag{5.2}$$

F_ℓ is the power flow in line ℓ, μ_ℓ is a constant (Lagrange multiplier) for each line, while ψ is the set of lines in the utility. Obtaining optimality in the online generating units is the overall objective of a PBDELD problem [7], mathematically formulated as:

$$Maximise: PF = RV - TC \tag{5.3}$$

Where *PF* is profit; *TC*, *RV* are total cost and revenue, defined respectively as shown in (5.4) and (5.5).

$$TC \sum_{t=1}^{T} \sum_{i=1}^{Ng} [C_{i,t}(Pg_{i,t}) + ST_t)] \tag{5.4}$$

$$RV \sum_{t=1}^{T} \sum_{i=1}^{Ng} \sigma_t (Pg_{i,t}) \tag{5.5}$$

The power demand constraint is given in (5.6), while the generation limit constraint is as shown in (4.4).

$$\sum_{t=1}^{T} \sum_{i=1}^{Ng} Pg_{i,t} \le D_t \tag{5.6}$$

Where:

C_{it}	=	*Production cost, as calculated from (6.1);*
$Pg_{i,t}$	=	*Power generated in i^{th} unit at t hour;*
ST_t	=	*Start-up cost;*
T	=	*Dispatch period;*
t	=	*Time index;*
i	=	*Generator index;*
Ng	=	*Number of online/committed generators;*
σ_t	=	*Forecasted market price of power at time, t;*
D_t	=	*Total system demand at time, t;*

In (5.3), (5.4) and (5.5), given the values of start up costs and/or spot prices at the various dispatch periods, revenue computation is a straight forward task and consequently, profit is easily deduced. This reduces the complexity of the PBDELD, with more research attention given to BBDELD.

5.2.2 Bid-Based DELD

In this formulation, both the suppliers and customers submit their bids (hourly or daily ahead) to ISO who obtains transmission line capability and availability from TRANSCO and uses the submitted bids to determine the corresponding supply and demand schedules of all generators and customers, and determine the market clearing price.

5.3 Previous BBDELD Solution Approaches

Several approaches have been proposed to solving the PBDELD problem including: Predicted-Corrected Interior Point Quadratic Programming Algorithm (PCIPQP) [121], Linear Programming (LP) [124], Dynamic Programming (DP) [7, 125, 126, 127], Simulated Annealing (SA) [128, 129, 130], Evolutionary Strategy (ES) [131], Genetic Algorithm (GA) [119, 120, 123, 132, 133, 134, 135], Particle Swarm Optimisation (PSO) [122, 136], and Differential Evolution (DE) [137].

The PCIPQP algorithm creates room for both the demand and supply sides participation in the spot market bidding mechanism, involving a multi-player, multi-period and various dynamic constraints. The initial quadratic programming problem formed is solved using interior point-based algorithm, hence the PCIPQP is said to be an extension of interior point quadratic programming (IPQP). Development of the IP-based QP problem involved the use of Fiancco and McCormick's barrier, Newton's and Lagrange's optimisation methods [121].

In [124], a Linear Programming (LP) approach was proposed for bidding of generators in restructured power industry by simulating risk profiles of generators for different bidding strategies. This approach minimises only linear objective functions with linear constraint variables. While solutions obtained have better computation efficiency and avoid problems of premature convergence, the optimisation limitation implies that any non-linear function will have to be approximated by linear functions.

In [125], a Successive Approximations Dynamic Programming (SADP) algorithm was developed for dispatching generation to predicted load over a given time-horizon. The central idea in the successive approximations method involves breaking large power system problems containing many control (power generation) variables into a number of smaller sub-problems, each containing only one control variable and only one state variable. This reduces the computational difficulty as the computational requirements of dynamic programming increase exponentially with the number of state variables. The iterations (successive approximations) are based upon pairing of generation units, with the resulting one-dimensional dynamic problem being solved by forward dynamic programming. At each time period, Bellman's Principle of Optimality [138], is applied to the resulting simple solution procedure to determine the optimum control (generation change) that will bring the unit to an acceptable output level.

In [126], a series of computer programs using Dynamic Programming (DP) was developed to calculate pseudo incremental heat rate curves that are monotonically increasing, including the actual economic dispatch points. Modelling unit input/output characteristics using third or higher order polynomial curves increases the accuracy of output, but results in some incremental heat rate curves that are not monotonically increasing. A great problem arises when many units with such characteristics are dispatched. Overcoming the problem involves using DP to calculate each unit's output over the range of the sum of minimums to the sum of the maximums on all units to be dispatched. A curve fitting routine is used to smooth the results of each unit's output over the range of system load. Another DP is used to divide the curve fit into connecting straight line segments that are monotonically increasing. The DP problem is characterised by stages, states, decision variables at every stage and a good (recursive) relationship. Each generating unit has an associated stage, with the number of stages in the economic dispatch problem being equal to the number of committed units. The states are total amount of generation at a particular stage, with values ranging from the sum of the minimum outputs of all units in that stage to the sum of the maximums of the units in that stage. The decision made at a particular stage is the amount of generation allocated to the generating unit associated with that stage, which are assumed to be discrete. A zoom feature was added during the iterative process of a DP in order to converge to the economic dispatch solution, reducing computer time and memory requirements [127]. The zoom feature reduces both the search step size and search range during successive iterations. While there is a rapid convergence with the approach, there are no restrictions on the shape of the generator operating costs functions. It could also be combined with short-term load forecast to minimise total operating costs over the present and short-term future.

Simulated Annealing (SA) is a much faster algorithm than DP [128]. SA is similar to local search techniques with the capability to guarantee only a local optimum solution, but its probabilistic approach in accepting candidate solution enables it to jump out of the local optimum solutions [129]. SA is advantageous in that, it does not need a complex mathematical model of the optimisation problem; the solution process is independent of the fuel cost characteristic function of the generators; the feasible solution is independent of the choice of initial solution, it could improve a solution output from other sub-optimal or heuristic methods; it has a faster convergence, and low memory requirements [129, 130]. However, the major disadvantage is the high computing time requirement. This could be improved by adopting parallel processing, by modifying the algorithm into a form suitable for execution in a multi-processor system.

In recent researches, solutions to BBDELD were realised using Particle Swarm Optimisation (PSO) [122], Genetic Algorithm (GA) [119, 123] and Differential Evolution (DE) [137]. All these approaches proved to be efficient in solving the problems, but they seem to lack the ability of finding global optimal solutions due to their inherent drawbacks. Besides, the problem cases were very limited (3 generators, 2 customers in 2 trading periods in [121, 122] and 4 generators, 3 customers in 3 trading periods in [119, 123]). In [137], the effects of three bidding strategies – low, medium and high on the amount of social profit realised were investigated, and tested with 6 generators, 2 customers in 2 trading periods. While the results confirm the high bidding strategy as best suitable for optimal dispatch in deregulated power systems (receiving higher customer benefits which lead to maximum social profit), again, the problem size is very limited. No literature has yet explored the performances of these approaches in larger test cases. It is required that the current solution algorithms work well for a wider variety of cases, and expanded to deal with larger scale energy optimisation tasks. This is because there is continual expansion of the electricity market as a result of the healthy competition which means addition of more generating units.

The social profit depends on the effectiveness of the bidding strategy, first introduced by A. K. David in [139]. Since the submitted bids determine GENCOs' revenues, attempts have been made to formulate strategies to get the best bids. Therefore, developing appropriate and optimal bidding strategies has been an issue of great concern among researchers in recent years [116]. Generally, bidding problems have been addressed subject to market structure, available generating units and bid functions [140]. According to the authors, bidding strategies in competitive electricity market include: auction, single-part bidding, multi-part bidding, demand-side bidding and supply-side bidding.

5.4 The BBDELD Problem Formulation

A bid is made up of the offer price from GENCOs and the amount of load demanded by the customers at a given despatch time period [142]. Two trading mechanisms exist in electricity markets: centralised auction and bilateral trading [137]. In centralised auction, both the suppliers and customers submit their bids to ISO who performs the dispatch depending on the price and quantity of power involved in the bidding. In bilateral trading, both the sellers and buyers submit their bids, trade the quantities involved at their own discretion without involving ISO. It is only notified of the transaction with a request to make transmission facilities available for a given quantity of power. The ISO then despatches the request and transaction charged accordingly, with payment made by both the sellers and buyers. The transaction model presented in this work is based on the centralised auction trade mechanism.

A consideration is made of both the supply and demand side bids to stabilise the power price by adjusting the balance of demand and supply during a multi-player transaction in a multi-period dispatch [122]. The social profit is computed as the difference between the total customers benefit and the total generators cost in a deregulated (competitive) electricity market. The bid price curve for each of the market participants is modelled by a quadratic function of the real power output. The BBDELD is formulated as:

$$Max\ PF = \sum_{t=1}^{T} \left| \sum_{j=1}^{Nd} B_j(D_{j,t}) - \sum_{i=1}^{Ng} C_i(Pg_{i,t}) \right| \tag{5.7}$$

Subject to:

$$B_j(D_{j,t}) = a_{dj} D_{i,t}^2 + b_{dj} D_{j,t} + c_{dj} \tag{5.8}$$

$$C_i(Pg_{i,t}) = a_{gi}Pg_{i,t}^2 + b_{gi}Pg_{i,t} + c_{gi} \tag{5.9}$$

Where:

$B_j(D_{j,t})$	=	*Benefit (or bid) function of customer, j;*
$C_i(Pg_{i,t})$	=	*Cost (or bid) function of generator, i;*
N_d	=	*Number of customers;*
N_g	=	*Number of generators;*
$D_{j,t}$	=	*Bid quantities of customer, j at t;*
$Pg_{i,t}$	=	*Bid quantities of generator, i at t;*
a_{dj}, b_{dj}, c_{dj}	=	*Benefit (or bid) coefficients of* j^{th} *customer;*
a_{gi}, b_{gi}, c_{gi}	=	*Cost coefficients of* i^{th} *generator;*
i	=	*Generator index, i.e., 1, 2, 3,…, Ng;*
j	=	*Customer index, i.e., 1, 2, 3,…, Nd;*
t	=	*1, 2, 3,…T;*
T	=	*Number of time period.*

The power demand, generation limit and bid quantities constraints are respectively shown in (5.10), (4.4) and (5.12).

$$\sum_{i=1}^{Ng} Pg_{i,t} = \sum_{j=1}^{Nd} D_{j,t} + Loss_t \tag{5.10}$$

$$D_{j,t\,\min} \le D_{j,t} \le D_{i,t\,\max} \tag{5.11}$$

Where:

$D_{j,t\,min}$ = *Minimum bid limit of customer j at time t.*

$D_{j,t\,max}$ = *Maximum bid limit of customer j at time t.*

5.5 The BBDELD Optimisation Procedure

5.5.1 Bidding Strategies

GENCOs adopt the production cost bidding strategy. Here, they bid according to their plants' incremental cost, acting as a pure price taker in the market [141]. The generators' bid price curves are approximated as quadratic functions, as shown in (5.9). The bid function of the customers is also approximated in quadratic form, shown in (5.8). Three bidding strategies exist for participating customers, classified as: *low*, *medium* and *high* bidding, based on the bid price coefficients. Previous experiments in literature show that the value of customer bid coefficient, a_{dj}, is less than 0.01 for low, in the region of 0.05 for medium, and above 0.09 for high bidding strategies. The recommended range of customer bid coefficient, b_{dj}, is between 0 and the energy clearing (or equilibrium) price [137, 142].

5.5.2 SEA for the BBDELD Problem

For a system of N generators and T dispatch/trading periods, the control variables are represented by an array of dimension N x T. The following pseudo code describes the procedure used to solve the BBDELD problem:

START
DEFINE parameters and INPUT data
INITIALIZE a random population of chromosomes
EVALUATE Objective function (For each chromosome in the population)
FOR t = 1 to T
    FOR i = 1 to Nd and Ng

$$PF = \sum_{t=1}^{T} \left| \sum_{j=1}^{Nd} B_j(D_{j,t}) - \sum_{i=1}^{Ng} C_i(Pg_{i,t}) \right|$$

Subject to:

$$B_j(D_{j,t}) = a_{dj} D_{i,t}^2 + b_{dj} D_{j,t} + c_{dj}$$

$$C_i(Pg_{i,t}) = a_{gi} Pg_{i,t}^2 + b_{gi} Pg_{i,t} + c_{gi}$$

Compute power losses
Check for constraints violation in the power balance
Check for constraints violation in the generator bid limits
Check for constraints in ramp rate limits
Check for constraints violation in the customer bid limits
Violation of a constraint leads to penalty treatment, defined as:

$$Max\ F = \begin{cases} PF \\ PF + penalty \end{cases}$$

i.e.:
IF no constraint is violated,
        Penalty = 0
        Fitness function = Objective function
ELSE
        Fitness function = Objective function + Penalty
END IF
    END FOR
END FOR

WHILE (stopping criteria is not reached)
From the entire population
SELECT parents for breeding
CROSSOVER the genes of selected parents
Create an array to return children
Crossover parents to create children
Return the children in the array
*Perform **SMART MUTATION***
EVALUATE
REPLACE parent population with children population
Perform ELITISM (Keep a percentage of best individuals)
END WHILE
OUTPUT the optimal solution vector, and compute the cost, benefit and profit as:
Total cost = Sum of generation costs of in all the periods
Total benefit = Sum of customer benefits in all the periods
Social Profit = Total benefits – Total costs
STOP

The DELD solution algorithm in section 4.5.3 was utilised here, in realising the total costs in all dispatch periods. Similar to the DELD, with additional dynamic constraints involved (load balance, ramp rate limits and generators' output limits), handling penalty violation becomes a harder task, and consequently smart mutation is not as straightforward as it is in the SELD. In a given time period, an initial population of chromosomes contains feasible genes (outputs in each generating unit) which must be within the minimum and maximum generation limits. Checks are made to ensure that violations in power balance, ramp rate limits, generator bid limits and customer bid limits are handled. Violation of power balance and ramp rate limits leads to penalty treatment. Costs of individual genes are computed using (5.9), and the genes with the highest cost alongside with those that violate the above penalties are subjects of the smart mutation, defined in (5.12).

$$Pg_{i,t(new)} = Pg_{i,t(old)} - (random() \times Pg_{i,t(old)}) \tag{5.12}$$

The penalties augment objective function to form a generalised fitness function for social profit (SP) shown in equation (5.13), with the aim of maximising it.

$$Max\ SP = \sum_{t=1}^{T} \left| \sum_{j=1}^{Nd} B_j(D_{j,t}) - \sum_{i=1}^{Ng} C_i(Pg_{i,t}) \right| + q_1 \left(\sum_{t=1}^{T} \sum_{i=1}^{N} Pg_{i,t} - P_{D,t} - P_{L,t} \right)^2 + q_2 \left(\sum_{t=1}^{T} \sum_{i=1}^{N} Pg_{i,t} - P_{rr\,\mathrm{lim}} \right)^2 \tag{5.13}$$

Where: q_1 and q_2 are penalty parameters which reflect the violation of the power balance and ramp-rate limit constraints in each unit and at each dispatch period. Their values are tuned experimentally; and *random* () generates a small random positive number between 0 and 1. As the bid function of the customers is based on high bidding strategy, it is unlikely that the bid limits constraint is violated.

5.6 Experimental Design and Results

The BBDELD optimisation was performed using the smart mutator in three different experiments involving three problem cases:

(a) A system of 3 generators, 2 customers in 2 dispatch periods;
(b) A system of 6 generators, 2 customers in 2 dispatch periods;
(c) A system of 10 generators, 6 customers in 12 dispatch periods.

All the experiments were based on dynamic optimisation method M3 where the problem is solved as a single multi-part problem with suitably adjusted constraints, and the final population of preceding periods used to initialise the populations of succeeding periods. For the purpose of comparison with other approaches in literature with the same set of data, the genetic parameters were set as: crossover probability: 0.9, population size: 20, and maximum generation: 200.

5.6.1 Case I: 3 Generators, 2 Customers in 2 Dispatch Periods

This problem case involves a 5-bus test system, consisting of 3 generators and 2 customers, 6 lines, in 2 dispatch periods, shown in Figure 5.3 [121, 122]. This case aims at illustrating the operational transaction of BBDELD.

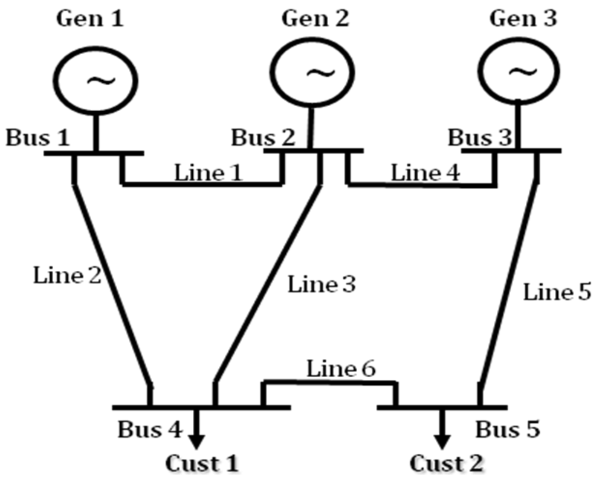

Figure 5.3: A 5-bus problem test (case I), consisting of 3 generators, 2 customers and 6 lines

Tables 5.2 and 5.3 show the bid data for generators and customers. For simplicity and ease of comparison with other approaches, ramp rates constraints and transmission losses were not considered. Table 5.4 gives a summary result showing the total customer benefits, total generation costs and social profit, average over 30 runs for the three dynamic optimisation approaches of M3 (M3_SEA1, M3_SEA2 and M3_SEA3); while the bid quantities for the generators and customers are optimally allocated among the two periods as shown in Table 5.5 for M3_SEA3.

	a_{gi}	b_{gi}	c_{gi}	Pg_{min}	Pg_{max}
Generator 1	0.001562	7.92	560	0	600
Generator 2	0.001940	7.85	310	0	400
Generator 3	0.004820	7.97	78	0	200

Table 5.2: Generators' bid data for case I (3 generators and 2 customers, in 2 dispatch periods)

				D_{min}		D_{max}	
	a_{dj}	b_{dj}	c_{dj}	***Period 1***	***Period 2***	***Period 1***	***Period 2***
Customer 1	-0.175	100	0	400	200	650	300
Customer 2	-0.150	110	0	200	300	350	400

Table 5.3: Customers' bid data for case I (3 generators and 2 customers, in 2 dispatch periods)

	Total Benefits ($)	***Total Costs ($)***	***Social Profits ($)***
M3_SEA1	64,705.67	13,060.79	51,644.88
M3_SEA2	66,824.78	13,467.59	53,357.19
M3_SEA3	67,462.09	13,217.62	54,244.47
PCIPQP [121]	66,246.00	13,158.00	53,088.00
GA [122]	67,005.00	13,294.00	53,711.00
PSO [122]	66,315.00	13,198.00	53,117.00
LP [122]	64,372.00	13,426.00	50,936.00

Table 5.4: Summary of results for case 1 (3 generators and 2 customers, in 2 dispatch periods), averaged over 30 runs

Parameters	Period 1	Period 2	Total
Customer 1	380.39	292.64	
Customer 2	374.85	329.77	
Total Demand (MW)	755.24	622.41	
Total Benefit ($)	37316.23	31204.71	68,520.93
Generator 1	411.80	432.76	
Generator 2	166.28	4.77	
Generator 3	186.17	185.99	
Total Gen (MW)	764.25	623.52	
Total Cost ($)	7484.04	6354.58	13,838.62
Losses (MW)	9.01	1.11	
Social Profit ($)			**54,682.31**

Table 5.5: Resources allocation in the best of 30 runs of M3_SEA3, for case I (3 generators and 2 customers, in 2 dispatch periods)

5.6.2 Case II: 6 Generators, 2 Customers in 2 Dispatch Periods

This problem case involves an 8-bus test system, consisting of 6 generators, 2 customers and 12 lines, in 2 dispatch periods shown is Figure 5.4 [136, 137].

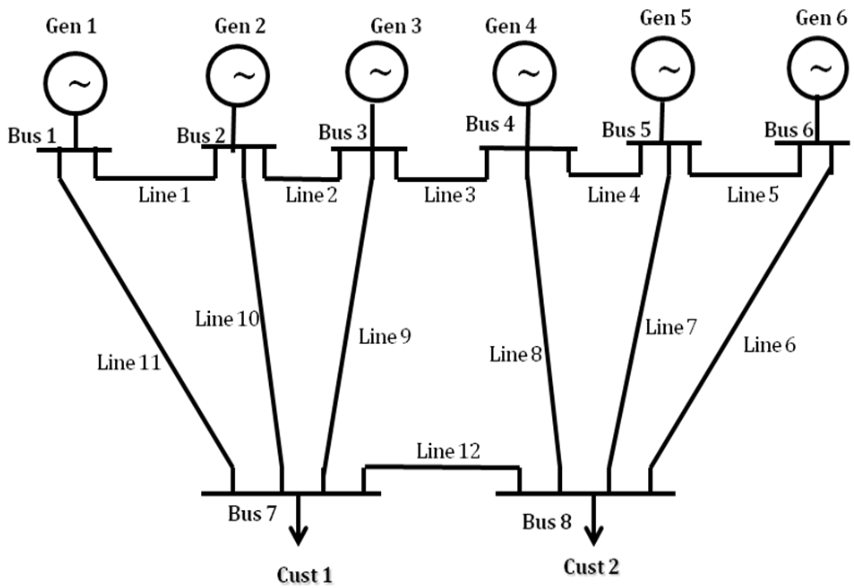

Figure 5.4: An 8-bus test system (case II), consisting of 6 generators 2 customers and 12 lines

Generators' and customers' bid data (Tables 5.6 and 5.7) and loss coefficients were taken from [54, 136, 137]. Optimised values of the bidding quantities for generators and customers, as well as total generation cost, total customer benefit and social profit for the bidding strategies are presented respectively for low, medium and high bidding strategies in Tables 5.8, 5.9 and 5.10.

Generators	a_{gi}	b_{gi}	c_{gi}	Pg_{min}	Pg_{max}	UR_i	DR_i
1	0.00375	2.00	0	50	200	65	85
2	0.01750	1.75	0	20	80	12	20
3	0.00625	1.00	0	15	50	12	15
4	0.00834	3.25	0	10	35	8	16
5	0.02500	3.00	0	10	30	6	9
6	0.02500	3.00	0	12	40	8	16

Table 5.6: Generators' bid data for case II (6 generators and 2 customers, in 2 dispatch periods)

$$B = \begin{bmatrix} 0.000200 & 0.000010 & 0.000015 & 0.000005 & 0.00000 & -0.00003 \\ 0.000010 & 0.000300 & -0.00002 & 0.000001 & 0.000012 & 0.00001 \\ 0.000015 & -0.00002 & 0.00010 & -0.00001 & 0.000001 & 0.000008 \\ 0.000005 & 0.000001 & -0.00001 & 0.00015 & 0.000006 & 0.000050 \\ 0.000000 & 0.000012 & 0.000010 & 0.000006 & 0.000250 & 0.00002 \\ -0.00003 & 0.000010 & 0.000008 & 0.00005 & 0.000020 & 0.000210 \end{bmatrix} \quad 5.14$$

	Customer 1 Low/Medium/High	*Customer 2 Low/Medium/High*
a_{dj} (\$/MWh^2)	-0.06/0.07/0.1	-0.08/0.05/0.09
b_{dj} (\$/h)	20	15
Load demand at Period 1	Min: 100 Max: 150	Min: 50 Max: 100
Load demand at Period 2	Min: 20 Max: 70	Min: 100 Max: 200

Table 5.7: Customers' bid data for case II (6 generators and 2 customers, in 2 dispatch periods)

In the low bidding strategy, bid coefficients, a_{dj} are -0.06 \$/MWh^2 and -0.08 \$/MWh^2 respectively for Customers 1 and 2 in both trading periods. The energy clearing price is \$20/h, and the bid coefficients, b_{dj} for Customers 1 and 2 are \$20 and \$15 respectively in both periods. In period 1, Customer 1 submits a minimum bid of 100MW and a maximum bid of 150MW, and Customer 2 submits a minimum bid of 50MW and maximum bid of 100MW. Under the medium and high bidding strategy, the same generators and customers bid data were used, except for the medium bid coefficients (0.07 \$/MWh^2 and 0.05 \$/MWh^2) and the high bid coefficients (0.1 \$/MWh^2 and 0.09 \$/MWh^2) respectively for Customers 1 and 2 in both trading periods. Also, the generators supplied maximum power demand by both customers (250MW in period 1 and 270MW in period 2) in both the medium and high bidding strategies. Table 5.8 shows the total generation costs, total customer benefits and social profits averaged over 30 runs, obtained using SEA compared with those of DE and PSO with the same set of data for the three bidding strategies.

		Total Benefits ($)	*Total Costs ($)*	*Social Profits ($)*
Low Bidding	M3_SEA1	3,762.28	1,409.82	2,352.45
	M3_SEA2	3,641.26	1,362.95	2,278.31
	M3_SEA3	3,698.17	1,343.88	2,354.28
	DE	4,101.30	989.72	**3,111.50**
	PSO	3,483.80	1,851.00	1,632.80
Medium Bidding	M3_SEA1	13,318.00	1,201.23	12,116.77
	M3_SEA2	13,318.00	1,196.70	12,121.30
	M3_SEA3	13,318.00	1,170.96	**12,147.04**
	DE	13,318.00	1,432.00	11,886.00
	PSO	12,141.00	1,928.00	10,213.00
High Bidding	M3_SEA1	16,140.00	1,169.44	14,970.56
	M3_SEA2	16,140.00	1,125.97	15,014.03
	M3_SEA3	16,140.00	1,115.67	**15,024.33**
	DE	16,140.00	1,432.00	14,708.00
	PSO	15,571.00	1,793.00	13,778.00

Table 5.8: Summary of results for case II (6 generators and 2 customers, in 2 dispatch periods), averaged over 30 runs, and comparison with other approaches

The results show the performance of SEAs to be better than DE and PSO in both medium and high bidding strategies. DE performed better in the low bidding strategy. The values of social profit are generally low in the low bidding strategy; therefore, economic dispatch of the resources with low bids is not advisable in a deregulated electricity market. The M3_SEA3 approach produced the highest social profits in both medium and high strategies. In all the three bidding strategies, based on the prevailing market price, customers submit their bids with the aim to yield higher benefits, while GENCOs submit their bids with the aim to increase their profits. ISO as the market operator regulates the bid resources of these participants, and ensures that social profit is maximised, which is the overall goal of DELD in a deregulated power system. Tables 5.9 shows optimal resource allocation under low, medium and high bidding strategies realised using the M3_SEA3 approach.

	Low Bidding			Medium Bidding			High Bidding		
	Period		Total	Period		Total	Period		Total
	1	2		1	2		1	2	
Cust 1	132.72	69.59		150	70		150	70	
Cust 2	71.50	127.33		100	200		100	200	
Total Dem	204.22	196.92		250	270		250	270	
Total Ben	2261.03	1714.14	3975.17	6575	6743	13,318	7650	8490	16,140
Gen 1	62.72	74.08		66.39	99.32		63.58	93.41	
Gen 2	34.61	55.94		28.01	32.87		25.02	22.13	
Gen 3	48.78	28.68		28.57	18.88		29.39	33.79	
Gen 4	22.19	21.60		18.86	11.23		15.66	22.02	
Gen 5	12.12	20.46		27.26	17.43		20.08	18.63	
Gen 6	33.73	36.41		13.45	14.78		17.26	23.61	
Total Gen	206.05	199.62		251.58	272.55		251.36	272.38	
Total Cost	531.26	643.50	1174.76	455.22	480.38	935.60	414.37	532.70	947.07
Losses	1.83	2.70		1.58	2.55		1.36	2.70	
Social Profit			**$2,800**			**$12,382**			**$15,193**

Table 5.9: Resources allocation in the best of 30 runs of M3_SEA3, under the 3 bidding strategies: low, medium and high, for case II (6 generators and 2 customers, in 2 dispatch periods)

This case has primarily achieved the main objective of BBDELD optimisation of matching both the demand and supply bids within the operational standards of ISO. The results have also successfully proved performance effectiveness of the SEAs in balancing the bids submitted by power generating companies and customers with a consideration of power transmission losses, which were effectively minimised. We can deduce from the results in Table 5.9 that social profits are higher from matching the bids of GENCOs and customers under the high bidding strategy. The implication of this finding is that, in a deregulated market, customers can buy power from any GENCO of their choice participating in the competition.

5.6.3 Case III: 10 Generators, 6 Customers in 12 Dispatch Periods

This problem case involves a 16-bus test system consisting of 10 generators, 6 customers and 24 lines, in 12 dispatch periods shown is Figure 5.5:

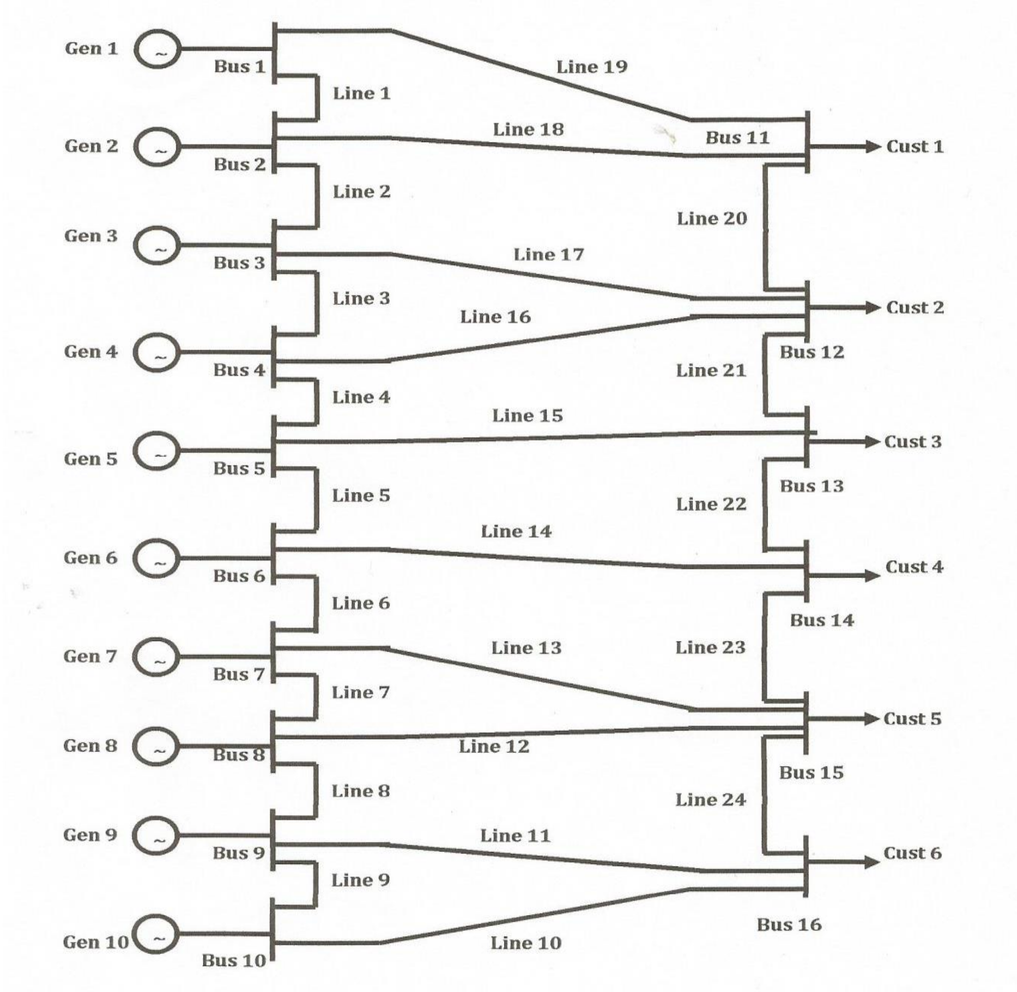

Figure 5.5: A 16-bus test system (case III), consisting of 10 generators 6 customers and 24 lines

We investigated the optimisation performance of M3_SEA1, M3_SEA2 and M3_SEA3 under the high bidding strategy. The generators bid data (Table 5.10) were taken from [109]. We generated the customers bid data (Table 5.11) using the recommendations and specifications of the high bidding strategy from [137, 142], to meet the load demand requirements of [4, 109]. The dispatch was for an arbitrary 12-hour period, and losses were not considered.

Gen.	a_{gi}	b_{gi}	c_{gi}	Pg_{min}	Pg_{max}	UR_i	DR_i
1	0.00043	21.60	958.20	150	470	80	80
2	0.00063	21.05	131.60	135	460	80	80
3	0.00039	20.81	604.97	73	340	80	80
4	0.00070	23.90	471.60	60	300	50	50
5	0.00079	21.62	480.29	73	243	50	50
6	0.00056	17.87	601.75	57	160	50	50
7	0.00211	16.51	502.70	20	130	30	30
8	0.00480	23.23	639.40	47	120	30	30
9	0.10908	19.58	455.60	20	80	30	30
10	0.00951	22.54	692.40	55	55	30	30

Table 5.10: Generators' bid data for case III (10 generators and 6 customers, in 12 dispatch periods)

Customers		***D1***	***D2***	***D3***	***D4***	***D5***	***D6***	***Demand Per Period***
a_{dj} ($/MWh^2)		0.1	0.099	0.097	0.094	0.093	0.09	
b_{dj} ($20/h)		20	19	17	16	15	12	
Maximum Load Demand Bids at each period	1	300	180	130	200	116	110	1036
	2	190	220	100	200	150	250	1110
	3	208	150	250	300	100	250	1258
	4	270	230	256	200	300	160	1406
	5	300	280	240	260	150	250	1480
	6	400	320	170	230	208	300	1628
	7	250	192	350	300	400	200	1702
	8	370	250	350	406	150	250	1776
	9	320	400	200	350	420	234	1924
	10	472	300	400	350	300	250	2072
	11	500	490	250	240	360	306	2146
	12	410	420	380	350	360	300	2220

Table 5.11: Customers' bid data for case III (10 generators and 6 customers, in 12 dispatch periods)

Table 5.12 and Figure 5.6 show optimised values for total generation costs, total customer benefits and social profits, while Figure 5.7 is a load curve.

	Total Generation Cost ($)	***Total Customer Benefit ($)***	***Social Profit ($)***	***Standard Deviation***
M3_SEA1	447,212.98	913,897.45	466,684.47	9437.23
M3_SEA2	462,658.97	913,897.45	451,238.48	14248.36
M3_SEA3	446,679.14	913,897.45	467,218.31	4853.70

Table 5.12: Summary of results for M3_SEA1, M3_SEA2 and M3_SEA3, averaged over 30 runs, for case III (10 generators and 6 customers, in 12 dispatch periods)

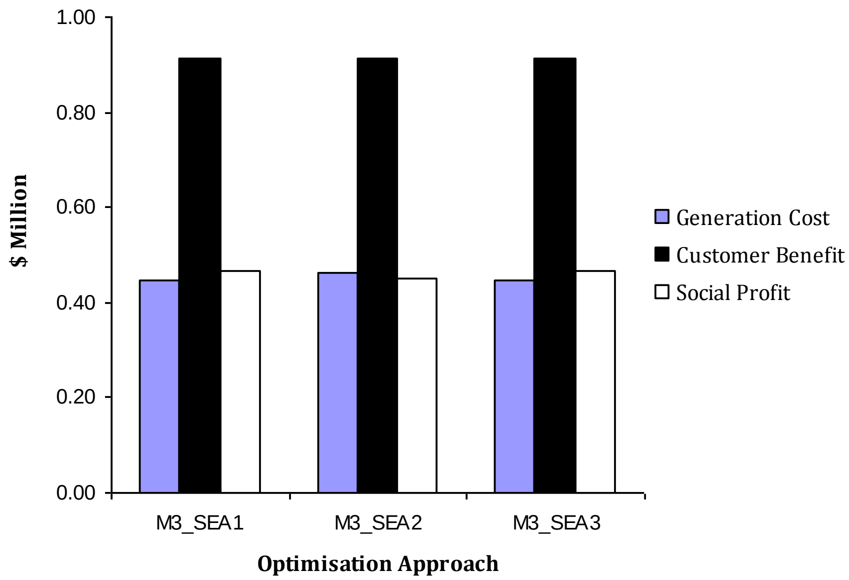

Figure 5.6: Bar chart showing the generation costs, customer benefits and social profits of M3_SEA1, M3_SEA2 and M3_SEA3, averaged over 30 runs, for case 3 (10 generators and 6 customers, in 12 dispatch periods)

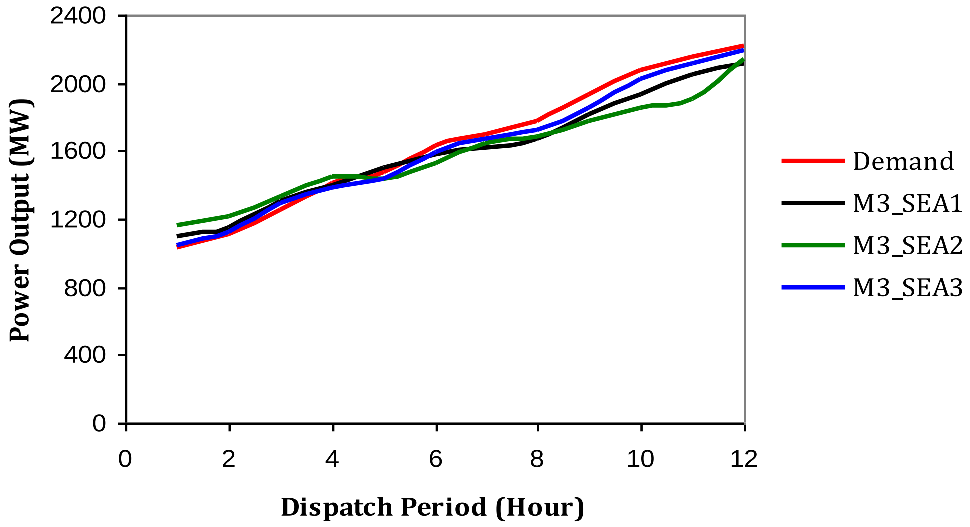

Figure 5.7: Load curve showing the hourly outputs of M3_SEA1, M3_SEA2 and M3_SEA3, in comparison with total hourly demand for case III (10 generators and 6 customers, in 12 dispatch periods)

As seen in Table 5.12 and Figure 5.6, for fixed customer benefits of $913897.45, M3_SEA3 yielded a higher social profit of $467,218.31, against $466,684.47 and $451,238.48 for M3_SEA1 and M3_SEA2 respectively. Although there is a very small difference in the values of the average social profits from M3_SEA3 and M3_SEA1, but there is much significant deviation between the values obtained using M3_SEA1 (9437.23) compared to those of M3_SEA3 (4853.70) in both social profits and total generation costs, averaged over 30 runs. From these results, and the load curve of Figure 5.7, all three methods show promising performance. Standard T-tests (one-tailed) with significance level $p < 0.1$ reveals 100% confidence that both M3_SEA1 and M3_SEA3 are better than M3_SEA2. However, there is no significant difference between M3_SEA1 and M3_SEA3, although we tend towards the suggestion that the latter is better, having slightly higher social profit, lowest standard deviation, and its load trend is closest to the demand trend than the former.

5.7 Summary

Based on the prevailing market price, customers (DISCOs) submit their bids with the aim of getting higher benefits, while power generating companies (GENCOs) submit their bids with the aim of minimising their costs (and consequently increasing their profits). The market regulator (ISO) therefore plays an important role in regulating the resources of GENCOs as well as the needs of DISCOs. The results of this chapter is very useful to ISO as a guide in the process of matching both the supply side and demand side bids in an economic way with the overall aim of maximising the social profit throughout the dispatch period.

This chapter investigated the use of smart EAs to maximise social profit in the context of a BBDELD formulation in a deregulated electricity environment. This formulation builds in part on earlier works in the SELD and DELD problems. We briefly examined the introduction of deregulation to global electricity market, and identified price-based DELD and bid-based DELD as profit-oriented DELDs involved in optimal transaction dispatch formulations in a deregulated market. The chapter reviewed related work in BBDELD, and investigated the effects of three bidding strategies on the social profit, where we found that social profits are greater from matching the bids of generating companies and customers (distribution companies) under the high bidding strategy. We compared results of the best two versions of the smart mutation operator with previous results in literature involving two test cases (3 units, 2 customers in 2 dispatch periods and 6 units, 2 customers in 2 dispatch periods). From the superior results obtained with the smart EA in these test cases over other approaches, we defined and presented results for a new and larger test case (10 units, 6 customers in 12 dispatch periods) that yielded very high social profit.

Chapter 6

Case Study: Application of BBDELD to Nigerian Electrical Power Sector Reforms

Nigeria, a West African country is bounded by Gulf of Guinea, Republics of Benin, Niger, Cameroon and Chad to the south, west, north, east and north-east respectively; and occupies a total area of 923,768 sq km. With a population of about 178 (Nov. 2014 estimate), she is ranked 7th in the world [143] [144]. In this chapter, we adopt a case study approach, applying the research to the Nigerian electrical power system that has recently been deregulated. We identify the challenges to effective power provision in the country that led to various reforms of the power sector. We describe the design and implementation of a constrained elitist genetic algorithm for the country's power system before the sector reforms, adapt the BBDELD formulation to the country's deregulated electricity market using the best two versions of the smart mutation operator, and investigate the ability of the SEA to deal with larger scale energy optimisation task in the bid based dispatch problems.

6.1 History of Electrical Power Supply in Nigeria

In Nigeria, electrical energy supply started in 1896 when two small generators were used to generate electricity at Ijora, Lagos by the British Colonial Administration [145]. In 1929, Nigerian Electricity Supply Company (NESCO) commenced operation with the establishment of a hydro-electric power station at Kura Falls in Plateau State [146]. The Electricity Corporation of Nigeria (ECN) and the Niger Dams Authority (NDA) were established in 1950 and 1962 to manage the diesel/coal plants, and the hydro electrical sub-sector respectively [147], with ECN solely charged with the responsibility of distributing and the sale of electricity. The merger of ECN and NDA formed the National Electric Power Authority (NEPA) in 1972, whose mandate was managing, maintaining, and overseeing electricity generation, transmission and distribution in the country [148]. It was restructured in 2005 and renamed Power Holding Company of Nigeria (PHCN) [149].

The main readily available fuel sources for electricity in Nigeria are coal, gas, oil and hydro. Until recently, the country had 14 generating plants consisting of 11 thermal and 3 hydro, which supply electricity to the National Grid, a 31-bus system made up of 4889.2km of 330kV line, 6319.33km of 132kV, 6098MVA transformer capacity at 330/132kV and 8090MVA transformer of 132/33kV. Table 6.1 shows the number, type, location of the generating plants, as well as their installed/available units and capacities as at 2008.

S/N	*Plants*	*Types*	*Location*	*No. of Units*	*Available Units*	*Installed Capacity (MW)*	*Available Capacity (MW)*
1	Sapele	Thermal	Delta	10	1	1020	90
2	Afam	Thermal	Rivers	20	3	702	350
3	Egbin	Thermal	Lagos	6	4	1320	880
4	Delta	Thermal	Delta	18	12	840	540
5	Ijora	Thermal	Lagos	9	9	270	270
6	Okpai	Thermal	Cross River	3	2	480	480
7	Omoku	Thermal	Rivers	6	4	150	100
8	Geregu	Thermal	Kogi	3	3	414	414
9	Ajaokuta	Thermal	Kogi	2	2	110	100
10	Omotosho	Thermal	Ondo	8	2	335	80
11	Olorunsogo	Thermal	Ogun	8	2	335	80
12	Kainji	Hydro	Niger	8	6	760	440
13	Jebba	Hydro	Niger	6	6	540	386
14	Shiroro	Hydro	Niger	4	2	600	600
		Total		**111**	**58**	**7876**	**4816**

Table 6.1: Power generating plants in Nigeria as at 2008 [148]

6.2 Challenges to Effective Power Provision in Nigeria

The country's geographical terrain consists of thick evergreen rainforests in the southern lowlands with lots of rivers, through the hills and plateaus of the central middle belt zone, to the open savannah plains of the far north. The climate varies with equatorial in south, tropical in centre and arid in north. This distribution trend presents enormous challenges for the efficient and effective provision of electrical power to all parts of the country, and demand for electrical energy is continuously on high increase to residential, commercial as well as industrial consumers, with the highest increase on residential consumption. Figure 6.1 shows a 10 years trend of electrical energy consumption by these three classes of consumers.

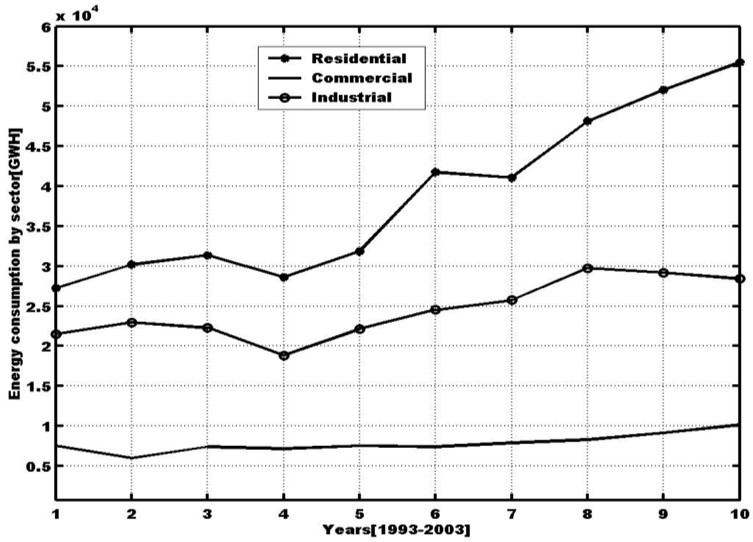

Figure 6.1: Trend in Electrical Energy Demand in Nigeria from 1993 to 2003 [150]

The greatest problem facing the country's electrical power sector is therefore increasing residential demand. This has posed several challenges in the generation, transmission and distribution sub-sectors.

6.2.1 Generation

Nigerian electricity sector is characterised by low generation [24]. As seen in Table 6.1, of the 111 installed units, only 58 units are being utilised. With an installed capacity of 7876MW and available capacity of 4816MW, total daily power generation at the end of 2008 was 2400MW. The generation rose to 3700MW at the end of 2009, and 4420MW in 2011. Identified causes of the low power generation include, but not limited to the following [147, 149]:

1. Obsolete equipment;
2. Poor maintenance culture;
3. Insufficient funding;
4. Employees' aptitude and sabotage;
5. Lack of interest to explore other renewable energy sources.

6.2.2 Transmission

Nigeria's electricity transmission network does not currently cover every part of the country. It does not also have the capacity to transmit all the power generated. Factors that account for this inefficiency include, but not limited to the following [149]:

1. The country's geographic terrain;

2. Federal government's sole funding;
3. Vandalisation of lines;
4. Lack of modern monitoring and communication technologies;
5. Outdated network grid;
6. Lack of regular human capacity re-training schemes.

6.2.3 Distribution

Currently, electricity distribution network in most parts of the country is not very efficient, resulting in consumers experiencing low voltage and inaccurate billing. The distribution sub-sector is characterised by [149]:

1. Inadequate, obsolete distribution network with overloaded transformers;
2. Poor billing system;
3. Regular theft and vandalism of distribution equipment;
4. Employees' sabotage and non payment of bills by consumers;
5. Poor communication equipment and customer service;
6. Lack of regular human capacity re-training schemes.

The result of these challenges is erratic, epileptic and unreliable supply of electricity, forcing most consumers to resort to self-generation of power. This situation has no doubt created a harsh environment for both national development as well as discouraging foreign investments. The power problem has been top in the list of priorities of successive leadership in Nigeria, making the country to witness several reforms in the electrical sector.

6.3 Electrical Power Sector Reforms

Nigeria's electricity per capita in both generation and consumption is one of the lowest in the world, making it the most problematic electricity sector. Figure 6.2 shows a pie chart comparison of Nigeria's generation per capita with contemporary developing nations in 2009 [24]. With a net electricity generation of 3700MW, Nigeria's generation per capita was 24.67MW for a population of 150 million, against South Africa's 1510.2MW for a population of 44 million and electricity generation of 46000MW; United Arab Emirate's 1184MW for a population of 4 million and electricity generation of 4740MW; Malaysia's 960MW for a population of 25 million and electricity generation of 24000MW; and Iran's 676.9MW for a population of 65 million and electricity generation of 44,000MW.

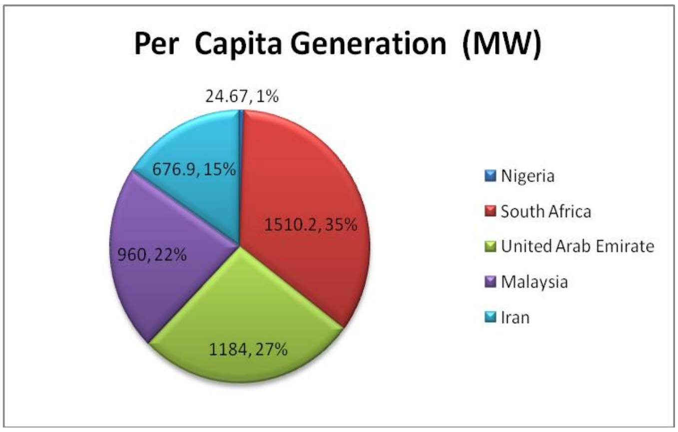

Figure 6.2: Per capita generation of Nigeria and selected developing energy Nations [24]

Table 6.2 compares the electricity consumption per capita of Nigeria (in KWh) with selected developing and developed countries from 2009 to 2013. Its worldwide ranking was 188th in 2013 [144, 151, 152, 153].

S/N	*Countries*	*2009*	*2010*	*2011*	*2012*	*2013*
1	Iceland	51,259	51,440	52,374	52,621	51,024
2	Norway	23,860	24,891	23,174	24,558	23,538
3	Canada	15,120	16,154	16,406	16,020	14,714
4	UAE	9,998	9,845	9,389	13,281	13,947
5	USA	12,914	13,395	13,246	11,920	12,391
6	France	7,340	7,735	7,289	7,023	6,878
7	Germany	6,753	7,162	7,081	6,697	6,754
8	Japan	7,838	8,378	7,848	6,750	6,750
9	UK	5,686	5,745	5,516	5,467	5,168
10	South Africa	4,532	4,654	4,694	4,373	4,222
11	China	2,633	2,944	3,298	3,494	3,494
12	Brazil	2,202	2,381	2,438	2,286	2,199
13	Mexico	1,870	1,916	2,092	1,579	1,773
14	Gabon	1,004	974	907	995	809
15	Cameroon	261	258	256	176	251
16	Ghana	280	299	344	284	248
17	Angola	255	247	248	186	202
18	Sudan	114	133	143	111	135
19	Kenya	146	155	155	125	128
20	Nigeria	120	135	149	107	103

Table 6.2: Electricity consumption per capita (in KWh) of selected countries from 2009 to 2013 [144, 151, 152, 153]

Only about 36% of the total population have access to the public grid power [24]. The wide gap between electricity demand and the installed/available capacity in the generating stations has resulted to acute power shortage in the country, leading to self-generation of power by residential, commercial and industrial consumers. It is estimated that an average of ₦2billion (about $12 million) is spent on this per week [154]. There are enormous opportunities in Nigeria's electricity power sector for private investment but for the witnessed poor performance, leading to huge losses. This has greatly hindered social development and economic growth. The factors that have led to this inefficiency as well as their consequences have been outlined in sub-sections 6.3.1, 6.3.2 and 6.3.3. But with the successes recorded in the deregulation of other sectors (e.g. telecommunications), the federal government realised that the future of the power sector rests on private sector participation.

The Privatisation and Commercialisation Decree No. 25 of 1988 established the Technical Committee on Privatisation and Commercialisation (TCPC), and led to the review of the failures of NEPA and consequently proposals to commercialise same [155]. With the enactment of the Public Enterprises Act of 1999, TCPC was replaced by the Bureau of Public Enterprises (BPE) [156]. This shifts emphasis from commercialisation to privatisation, encouraging core investors and promoting foreign investment. With the continued low electricity supply experienced in the country, the federal government in 2000 formed Electric Power Implementation Committee (EPIC), which prepared the National Electric Power Policy (NEPP) in 2001, with the following objectives [154]:

1. Attract private investors from both within and outside the country;
2. Establishment of an independent regulatory agency;
3. Development of a wholesale electricity market;
4. Establishment of a consumer assistance fund to ensure the efficient and targeted application of subsidies to less privileged Nigerians;
5. Establishment of a Rural Electrification Agency (REA) to manage the rural electrification fund.

These objectives were legally backed up by the Electric Power Sector Reform Act (EPSRA) of 2005, removing the operational and regulatory responsibilities of the electricity sector from the federal government [145]. NEPA was renamed PHCN, and unbundled into 18 successor companies, comprising of: 6 GENCOs, 1 TRANSCO and 11 DISCOs [157]. The Act established the Nigeria Electricity Regulatory Commission (NERC) as an independent industry regulator, for effective monitoring of tariffs and quality of service, with supervisory input from the Federal Ministry of Energy.

In August 2010, a Roadmap on the power sector reform was presented by the Federal Government to the stakeholders (National Council on Privatisation, Bureau of Public Enterprises, Ministry of Energy and NERC) and investors in the power sector [158]. Private sector investment/participation was encouraged for generation and distribution, while the TRANSCO belonged exclusively to the Federal Government.

Until its liquidation in 2011, the 18 successor companies operated as subsidiaries of PHCN, but have now inherited all its operations. Consequently, the Nigerian Electricity Liability Management Company (NELMCO) was created to inherit and manage all the liabilities of the successor companies. The Nigerian Bulk Electricity Trading Plc (NIBET) was created and inaugurated in 2011 as middle machinery between the GENCOs and the DISCOs through bulk electricity purchase from GENCOs and Independent Power Producers (IPP), and re-sell same to DISCOs until the latter are financially able to function independently [158]. The reform created a platform/enabling environment for IPP's participation in electricity generation and distribution, with the granting of more IPP licenses. Table 6.3 shows the approximate sector's reform targets for generation, transmission and distribution from 2010 to 2013.

Section	***Yearly Target Capacity (MW)***			
	2010	**2011**	**2012**	**2013**
Generation	6,000	9,867	11,879	14,218
Transmission	5,500	7,488	8,986	9,885
Distribution	5,300	7,485	8,061	9,057

Table 6.3: Nigerian power system reform targets [158]

In line with the vision 20:2020 on energy sector (the target of being among the world's top 20 energy sector by the year 2020) [158], the Federal Government aims a target generation of 40,000MW with an annual investment of ₦1.5 trillion ($10 billion) suggested by the reform roadmap. In 2012, a true deregulation was put in place, with the successor companies (Table 6.4) commencing independent activities in 2013 with a full privatisation [159]. The current research is therefore a timely intervention on this subject.

GENCOs	TRANSCO	DISCOs
Afam Power Plc	Transmission Company of Nigeria	Abuja Electricity Distr. Co. Plc
Geregu Power Plc		Benin Electricity Distr. Co. Plc
Kainji Power Plc		Eko Electricity Distr. Co. Plc
Sapele Power Plc		Enugu Electricity Distr. Co. Plc
Shiroro Power Plc		Ibadan Electricity Distr. Co. Plc
Ughelli Power Plc		Ikeja Electricity Distr. Co. Plc
		Jos Electricity Distr. Co. Plc
		Kaduna Electricity Distr. Co. Plc
		Kano Electricity Distr. Co. Plc
		Port Harcourt Electricity Distr. Co. Plc
		Yola Electricity Distr. Co. Plc

Table 6.4: PHCN Successor Companies [159]

6.4 Case Study I: Application of SEA to the Nigerian Power System before Deregulation

In this section, we applied the smart evolutionary algorithm to solve the problem of real power dispatch for the Nigerian power system grid before the sector reforms that preceded the deregulation. The aim is to reduce both the total cost of generation and transmission power loss, while maintaining an acceptable generation output in the context of SELD. The grid involves a 31-bus, 330kV network interconnecting 4 thermal and 3 hydro stations to several load centres.

6.4.1 Related Work

Very limited work in ELD has been done so far in relation to the electrical power system of Nigeria. A vast majority of published work ([145, 146, 147, 148, 149, 150, 154, 157, 159]), amongst others merely describes theoretically the challenges and reforms within of the power sector.

In [53], conventional genetic algorithm (CGA) and micro genetic algorithm (μGA) were developed for the ELD. The effectiveness of the approaches was tested with the IEEE 6-bus and 31-bus Nigerian grid system. While satisfactory results were realised in both approaches over classical method (CM) and Hopefield Neural Network (HNN) [160], CGA has a major drawback of long computation time, while μGA worked well on small population to reduce computation time [161], with mutation set to 0 [53]. This leads to a quicker convergence, and may not be a good approach for bigger population, where solutions might be stuck in local optima. Much genetic variations are not introduced. These problems were overcome in the present application case study, whose strength lies in the preservation of the “elite members” and copying them across generations to avoid losing promising solutions in a given generation.

Power flow and contingency analysis were carried out in [162] to compute bus voltages, bus powers and transmission losses in a section of the Nigerian transmission network. This was achieved using Guass-Siedel iteration method to evaluate voltages for the 330kV ring network (Benin – Ikeja West – Ayede – Oshogbo – Benin), to determine the weak areas resulting in high technical losses in order to improve the efficiency of the network. The 330kV electricity transmission network of Nigeria consists of a long radial transmission lines and this loop connection within the network [163]. Performance comparison of the results obtained was made with Newton-Raphson method in Power World Simulator environment [164] which proved to be satisfactory and reliable. Technical and financial losses in the network were estimated in [163]. Results from all the studies show that the present state of the Nigerian electricity grid is not up to expectation, requiring lots of modifications.

In [165], power distribution planning problem of Nigeria was solved using the decomposition approach and integer programming method. The large problem was divided into three optimisation stages: substation, feeder and outage cost optimisations, with each stage formulated as a quadratic mixed integer programming problem and solved sequentially. Radial distribution network system used in the work ensured that electricity leaves the substation, passing through the network without any connection to other distributors. The study was however limited to Enugu DISCO.

6.4.2 Problem Formulation

In a system of generating units connected to a transmission network, modelling and formulating the ELD problems in relation to Nigerian power system involve three different cases:

(i) Neglecting both generators’ limits and power losses;
(ii) Including generators’ limits, but neglecting losses;
(iii) Including both generators’ limits and power losses.

In case (i), the total power demand is equal to the sum of all generating units output, where a cost function is assumed to be known for each unit. The problem here is the computation of total generation cost, C_T, and determination of the generators output for such that C_T as defined in (2.11) and (2.12) is minimised, subject to (2.13). In case (ii), the generating limits constraint of (2.14) is considered. Case (iii) involves the power transmission losses, P_L, of (2.18), the power balance constraint of (2.15), in addition to the generating limits constraint of (2.14), which is the case considered in this application study.

6.4.3 Optimisation Approach

The following is the work flow/description of the algorithm adopted in the optimisation approach for this application case study. The key concept used here is elitism. It is a strategy that ensures the best solutions of the previous generation are maintained in the next population.

(i) Generate an initial population P of size N.

(ii) Evaluate the fitness of each solution, S.

(iii) Select the best solution(s) S_{best}, to form the mating/breeding pool (parents).

(iv) Perform breeding (crossover and mutation operations) on the parent solutions from the mating pool and obtain a new population, P_{new}. The smart mutation approach is adopted in this SELD problem, using SEA1, SEA2 and SEA3.

(v) Evaluate the fitness of each solution S of P_{new} with S_{best}. If the optimum solution is found or maximum generation is reached (stopping criteria), output result, otherwise process elitist strategy.

(vi) In the elitist process, if the fitness of each solution in P_{new} is less than the fitness of S_{best}, replace the worst solution of P_{new} with S_{best}. Sort chromosomes in the resultant population. Based on elitism rate ($x\%$ of N), compute the elites and copy to next generation. Go back to step *(ii)* above.

6.4.4 Experimental Design, Results and Discussion

In practical implementation, (2.15) can be re-written as an error function in (6.1), with an aim to minimising it.

$$\varepsilon = \left| \sum_{i=1}^{ng} P_{gi} - P_D - P_L \right| \tag{6.1}$$

Convergence is achieved when the error decreased to a tolerance level. To speed up the convergence and realise the best solution, the fitness values were normalised to a range between 0 and 1, with the fitness function defined by (6.2), where *k* is a penalty factor which reflects the violation of the power balance constraint, assigning a high cost of penalty to affected ones far from the feasible region [108, 101], and σ a scaling factor defined by (6.3).

$$f = \frac{1}{1 + (k * \varepsilon * \sigma)} \tag{6.2}$$

$$\sigma = \frac{1}{P_D} \tag{6.3}$$

This process iterates until convergence occurs. With the best solution obtained, the total power generated, total generation cost, and total transmission loss are computed. Table 6.5 shows parameters for the capacities and cost coefficients from [53], for the generators in 4 of the 7 thermal stations used in this case study.

Power Station	***P_g^{min} (MW)***	***P_g^{max} (MW)***	***a (₦/hr)***	***B (₦/MWhr)***	***c (₦/MW^{2}hr)***
Egbin	275.0	1100.0	12787.00	13.10	0.031
Sapele	137.5	550.0	6929.00	7.84	0.130
Delta	75.0	300.0	525.74	-6.13	1.200
Afam	135.0	540.0	1998.00	56.00	0.092

Table 6.5: Nigerian thermal stations generating capacities and cost coefficients

Parameters for the 3 hydro plants (Shiroro, Kainji and Jebba) were fixed according to the respective utility's operating practices [53]. The power loss was computed using (2.18), with the loss formula (B-coefficients) for the Nigerian power system shown in (6.4), (6.5) and (6.6), as provided in [53].

$$B = \begin{bmatrix} 0.0037 & 0.0002 & -0.0074 & 0.0005 & 0.012 & -0.0076 & -0.0036 \\ 0.0002 & 0.0103 & 0.0033 & -0.0031 & 0.0022 & 0.0005 & -0.0011 \\ -0.0074 & 0.0033 & -0.0076 & 0.0023 & 0.013 & 0.0042 & 0.0152 \\ 0.0005 & -0.0031 & 0.0023 & 0.0056 & 0.0149 & 0.0004 & -0.0096 \\ 0.012 & 0.0022 & 0.013 & 0.0149 & 0.0935 & -0.0248 & -0.1354 \\ -0.0076 & 0.0005 & 0.0042 & 0.0004 & -0.0248 & 0.0127 & 0.0649 \\ -0.0036 & -0.0011 & 0.0152 & -0.0096 & -0.1354 & 0.0649 & 0.0769 \end{bmatrix} \tag{6.4}$$

$$B_0 = \begin{bmatrix} -0.0154 \\ -0.0119 \\ -0.0812 \\ -0.0881 \\ 0.0049 \\ -0.2679 \\ 0.2456 \end{bmatrix} \tag{6.5}$$

$$B_{00} = 0.2278 \tag{6.6}$$

SEA1, SEA2 and SEA3 were applied using these data. For the purpose of comparison with other approaches in literature with the same set of data (CGA and μGA), the genetic parameters were set as shown in Table 6.6.

Parameters	***Values***
Generation	350
Population size	100
Selection method	Tournament
Crossover rate	0.7
Mutation rate	0.01
Smart Mutation Probability (in SEA2)	0.6
Elitism rate	10%
Penalty factor	50,000

Table 6.6: Experimental parameters for the application of SEA1, SEA2 and SEA3 to the Nigerian Power System before Deregulation

Table 6.7 summarises the results for the application of SEA1, SEA2 and SEA3 to the Nigerian Power System before deregulation, averaged over 30 runs, and compared with CGA and μGA [53]; while Table 6.8 gives the resources allocation in the best of 30 runs of all the approaches. As seen in Table 6.8, the total generation cost and power loss of ₦99,143.50/hr and 9.45MW respectively are the lowest and best obtainable from literature till date.

Cost (₦/hr)	*SEA1*	*SEA2*	*SEA3*	*CGA [53]*	*μGA [53]*
Average	106,634.97	105,813.69	102,752.05	-	-
Std Dev	5,668.90	5,347.82	4,707.22	-	-
Maximum	117,312.34	116, 273.05	112,417.22	-	-
Minimum	95,072.02	100,504.81	91,701.62	110,640	107,540

Table 6.7: Summary of results for the application of SEA1, SEA2 and SEA3 to the Nigerian Power System before Deregulation, over 30 runs, and comparison with other approaches

	CGA [53]	*μGA [5]*	*SEA1*	*SEA2*	*SEA3*
Egbin (Pg_1)	877.15	1011.39	1011.90	984.54	1044.40
Sapele (Pg_2)	194.64	173.28	276.57	178.22	279.05
Delta (Pg_3)	106.49	111.18	109.57	77.72	83.12
Afam (Pg_4)	371.13	261.43	146.52	303.36	135.99
Shiroro (Pg_5)	490.00	490.00	490.00	490.00	490.00
Kainji (Pg_6)	350.00	350.00	350.00	350.00	350.00
Jebba (Pg_7)	450.00	450.00	450.00	450.00	450.00
Total generated	2839.41	2847.28	2834.56	2834.84	2832.55
Total demand	2823.10	2823.10	2823.10	2823.10	2823.10
Power loss	16.31	24.18	11.46	11.74	9.45
Total cost (₦/hr)	110,640.00	107,540.00	103,264.07	102,940.36	99,143.50

Table 6.8: Resources allocation in the best of 30 runs of SEA1, SEA2 and SEA3 applied to the Nigerian Power System before Deregulation, and comparison with other approaches

Tables 6.7 and 6.8 show that SEA1, SEA2 and SEA3 approaches performed better than μGA and CGA, two similar approaches identified in literature with the same set of data in terms of both generation cost and power loss in this application case; with SEA3 having a comparatively lowest cost and power loss than all the approaches. A standard T-test (one-tailed) with significance level $p < 0.1$ shows 99.7% confidence that SEA3 is better than SEA1 in this problem. We also confirmed that SEA3 is superior to SEA2 with 97.5% confidence. There is insufficient information to compare statistically with μGA and CGA. On the overall, the results indicate that SEA3 is the best solution approach for this problem. The SEAs kept the good features of EA over μGA in being able to optimise problems with higher population size, while eliminating the major draw-back of CGA which is long execution time.

6.5 Case Study II: A BBDELD Solution Approach for the Nigerian Deregulated Electricity Market

With the approval and granting of independent power provision (IPP) licenses as part of the sector reforms preceding the market deregulation, a total number of 43 electricity power stations from the initial 14 stations was targeted, for a capacity of 24,106MW

[148, 166]. The 6 generating companies (GENCOs) share these 43 power stations in terms of geographical proximity to reduce transmission losses. At the moment, their status vary from existing, ongoing, new IPP commissioned IPP, approved IPP, approved licenses IPP, license granted IPP [166]. In this section, we developed a new BBDELD for the Nigerian deregulated electricity market, consisting of 40 generators and 11 customers (DISCOs) in a 24 trading (dispatch) periods.

In the proposed model, we investigated the performance of two optimisation solution approaches that have recently provided superior results on related benchmark problems for the BBDELD in 12 trading periods, under the high bidding strategy (M3_SEA1 and M3_SEA3), and we now adapt them for larger energy optimisation scenario in 24 trading (dispatch) periods. The generators bid data with valve-point loading effects were taken from [167]; a larger and more complex test case than previous experiments. There is therefore a tendency of having several local optima, and consequently a difficulty in locating global optima. Solutions involving this large test case (total load demand of 10,500MW) had been realised using EP [167], BCO [168], PSO [169, 170], DE [45], BBO [171] SI [172, 173], MGSO [174], SA [175]. But all these were for an SELD problem. This therefore makes the current test case study very challenging, but appropriate and timely. We generate the customers bid data (Table 6.9) using the recommendations and specifications of the high bidding strategy from [137, 142], to meet the load demand requirements of [4, 109], with modifications to incorporate the larger generators bid data of [167], as shown in Table 6.10.

Customers	***D1***	***D2***	***D3***	***D4***	***D5***	***D6***	***D7***	***D8***	***D9***	***D10***	***D11***
a_{dj} *($/MWh^2)*	0.1	0.099	0.098	0.097	0.096	0.095	0.094	0.093	0.092	0.091	0.09
b_{dj} *($/h)*	20	19	18	17	16	15	14	13	12	11	10

Table 6.9: Customers bid data for the Nigerian deregulated electricity market

Hour	*D1 (MW)*	*D2 (MW)*	*D3 (MW)*	*D4 (MW)*	*D5 (MW)*	*D6 (MW)*	*D7 (MW)*	*D8 (MW)*	*D9 (MW)*	*D10 (MW)*	*D11 (MW)*	*Total (MW)*
1	1300	1000	1200	1100	1060	1100	1000	700	500	800	600	10,360
2	1400	1300	1200	900	1000	1000	900	900	800	600	500	10,500
3	1080	1300	1500	1400	900	1100	1200	1400	700	1000	1000	12,580
4	1700	1800	1560	1200	1600	1500	1000	1000	1000	800	900	14,060
5	1400	1800	1200	1100	1500	1700	1200	1500	1000	1100	1300	14,800
6	2100	1200	1700	1300	1080	2000	1400	1400	1300	1600	1200	16,280
7	1500	1920	1500	2000	2100	2000	1100	1400	1000	1200	1300	17,020
8	2400	2500	2000	2060	1400	1500	1300	1000	1500	1000	1100	17,760
9	2200	1400	2000	2500	2200	2340	1600	1200	1400	1300	1100	19,240
10	2720	2000	2600	1500	1000	2500	1400	1200	2000	1800	2000	20,720
11	2000	1900	1500	2100	1600	2060	2500	2300	2100	1900	1500	21,460
12	3100	2200	1800	2500	2000	2400	2200	1600	1300	1700	1400	22,200
13	2720	2000	2600	1500	1000	2500	1400	1200	2000	1800	2000	20,720
14	2200	1400	2000	2500	2200	2340	1600	1200	1400	1300	1100	19,240
15	2400	2500	2000	2060	1400	1500	1300	1000	1500	1000	1100	17,760
16	2000	1600	1300	1240	1200	1900	1600	1400	1000	1200	1100	15,540
17	1400	1800	1200	1100	1500	1700	1200	1500	1000	1100	1300	14,800
18	2100	1200	1700	1300	1080	2000	1400	1400	1300	1600	1200	16,280
19	2400	2500	2000	2060	1400	1500	1300	1000	1500	1000	1100	17,760
20	2720	2000	2600	1500	1000	2500	1400	1200	2000	1800	2000	20,720
21	2200	1400	2000	2500	2200	2340	1600	1200	1400	1300	1100	19,240
22	2100	1200	1700	1300	1080	2000	1400	1400	1300	1600	1200	16,280
23	1500	1100	1300	1220	1200	1400	1100	1200	1000	1200	1100	13,320
24	1040	1300	1200	1400	900	1100	1200	1300	700	900	800	11,840

Table 6.10: Hourly maximum customers' load demands for the Nigerian deregulated electricity market

The total hourly load demands therefore range from 10,360MW to 22,200MW depending on the time of the day; in line with the targeted generating capacity of 24,106MW for the Nigerian deregulated electricity market following the power sector reforms [148, 166]. Due to lack of related data on loss coefficients in literature for comparison and validation purposes, transmission losses were neglected.

For the purpose of comparison with other approaches in literature with the same set of data (where appropriate), the following parameters Table 6.11 were used.

Parameters	***Values***
Generation	200
Population size	100
Selection method	Tournament
Crossover rate	0.7
Mutation rate	0.01
Smart Mutation Probability (in SEA2)	0.6
Elitism rate	10%
Penalty factor	50,000
Scaling factor	0.2

Table 6.11: Experimental parameters for use in the BBDELD for the Nigerian deregulated electricity market

Table 6.12 and Figure 6.3 show optimised values for total generation costs, total customer benefits and social profits of the two optimisation approaches averaged over 30 runs, which shows both approaches realising very high social profit, with M3_SEA3 slightly higher.

	M3_SEA_1	*M3_SEA_3*
Total Customer Benefit ($)	69,816,712.80	69,816,712.80
Total Generation Cost ($)	2,878,213.85	2,854,985.59
Average Social Profit ($)	66,938,498.95	66,961,727.21
Standard Deviation	31,279.67	18,853.54
Minimum Social Profit ($)	66,922,574.43	66,933,214.72
Maximum Social Profit ($)	66,964,004.04	67,011,371.19

Table 6.12: Summary of results of M3_SEA1 and M3_SEA3, averaged over 30 runs, in the BBDELD for the Nigerian deregulated electricity market

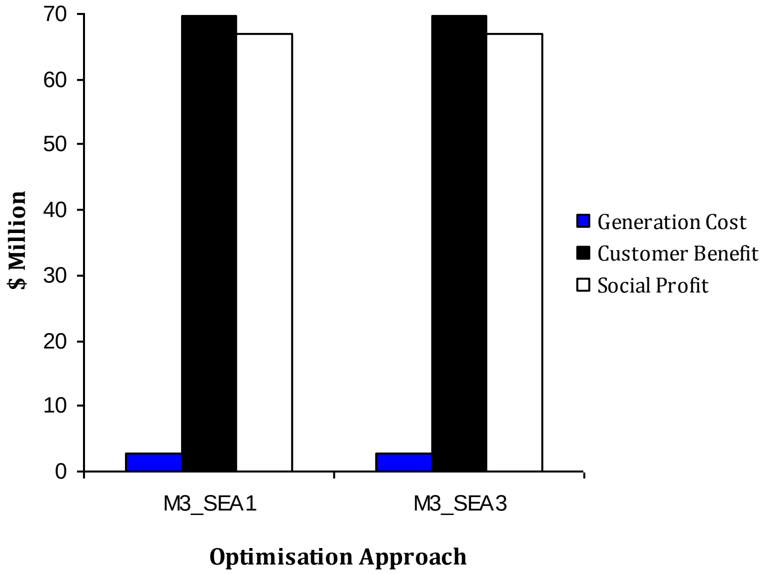

Figure 6.3: Bar chart showing the generation costs, customer benefits and social profits of M3_SEA1 and M3_SEA3, averaged over 30 runs, in the BBDELD for the Nigerian deregulated electricity market, consisting of 40 generators and 11 customers, in 24 dispatch periods

To explore the relative performance of the two approaches, we considered the distribution pattern of social profits across the 30 independent runs as shown in Figure 6.4. Standard T-test (one-tailed) with significance level $p < 0.1$ was also applied. We confirmed that M3_SEA3 is superior to M3_SEA1 with 99.9% confidence, and the best for the current large scale problem. Figure 6.5 is a load curve comparing the two approaches with the total hourly load demands across the entire dispatch period.

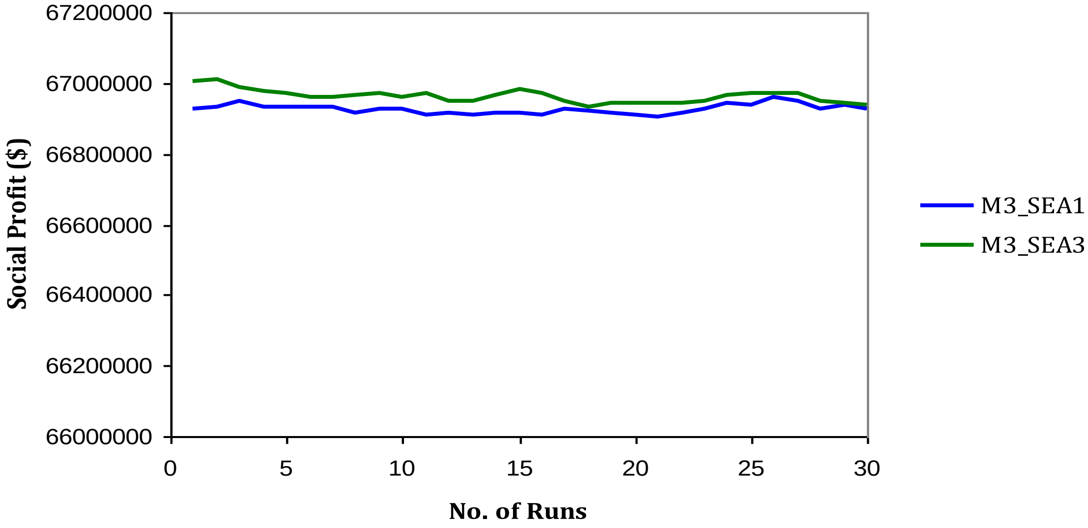

Figure 6.4: Distribution of social profits of M3_SEA1 and M3_SEA3, averaged over 30 runs, in the BBDELD for the Nigerian deregulated electricity market

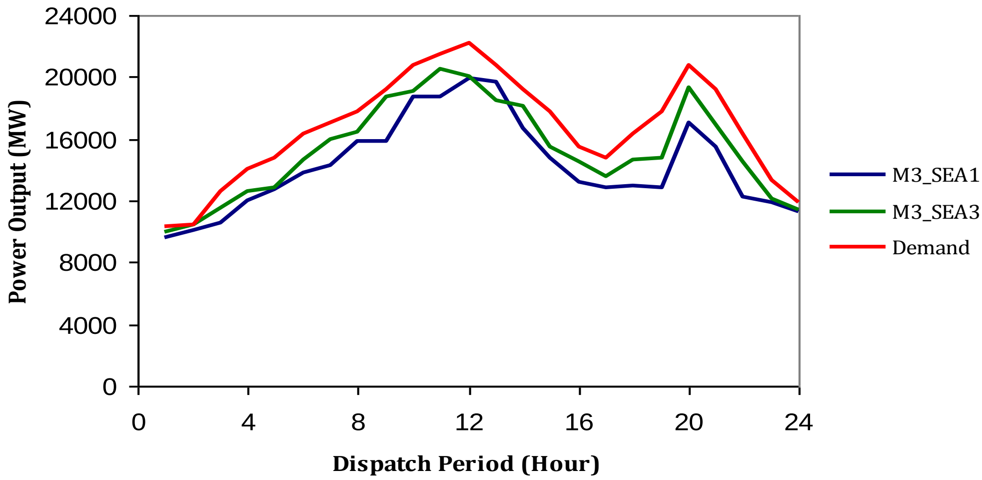

Figure 6.5: Load curve comparing the outputs of M3_SEA1 and M3_SEA3 with hourly load demands in the entire dispatch in the BBDELD for the Nigerian deregulated electricity market

Table 6.13 compares the total generation costs of M3_SEA1 and M3_SEA3 with other approaches in literature for the same set of data, as in dispatch period 2 (total load of 10,500MW). The best generation cost reported in literature until now for this load demand is $121,412.55/h [175], but much lower costs were realised using both M3_SEA1 and M3_SEA3 approaches.

Approach	*Min Cost ($/hr)*	*Avg Cost ($/hr)*	*Max Cost ($/hr)*
IFEP [167]	122,624.35	123,382.00	125,740.63
ABC [168]	121,432.39	121,995.82	122,123.77
SOH_PSO [169]	121,501.14	121,853.57	122,426.30
DE [45]	121,416.29	121,422.72	121,431.47
ICA_PSO [170]	121,413.20	121,428.14	121,453.56
BBO [171]	121,426.95	121,508.03	121,688.66
SI [172, 173]	121,415.59	121,615.85	-
MGSO [174]	121,412.57	-	-
SA [175]	121,412.55	121,418.05	121,425.28
M3_SEA 1	121,379.15	121,410.72	122,307.02
M3_SEA 3	121,315.60	121,339.30	121,669.75

Table 6.13: Comparison of generation costs for M3_SEA1 and M3_SEA3 with other approaches in dispatch period 2 of the same load demand (P_D=10,500MW)

6.6 Summary

This chapter applied various elements of the research presented in earlier chapters to the Nigerian electrical power system. It presented a chronological history of power generation in Nigeria, identifying the challenges to effective power provision which led to the various power sector reforms. We applied the three SEA approaches (SEA1, SEA2 and SEA3) to the country's power system before the sector reforms that preceded deregulation, a SELD problem, and demonstrated that these approaches show superior performance in comparison with Micro GA (μGA) and Conventional GA (CGA), two similar approaches identified in literature with the same set of data in terms of both generation cost and power loss. As seen from the resources allocation of Table 6.8, the total generation cost and total power loss of ₦99,143.50/hr and 9.45MW respectively, realised using SEA3, are the lowest and best obtainable from literature till date. This is in comparison with total generation costs of ₦102,940.36/hr, ₦103,264.07/hr, ₦107,540.00/hr, ₦110,640.07/hr respectively for SEA2, SEA1, μGA, CGA; and total power loss of 11.74MW, 11.46MW, 24.18MW, and 16.31MW respectively for SEA2, SEA1, μGA, CGA. Standard T-test (one-tailed) with significance level $p < 0.1$ shows 99.7% confidence that SEA3 is better than SEA1 in this problem, and also that SEA3 is superior to SEA2 with 97.5% confidence. There is insufficient information to compare statistically with μGA and CGA. On the overall, the results indicate that SEA3 is the best solution approach for this SELD problem.

From the good solutions obtained in the SELD problem, we adapted the BBDELD formulation and proposed a solution model to the country's deregulated electricity market using our smart EA, and demonstrated the ability of our SEA to deal with larger scale energy optimisation tasks in the bid based dispatch problems by means of a very high social profit, and superior performance over other reported approaches where the same set of data was used. We investigated the performance of two optimisation solution approaches that earlier provided superior results on related benchmark problems for the BBDELD in 12 trading periods, under the high bidding strategy (M3_SEA1 and M3_SEA3), and adapted them for a larger energy optimisation scenario in 24 dispatch periods. Optimised values for total generation costs, total customer benefits and social profits using the two approaches (Table 6.12 and Figure 6.3) show both approaches realising very high social profit, with M3_SEA3 slightly higher ($66,961,727.21 against $66,938,498.95 for M3_SEA1). From the distribution pattern of social profits across the 30 independent runs (Figure 6.4), standard T-test (one-tailed) with significance level $p < 0.1$, and load curve comparing the two approaches with the total hourly load demands across the entire dispatch period (Figure 6.5), we confirmed that M3_SEA3 is superior to M3_SEA1 with 99.9% confidence, and the best for the current large scale problem. Further exploration of the relative performance of the SEA was made by comparing the total generation costs of M3_SEA1 and M3_SEA3 with other approaches in literature for the same set of data, including the same static load demand in a common dispatch period (period 2, with total load demand, P_D of 10,500MW). The lowest/best generation cost reported in literature until now for this load demand is $121,412.55/h using Simulated Annealing (SA) approach [175], but much lower costs were realised using both M3_SEA1 and M3_SEA3 approaches - $121,315.60/h and $121,379.15/h respectively.

Chapter 7

Conclusion

7.1 Summary

Chapter 2 presented a background discussion of the research. The concept of optimisation was described, including the process, categories, cycle and its use as a problem solving tool. The chapter introduced the concept of electrical energy and electrical power, identifying power generation as the primary research domain. It highlighted the concept of optimal power flow, distinguishing between unit commitment and economic load dispatch (ELD) problems. In addition, it presented the various formulations of ELD problems, chronicled the historical developments in the solution approaches to ELD problems, and identified the two major types/categories of ELD problems – static ELD and dynamic ELD problems. The second part of the chapter presented a survey and study of evolutionary algorithm (EA), the optimisation tool used in the research. It described the design and implementation details of a standard EA, investigated and justified its suitability to solving ELD problems, reviewed recent and related work on power system optimisation. Lastly, it discussed the concepts of multi-objective EA, other non-evolutionary search algorithms and a review of smart mutation.

Chapter 3 described and evaluated a new EA for solving SELD problems. The formulation considered practical instances of SELD problems involving: minimum/maximum generation limits, power balance, ramp rates and prohibited operating zones constraints. In addition, work in the chapter involved various methods of handling constraints to keep the control variables in feasible regions where all constraints are satisfied. A technique for handling these constraints constitutes the "smartness" within the proposed EA approach to the SELD problem. The chapter described the features of a conventional EA used as a baseline approach, and gradually progressed to develop three versions of the smart mutation operator that are tailored for constrained ELD problems. It performed rigorous tuning of genetic parameters to select values for further experimental runs in the three versions of the smart mutation operator on three benchmark cases involving 6, 15 and 20 generating units; the ones commonly explored in recent literature, with a focus on the larger problem cases. Performance evaluation was made on the methods, and the results were compared with those reported for a range of recent alternative algorithms.

Chapter 4 described and evaluated a new EA for solving DELD problems, which extended the SELD formulation to a dynamic context. DELD is one of several optimisation problems that need repeatedly to be solved in the electricity sector. The EA developed in the SELD problems minimised total generation costs among the committed units satisfying all constraints. But in practical systems involving ramp-rate limit constraints, operational decisions at a given hour will affect the decision at a later hour. Due to the change in load conditions arising from these limits, the power generation has to be altered to meet the demand. DELD takes into consideration the dynamic costs involved in changing from one output level to another and is therefore a more applicable formulation of the ELD problem as it is faced by generating stations worldwide. The chapter showed the various steps involved in DELD formulation, and reviews recent literature of related work in the DELD problems. It described the DELD problems formulation involving dynamic constraints, and the computational algorithm used in the optimisation. From the results that provided superior results on the three versions of the smart mutation operator from SELD, the chapter adapted the solution algorithm for the dynamic case, and investigated three optimisation approaches. It compared the results of two problems test cases: systems of 5 and 10 generating units (involving a total of 18 instances of the algorithm – the three versions of the smart mutation operator in each of the three dynamic approaches, applied to the two test cases) with each other, and with identified methods from literature.

Chapter 5 investigated the use of EAs to maximise social profit in the context of BBDELD formulation in a deregulated electricity environment. It discussed the introduction of deregulation to the global electricity market, and identified price-based DELD and bid-based DELD as profit-based DELD formulations involved in optimal transaction dispatch in a deregulated market. The chapter reviewed related work in BBDELD and described the solution optimisation procedure, involving three bidding strategies – low, medium and high. It compared results of the best two versions of the smart mutation operator with previous results in the literature involving two test cases: 3 units, 2 customers in 2 dispatch periods and 6 units, 2 customers in 2 dispatch periods. Finally, the chapter defined and showed results on a new and larger test case – system of 10 generating units, 6 customers in 12 dispatch periods.

Chapter 6 adopted a case study approach, relating the work in previous chapters to the Nigerian electrical power system that has recently been deregulated and privatised. It presented the history of power generation in Nigeria and identified the challenges to effective power provision in generation, transmission and distribution of electricity that led to various reforms which took place in the power sector. It described the design and implementation of a constrained elitist genetic algorithm for the country's power system prior to the sector reforms, adapted the BBDELD formulation to the country's deregulated electricity market using the best two versions of the smart mutation operator from chapter 5, and investigated the ability of the SEA to deal with larger scale energy optimisation task in the bid based dispatch problems.

7.2 Contributions

The following summarise the main contributions of this thesis:

- Rather than use a generic/off-the-shelf EA-based optimisation package, we developed and demonstrated a novel approach to solving certain kinds of real-world problems. Here, we implemented a smart evolutionary algorithm (SEA), which combines a standard EA with a "smart mutation" operator that is tailored to the problem domain. This operator focused mutation on genes contributing most to cost and penalty violations in the fitness function. We investigated and analysed three distinct variants of the smart mutation operator. The pseudo-code of this algorithm was described in section 3.4, building on the smart mutation first described in section 2.9. We also demonstrated that this approach is successful on a range of problems in the electricity supply industry.
- We contributed a new approach to solving SELD problems. This optimisation approach was described in sections 3.2, 3.3 and 3.4, including how penalty violations were handled. In section 3.5, we demonstrated that this approach show superior performance to other previously published algorithms, on the basis of the common published test problems used in the literature on three benchmark cases involving 6, 15 and 20 generating units in terms of lowest cost and meeting load demand, with minimal power losses (see Tables 3.12, 3.22, 3.23, 3.30 and 3.31). In the 6-unit problem case, simulation results of SEA1, SEA2 and SEA3 on the basis of resources allocation in the best of 30 runs of the three approaches show that they are all better than other approaches in literature

with the same set of data in terms of both lower generation costs and lower power losses. The generation cost of $709.60/h from the resources allocation in the best of 30 runs of SEA1 (Table 3.12) is the best we have seen in the literature to date for this problem, in terms of lowest cost. In the 15-unit problem case, although SEA3 has the lowest average cost of $32,549.90/h, but a solution from SEA1 has the lowest minimum cost of $32,146.66/h. Besides having lower generation costs, all three approaches did not over produce power. Standard T-test (one tailed) with significance level $p < 0.1$ (confidence level 90%) shows no significant difference statistically between the three approaches for this problem, but the generation cost and power loss of $32,556/h and 0.05MW from the resources allocation in the best of 30 runs of SEA3 (Table 3.23) are the best we have seen in the literature to date for this problem, in terms of lowest cost and meeting load demand. In the 20-unit problem case, SEA1 has both average cost, but from a typical resources allocation of SEA3 has the lowest generation cost of $60483/h; and 0.0MW respectively, the best we have seen in the literature to date for this problem. No power was lost in SEA3, as it exactly generated the customers' load demand of 2500MW (Table 3.31). Standard T-test (one tailed) with significant level $p < 0.1$ show that SEA1 and SEA3 appear to be the best optimising approaches in this problem. We also investigated the overall performance of the three approaches by comparing their penalties handling capabilities. This contributes to the 'smartness' of the algorithms. SEA3 shows the most active performance in reducing penalty violations (see Figure 3.7), an evidence that it will perform well in larger scale test problems.

- We contributed a new approach to solving DELD problems, by taking the approaches that provided superior results in SELD problems and adapted them to solving DELD problems. The various steps involved in this optimisation approach, problem formulation and computational algorithm) are described in sections 4.2, 4.4 and 4.5 respectively. In the experimental design, we investigated three dynamic optimisation methods in conjunction with each of the three smart mutation variants, a total of 18 distinct experiments on benchmark cases involving 5 and 10 generating units (see Tables 4.3 and 4.8), the major test cases in the literature for which we have comparative results for other algorithms. Our results in section 4.6 suggest that the third method (M3), which exploits the dynamic nature of the problem, was capable of superior to the other two methods (M1and M2). Comparisons with all approaches so far in the literature that have addressed these problems show that these EA-based approaches, especially M3_SEA3 is superior to other algorithms (see Tables 4.4 and 4.9). In both test cases, our average and minimum best costs are better than those of the published approaches whenever the comparative figure is obtainable.

- We contributed a new approach to solving the BBDELD problem in a deregulated electricity market by building on the successes of the SEAs for SELD and DELD problems. We matched the bids submitted by GENCOs and their customers in each trading transaction in a multi-player/multi-period transaction to maximize social profit. We investigated the effects of three bidding strategies (low, medium and high) on the social profit, where we found, by experiments, that social profits are higher from matching the bids under the high bidding strategy (see Tables 5.8 and 5.9). The problem formulation and optimisation procedure, including the bidding strategies and computational algorithm are described in sections 5.4 and 5.5 (including the pseudo-code). In section 5.6, we demonstrated that our SEAs (based on dynamic method M3) show superior performances to other published algorithms on BBDELD on the basis of the common published test problems used in the literature – systems of 3 generators, 2 customers in 2 dispatch periods, and 6 generators, 2 customers in 2 dispatch periods (see Tables 5.4, 5.8). We also defined and showed results on a new and larger test case – a system of 10 generators, 6 customers in 12 dispatch periods (section 5.6.3), where we investigated the optimisation performance of M3_SEA1, M3_SEA2 and M3_SEA3. We generated the customers bid data (see Table 5.11) using the recommendations and specifications of the high bidding strategy from literature to meet a corresponding set of load across the entire 12-hour dispatch. The results (see Table 5.12 and Figures 5.7 and 5.8) show that the approaches (especially M3_SEA1 and M3_SEA3) are capable of large scale optimisations.

- We also contributed a new technique to solving the BBDELD in a case study involving the application of the bid-based SEAs to power sector reforms that led to the deregulation of the Nigerian electrical power industry. In section 6.4, we applied the three SEAs to solve the SELD problem of real power ELD for the Nigerian power system grid before the sector reforms. Here, we demonstrated that SEA1, SEA2 and SEA3 performed better than micro-GA and convectional

GA, two similar approaches from literature with common test problems in terms of lowest cost and power loss, with the best result from SEA3. In 6.5 we developed a novel solution model for the country's deregulated electricity market involving 40 generating power stations from 6 generating companies (GENCOs) and 11 customers (DISCOs) in a 24 trading (dispatch) periods. We generated the customers bid data (Tables 6.9 and 6.10) using the recommendations and specifications of the high bidding strategy from related published work in literature to meet the load demand requirements of the case study, with modifications to incorporate the larger generators bid data, where the total hourly load demands range from 10,360MW to 22,200MW depending on the time of the day. This is in line with the targeted generating capacity of 24,106MW for the Nigerian deregulated electricity market following the power sector reforms. We demonstrated the ability of both M3_SEA1 and M3_SEA3 to deal with larger scale energy optimisation task in the bid-based dispatch problems by means of a very high social profit (see Table 6.12 and Figures 6.3, 6.4 and 6.5), and superior performance over other reported approaches where the same set of data was used (see Table 6.13). Finally, from standard T-test (one-tailed) with significance level $p < 0.1$, we confirmed that M3_SEA3 is superior to M3_SEA1 with 99.9% confidence, and the best for the case study optimisation problem.

7.3 Future Work

The contributions in this thesis are partly in the area of applied computer science and partly in regard to a specific application area (electrical power system optimisation). Regarding applied computer science, we have shown how a specific approach to designing the mutation operator in an EA can advance performance on certain types of problem. Regarding specific application area (electrical power system optimisation), we have contributed methods that have advanced the state of the art in solving three specific optimisation problems under study in the area of Economic Load Dispatch. Areas of future work similarly align with these two areas of contribution. On the one hand, a number of related optimisation problems in economic load dispatch could be addressed with the smart mutation approach described in this thesis, and potentially provide further improvement in performance. On the other hand, there are many opportunities to further investigate variations in the general approach to smart mutation.

Among the many possibilities, we outline four specific areas of further work that may be fruitful in the first instance.

1. The power dispatch considered in this work is for thermal plants only (driven by heat released from burning of fossil fuels - coal, petroleum, natural gas). The power allocations to hydro units (where applicable) were fixed in-line with the utility's operating practices [53]. One potentially fruitful area of future work is to extend the approach to handle generating stations that use multiple fuel types. In such cases the quadratic relationships and constraints could be formulated in a very similar way to that of the SELD.

2. Generation of electricity from fossil fuel releases hydrocarbon emissions (gaseous pollutants - oxides nitrogen, sulphur and carbon dioxide) that pose environmental risks from thermal plants [27, 30, 31]. Practical generators' dispatch therefore requires a consideration of emission constraints. With the growing interests in this area, another possibility is to extend the formulation of the ELD problems to a true multi-objective optimisation with the addition of emission dispatch. In the literature surveyed, there were no sufficiently available data for comparison, especially for larger optimisation tasks considered in this work. Previous works were restricted to a maximum of 6 generators which is considered relatively too small in the present context. The different types of emissions could therefore be additional objectives.

3. Investigate more variants of the smart mutation operator. When using smart mutation, it is possible that in some problems, the chosen genes will cycle, i.e., mutating gene 1 moves the problem to gene 3. Mutating gene 3 in the next time moves the problem back to gene1, and so on. In the current work we focussed on performance in the application domain, and did not investigate this and related issues in depth. Future work could attempt a detailed investigation of this phenomenon, leading to designs for new, adaptive approaches to smart mutation that could avoid this situation. Regarding other aspects of the smart mutation approach, it seems a good idea in general to further investigate adaptive versions. We found, in this thesis that simple linear adaptation (linearly increasing the probability of a smart mutation from beginning to end of an optimisation run) seemed to outperform the use of smart mutation at a fixed rate. However, non-linear adaptive schedules could be explored, and it may also be sensible to explore schedules that adapt according to the total level of penalties

in the population. Further alternative adaptive approaches could include, for example, setting the smart mutation probability according only to the penalty value of the selected chromosome (perhaps normalised according to total penalties in the population).

4. In the DELD formulation, we only looked at three different approaches from a wide variety of possibilities:

   (i) Treating the DELD simply as a series of static problems,

   (ii) Treating the DELD as a single many-parameter problem, and

   (iii) A basic dynamic optimisation approach, in which the final population of one part of the DELD became the initial population of the next.

   At least, we showed that the general direction of using the final population of previous step seems promising. However, there are so many possibilities we could try here, such as using elitism. We keeping the best percentage of the population unchanged, and randomly generating the rest. Or, we could keep the best population, and generate the rest through mutations applied to these best populations. Also, in the context of a 24-hour DELD, at 4pm, we could use chromosomes from the previous population (e.g. 3pm Monday), or we could use the best chromosomes from 4pm Monday last week, etc.

Bibliography

[1] T. T. Nguyen, S. Yang and J. Branke, "Evolutionary Dynamic Optimisation: A Survey of the State of the Art," *Journal of Swarm and Evolutionary Computation*, vol. 6, pp. 1 – 24, Elsevier, 2012.

[2] D. P. Kothari, "Power System Optimisation," in *Proceeding of 2nd National Conference on Computational Intelligence and Signal Processing (CISP)*, Guwahati, Assam, pp. 18 – 21, 2012.

[3] A. Jager-Waldau and H. Ossenbrink, "Progress of Electricity from Biomass, Wind and Photovoltaics in the European Union," *Renewable and Sustainable Energy Reviews*, vol. 8, no. 2, pp. 157 – 182, 2004.

[4] G. Sreenivasan, C. H. Saibabu and S. Sivanagaraju, "Solution of Dynamic Economic Load Dispatch Problem with Valve Point Loading Effects and Ramp Rate Limits using PSO," *International Journal of Electrical and Computer Engineering*, vol. 1, no. 1, pp. 59 – 70, 2011.

[5] D. Bisen and H. M. Dubey, "Dynamic Economic Load Dispatch with Emission and Loss using GAMS," *International Journal of Engineering Research and Technology*, vol. 1, no. 3, pp. 1 – 7, 2012.

[6] S. Hemamalini and S. P. Simon, "Dynamic Economic Dispatch Problem with Valve Point Effects using Maclaurin Series Based Lagrangian Method," *International Journal of Computer Applications (0975-8887)*, vol. 1, no. 17, pp. 71 – 77, 2010.

[7] R. W. Ferrero and S. M. Shahidehpour, "Dynamic Economic Dispatch in Deregulated Systems," *Journal of Electrical Power and Energy Systems*, vol. 19, no. 7, pp. 433 – 439, 1997.

[8] K. Lakshmi and S. Vasantharathna, “Improved Genetic Algorithm for Constrained Generation Scheduling in Restructured Electricity Markets,” *Journal of Engineering Research and Studies*, vol. 1, no. 1, pp. 71 – 82, 2010.

[9] C. W. Ritchter and G. B. Sheble, “A Profit-Based Unit Commitment GA for the Competitive Environment,” *Transaction on Power Systems*, vol. 15, no. 2, pp. 715 – 721, IEEE, 2000.

[10] B. K. Pokharel, G. B. Shrestha, T. T. Lie and S. E. Fleten, “Profit-Based Unit Commitment in Competitive Markets,” in *Proceedings of International Conference on Power System Technology*, Singapore, 21st – 24th November, 2004.

[11] I. J. Raglend, C. Raghuveer, G. R. Avinash, N. P. Padhy and D. P. Kothari, “Solution to Profit-Based Unit Commitment Problem using Particle Swarm Optimisation,” *Journal of Applied Soft Computing*, vol. 10, pp. 1247 – 1256, Elsevier, 2010.

[12] T. Venkatesan and M. Y. Sanavullah, “SFLA Approach to Solve PBUC Problem with Emission Limitation,” *Journal of Electrical Power and Energy Systems*, vol. 46, pp. 1 – 9, Elsevier, 2013.

[13] C. C. Columbus and S. P. Simon, “Profit-Based Unit Commitment: A Parallel ABC Approach using a Workstation Cluster,” *Journal of Computers and Electrical Engineering*, vol. 38, pp. 724 – 745, Elsevier, 2012.

[14] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*. John Wiley, 2004.

[15] J. W. Chinneck, *Practical Optimisation: A Gentle Introduction*. [Online], Available at: www.sce.carleton.ca/faculty/chinneck/po.html, accessed: 05/01/2014.

[16] Z. Michalewicz and D. B. Fogel, How *to Solve it: Modern Heuristics*. Springer-Verlag, 2004.

[17] T. Hohm *et al*, “An Evolutionary Algorithm for the Block Stacking Problem,” in *N. Monmarch´e et al. (Eds.), EA 2007 LNCS*, vol. 4926, pp. 111-123, Springer-Verlag, 2008.

[18] C.C. Coello, “An Updated Survey of GA-Based Multi-Objective Optimization Techniques,” *ACM Computing Surveys*, vol. 32, no. 2, pp. 109-143, 2000.

[19] C. C. Coello, “Evolutionary Multi-Objective Optimization: A Historical View of the Field,” *IEEE Computational Intelligence Magazine*, pp. 28 – 36, IEEE Computational Intelligence Society, 2006.

[20] J. Arrillaga and C. P. Arnold, *Computer Analysis of Power Systems*. John Wiley, 1994.

[21] J. A. Momoh, *Electrical Power System: Applications of Optimisations*. Marcel Dekker, 2001.

[22] J. D. Glover and M. S. Sarma, *Power System Analysis and Design*. 3rd ed., Thomson Learning, 2002.

[23] J. Stewart, “Intermediate Electromagnetic Theory,” *World Scientific*, p. 50, 2001.

[24] C. O. Ahiakwo and S. Orike, “Distributed Generation (Renewable Energy): Best Option for Oil Bearing Communities,” *Journal of Sciences and Multidisciplinary Research*, vol. 2, pp. 9 – 14, 2010.

[25] Enerdata, *Global Energy Statistical Yearbook 2013* [Online]. Available at: http://yearbook.enerdata.net, accessed: 05/01/2014.

[26] R. C. Bansal, “Optimisation Methods for Electrical Power Systems: An Overview,” *International Journal for Emerging Electric Power Systems*, vol. 2, no. 1, pp. 1 – 23, 2005.

[27] M. A. Abido, “Multi-Objective Evolutionary Algorithms for Electric Power Dispatch Problem,” *Transaction on Evolutionary Computation*, vol. 10, no. 3, pp. 315 – 329, IEEE 2006.

[28] E. Zio, P. Baraldi and N. Pedroni, “Optimal Power System Generation Scheduling by Multi-Objective Genetic Algorithms with Preferences,” *Reliability Engineering and System Safety*, vol. 94, pp. 432 – 444, 2009.

[29] C. O. Ahiakwo and C. O. Igwe, “Loss Minimisation in Optimal Power Flow”. *European Journal of Scientific Research*, vol. 13, no. 4, pp. 381 – 387, 2006.

[30] A. L. Devi and O. V. Krishna, “Combined Economic and Emission Dispatch using Evolutionary Algorithms – A Case Study,” *ARPN Journal of Engineering and Applied Science*, vol. 3, no. 6, pp. 28 – 34, 2008.

[31] J. Dhillon and S. K. Jain, “Multi-Objective Generation and Emission Dispatch using NSGA-II,”. *IACSIT International Journal of Engineering and Technology*, vol. 3, No. 5, pp. 460 – 466, 2011.

[32] G. L. Gaing, “Particle Swarm Optimisation to Solving the Economic Dispatch Considering the Generator Constraints,” *Transactions on Power Systems*, vol. 18, no. 3, pp.1187 – 1195, IEEE, 2003.

[33] L. S. Coelho and C. Lee, “Solving Economic Load Dispatch Problems in Power Systems using Chaotic and Gaussian Particle Swarm Optimisation Approaches,” *International Journal of Electrical Power and Energy Systems*, vol. 30, no. 5, pp. 297 – 307, Elsevier, 2008.

[34] S. Orike and D. W. Corne, “Improved Evolutionary Algorithms for Economic Load Dispatch Optimisation Problems,” in *Proceedings of 12th UK Workshop on Computational Intelligence (UKCI)*, Edinburgh, IEEE, 2012.

[35] M. Vanitha and K. Thanushkodi, “Solution to Economic Dispatch Problem by Differential Evolution Algorithm Considering Linear Equality and Inequality

Constraints," *International Journal of Research and Reviews in Electrical and Computer Engineering*, vol. 1, no. 1, pp. 21 – 26, 2011.

[36] T. Bouktir, L. Slimani and M. Belkacemi, "A Genetic Algorithm for Solving the Optimal Power Flow Problem," *Leonardo Journal of Sciences*, vol. 4, pp. 44 – 58, 2004.

[37] M. S. Osman, M. A. Abo-Sinna and A. A. Mousa, "A Solution to the Optimal Power Flow using Genetic Algorithm," *Applied Mathematics and Computation*, vol. 155, pp. 391 – 405, 2004.

[38] A. S. Uyar and B. Turkay, "Evolutionary Algorithms for the Unit Commitment Problem," *Turk Journal of Electrical Engineering*, vol.16, no.3, pp. 239 – 255, Tubitak, 2008.

[39] K. Abookazemi, M. W. Mustafa and H. Ahmad, "Structured Genetic Algorithm Technique for Unit Commitment Problem," *International Journal of Recent Trends in Engineering*, vol. 1, no. 3, pp. 135 – 139, 2009.

[40] C. Kumar and T. Alwarsamy, "Dynamic Economic Dispatch – A Review of Solution Methodologies," *European Journal of Scientific Research*, vol. 64, no. 4, pp. 517 – 537, 2011.

[41] D. Srinivasan and J. Chazelas, "A Priority List-Based Evolutionary Algorithm to Solve Large Scale Unit Commitment Problem," in *Proceedings of 2004 International Conference on Power System Technology (PowerCon 2004)*, vol. 2, pp. 1746 – 1751, 2004.

[42] L. K. Kirchmayer, *Economic Operation of Power Systems*, John Wiley, 1958.

[43] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation and Control*. 2nd Ed., John Wiley, 2012.

[44] C. L. Chiang, "Improved Genetic Algorithm for Power Economic Dispatch of Units with Valve-Point Effects and Multiple Fuels," *Transactions on Power Systems*, vol. 20, no. 4, pp. 1690 – 1699, IEEE, 2005.

[45] N. Noman and H. Iba, "Differential Evolution for Economic Load Dispatch Problem," *Journal of Electric Power Systems Research*, vol. 78, no. 3, pp. 1322 – 1331, 2008.

[46] A. A. El-Fergany, "Solution of Economic Load Dispatch Problem with Smooth and Non-Smooth Fuel Cost Functions including Line Losses using Genetic Algorithm," *International Journal of Computer and Electrical Engineering*, vol. 3, no. 5, pp. 706 – 710, 2011.

[47] M. Pourakbari-Kasmaei and N. Rashidi-Nejad, "An Effortless Hybrid Method to Solve Economic Load Dispatch Problem in Power Systems," *Journal of Energy Conversion and Management*, vol. 52, pp. 2854 – 2860, 2011.

[48] J. H. Holland, *Adaptation in Natural and Artificial Systems.* University of Michigan Press, 1975.

[49] U. C. Chukwu, C. O. Ahiakwo and M. A. Nanim, "Solving Power System problem using Fast-Decoupled Algorithm," *European Journal of Scientific Research*, vol. 17, no. 2, pp. 160 – 172, 2007.

[50] H.A. Smolleck, "A Classroom Method for Structuring the Bus Admittance Matrix from Synthesis of Coupling-free Equivalents," *Transactions on Power Systems*, vol. 7, no. 1, pp. 383 – 388, IEEE, 1992.

[51] X. Xia and A. M. Elaiw, "Optimal Dynamic Economic Dispatch of Generation: A Review," *Journal of Electrical Power Systems Research*, vol. 80, pp. 975 – 986, Elsevier, 2010.

[52] A. U. M. Abdulaziz and H. I. Alhabib, "Power Network Planning using Mixed-Integer Programming," *JKAU: Eng. Sci.*, vol. 21, no. 2, pp. 15 – 34, 2010.

[53] G. A. Bakare, U. O. Aliyu, G. K. Venayagamoorthy and Y. K., Shu'aibu, "Genetic Algorithm Based Economic Dispatch with Application to Coordination of Nigerian Thermal Power Plants," in *Power Engineering Society General Meeting*, vol. 1, pp. 551 – 556, IEEE, 2005.

[54] S. Sayah and K. Zehar, "Using Evolutionary Computation to Solve the Economic Load Dispatch Problem," Leonardo Journal of Sciences, vol. 12, pp. 67 – 78, 2008.

[55] B. Shaw, S. Ghoshal, V. Mukherjee and S. P. Ghoshal, "Solution of Economic Load Dispatch Problems by a Novel Seeker Optimisation Algorithm," *International Journal on Electrical Engineering and Informatics,* vol. 3, no. 1, pp. 26 – 42, 2011.

[56] C. T. Su and C. T. Lin, "New Approach with a Hopfield Modelling Framework to Economic Dispatch," *Transactions on Power Systems*, vol. 15, no. 2, pp. 541 – 545, IEEE, 2000.

[57] R. Haque and N. Chowdhury, "An Artificial Neural Network Based Transmission Loss Allocation for Bilateral Contracts," in *Proceedings of the 18th Annual Canadian Conference on Electrical and Computer Engineering*, pp. 2197 – 2201, 2005.

[58] S. Pothiya, I. Ngamroo and W. Kongprawechnon, "Application of Multiple Tabu Search Algorithm to solve Dynamic Economic Dispatch considering Generator Constraints," *Journal of Energy Conversion and Management*, vol. 49, no. 4, pp. 506 – 516, 2007.

[59] M. R. Narimani, "A New Modified Shuffle Frog Leaping Algorithm for Non-Smooth Economic Dispatch," *World Applied Science Journal*, vol. 12, no. 6, pp. 805 – 814, 2011.

[60] C. Chokpanyasuwan, S. Anantasate, S. Pothiya, W. Pattaraprakorn and P. Bhasaputra, "Honey Bee Colony Optimisation to Solve Economic Dispatch Problem with Generator Constraints," in *International Conference on*

Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), vol. 1, pp. 200 – 203, IEEE, 2009.

[61] R. Behera, B. B. Pati and B. P. Panigrahi, “Economic Power Dispatch using Artificial Immune System,” in *Proceedings of 16th National Power Systems Conference*, pp. 664 – 668, 2010.

[62] T. K. A. Rahman, S. I. Suliman and I. Musirin, “Artificial Immune-Based Optimisation Technique for Solving Economic Dispatch in Power System,” in *B. Apolloni et al. (Eds.): LNCS* vol. 3931, pp. 338 – 345, Springer-Verlag, 2006.

[63] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison Wesley, 1989.

[64] C. K. Panigrahi, P. K. Chattopadhyay, R. N. Chakrabarti, M. Basu, “Simulated Annealing Technique for Dynamic Economic Dispatch,” *Journal of Electrical Power Components Systems Research*, vol. 34, no. 5, pp. 577 – 586, 2006.

[65] A. Pereira-Neto, C. Unsihuay and O. R. Saavedra, “Efficient Evolutionary Strategy Optimisation Procedure to solve the Non-convex Economic Dispatch Problem with Generator Constraint,” in *Proceedings on Generation, Transmission and Distribution*, vol. 152, no. 5, pp. 653 – 660, IEEE, 2005.

[66] S. K. Shakya, “Probabilistic Model Building Genetic Algorithm: A Survey,” *Technical Report*, The Robert Gordon University, Aberdeen, 2003.

[67] R. Storn and K. V. Price, “Differential Evolution, a Simple and Efficient Heuristic for Global Optimisation over Continuous Spaces,” *Journal of Global Optimisation*, vol. 11, vo. 4, pp. 341 – 359, 1997.

[68] J. McCall, “Genetic Algorithms for Modelling and Optimisation,” Journal of Computational and Applied Mathematics, vol. 184, pp. 205 – 222, Elsevier, 2005.

[69] Y. Yen and R. Langari, *Fuzzy Logic: Intelligence, Control, and Information*, Prentice Hall, 1999.

[70] M. Dorigo and T. Stützle, *Ant Colony Optimization*. MIT Press, 2004.

[71] J. Kennedy and R. Eberhart, *Swarm Intelligence.* Morgan Kaufmann, 2001.

[72] D. Karaboga, "An Idea Based on Honey Bee Swarm for Numerical Optimisation," Technical Report, Erciyes University, Turkey, 2005.

[73] D. Saxena, S. N. Singh and K. S. Verma, "Application of Computational Intelligence in Emerging Power Systems," *International Journal of Engineering, Science and Technology*, vol. 2, no. 3, pp 1 – 7, 2010.

[74] R. C. Bansal, "Literature Survey on Expert System Applications to Power Systems," *International Journal Engineering Intelligent Systems,* vol. 11, no. 3, pp.103 – 112, 2003.

[75] J. D. Farmer, N. Packard and A. Perelson, "The Immune System, Adaptation and Machine Learning," *Physica D*, vol. 2, pp. 187–204, 1986.

[76] M. Sudhakaran and P. A. Raj, "Integrating Genetic Algorithms and Tabu Search for Unit Commitment Problem," *International Journal of Engineering, Science and Technology*, vol. 2, no. 1, pp. 57 – 69, MultiCraft, 2010.

[77] J. R. Koza, "Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems," Technical Report (STAN-CS-90-1314), Department of Computer Science, Stanford University, 1990.

[78] D. He, F. Wang and Z. Mao, "A Hybrid Genetic Algorithm Approach based on Differential Evolution for Economic Dispatch with Valve-Point Effect," *Journal of Electrical Power and Energy Systems*, vol. 30, pp. 31 – 38, 2008.

[79] C. C. Kuo, "A Novel Coding Scheme for Practical Economic Dispatch by Modified Particle Swarm Approach," *Transactions on Power Systems*, vol. 23, no. 4, pp.1825 – 1835, IEEE, 2008.

[80] J. S. Alsumait, J. K. Sykulski and A. K. Al-Othman, "A Hybrid GA-PS-SQP Method to Solve Power System Valve-Point Economic Dispatch Problems," *Journal of Applied Energy*, vol. 87, pp. 1773 – 1781, Elsevier, 2010.

[81] D. Poole, A. Mackworth and R. Goebel, *Computational Intelligence, a Logical Approach*, Oxford University Press, 1999.

[82] A. P. Engelbrecht, *Computational Intelligence, an Introduction*. John Wiley, 2003.

[83] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, 1998.

[84] J. E. Baker, "Adaptive Selection Methods for Genetic Algorithms," in *Proceedings of the First International Conference on Genetic Algorithms and their Applications*, pp. 101 – 111, 1985.

[85] M. Sipper and H. A. Simon, "On the Origin of Environments by Means of Natural Selection,". *AI Magazine*, vol. 22, no. 4, pp. 133 – 140, 2001.

[86] A. Rangel-Merino, J. L. López-Bonilla, and R. Linares y Miranda, "Optimisation Method based on Genetic Algorithms," *Apeiron*, vol. 12, no. 4, pp. 393 – 408, Roy Keys Inc, 2005.

[87] J. Cogley, "Designing, Implementing and Optimising an Object-Oriented Chess System using a Genetic Algorithm in Java and its Critical Evaluation," *Research Report*, The Open University, 2001.

[88] E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multi-Objective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173 – 195, 2000.

[89] A. Konak, D. W. Coit and A. E. Smith, "Multi-Objective Optimization using Genetic Algorithms: A Tutorial," *Reliability Engineering and System Safety*, vol. 91, no. 9, pp. 992 – 1007, Elsevier, 2006.

[90] C. C. Coello, "Evolutionary Multi-Objective Optimization: A Historical View of the Field," *Computational Intelligence Magazine*, vol. 1, no. 1, pp. 28 – 36, IEEE, 2006.

[91] A. A. Freitas, "A Critical Review of Multi-Objective Optimization in Data Mining," in *ACM SIGKDD Explorations*, vol. 6, no. 2, pp. 77 – 86, 2005.

[92] P. J. Fleming, "Introduction to Multi-Objective Optimisation: Automatic Control and Systems Engineering," *Lecture Notes*, The University of Sheffield, 2014.

[93] A. Marczyk, "Genetic Algorithms and Evolutionary Computation," in *TalkOrigins Archive, Exploring the Creation/Evolution Controversy, 2004,* [Online] Available: http://www.talkorigins.org/faqs/genalg/genalg.html, Retrieved: 05/01/2014.

[94] S. Sayah, K. Zehar and N. Bellaouel, "A Successive Linear Programming based method for solving Optimal Power Flow Problems," in *Proceedings of the 1st International Meeting on Electronics and Electrical Science and Engineering*, University of Djelfa, Algeria, 2006.

[95] T. T. Nguyen, X. Yao, "Dynamic Time-Linkage Problems Revisited," in *LNCS: Applications of Evolutionary Computing*, vol. 5484, pp. 735 – 744, Springer, 2009.

[96] S. Ganesan and S. Subramanian, "Dynamic Economic Dispatch Based on Simple Algorithm," *International Journal of Computer and Electrical Engineering*, vol. 3, no. 2, pp. 1793 – 8163, 2011.

[97] P. Attaviriyanupap, H. Kita, E. Tanaka and J. Hasegawa, "A Fuzzy-Optimisation Approach to Dynamic Economic Dispatch Considering Uncertainties," *Transactions of Power Systems*, vol. 19, no. 3, pp. 1299 – 1307, IEEE, 2004.

[98] M. Basu, “Dynamic Economic Emission Dispatch using Non-dominated Sorting Genetic Algorithm-II,” *Journal of Electrical Power Energy Systems*, vol. 30, no. 2, pp. 140 – 149, 2008.

[99] K. Asano, M. Nakatsuka and T. Kumano, “Calculus of Variation and Genetic Algorithm considering Ramp Rate,” in *Proceedings of 15th International Conference on Intelligent System Applications to Power Systems*, pp. 1 – 6, Curitiba, 2009.

[100] S. F. Mekhamer, A. Y. Abdelaziz, M. Z. Kamh and M. A. L. Badr, “Dynamic Economic Dispatch using a Hybrid Hopfield Neural Network/Quadratic Programming Based Technique,” *Electrical Power Components and Systems*, vol. 37, no. 3, pp. 253 – 264, 2009.

[101] V. R. Pandi and B. K. Panigrahi, “Dynamic Economic Load Dispatch using Hybrid Swarm Intelligence Based Harmony Search Algorithm,” *Journal of Expert Systems with Applications*, vol. 38, pp. 8509 – 8514, 2011.

[102] T. Niknam and F. Golestaneh, “Enhanced Adaptive Particle Swarm Optimisation Approach for Dynamic Economic Dispatch of Units Considering Valve-Point Effects and Ramp Rates,” *IET Generation, Transmission and Distribution*, vol. 6, no. 5, pp. 424 – 435, 2002.

[103] Z. L. Gaing, “Constrained Dynamic Economic Dispatch Solution using Particle Swarm Optimisation,” in *Proceedings of IEEE Power Engineering Society General Meeting*, vol. 1, pp. 153 – 158, 2004.

[104] T. A. A. Victoire. and A. E. Jeyakumar, “Deterministically Guided PSO for Dynamic Dispatch considering valve-point effect,” *Electrical Power System Research*, vol. 73, no. 3, pp. 313-322, 2005.

[105] B. K. Panigrahi, V. R. Pandi and S. Das, “Adaptive Particle Swarm Optimisation Approach for Static and Dynamic Economic Load Dispatch,” *Journal of Energy Conversion and Management*, vol. 49, pp. 1407 – 1415, Elsevier, 2008.

[106] H. Shayeghi and A. Ghasemi, "Application of MOPSO for Economic Load Dispatch Solution with Transmission Losses," *International Journal on Technical and Physical Problems of Engineering*, vol. 4, no. 1, pp. 27 – 34, 2012.

[107] P. Attaviriyanupap H. Kita, E. Tanaka and J. Hasegawa, "A Hybrid EP and SQP for Dynamic Economic Dispatch with Nonsmooth Fuel Cost Function," *Transactions on Power Systems*, vol. 17, no. 2, pp. 411 – 416, IEEE, 2002.

[108] R. Balamurugan and S. Subramanian, "Differential Evolution-based Dynamic Economic Dispatch of Generating Units with Valve-point Effects," *Electrical Power Components and Systems*, vol. 36, no. 8, pp. 828 – 843, 2008.

[109] D. He, G. Dong, F. Wang, and Z. Mao, "Optimisation of Dynamic Economic Dispatch with Valve-Point Effect using Chaotic Sequence based Differential Evolution Algorithms," *Journal of Energy Conversion and Management*, vol. 52, no. 2, pp. 1026 – 1032. Elsevier, 2011.

[110] M. Basu, "An Interactive Fuzzy Satisfying-based Simulated Annealing Technique for Economic Emission Load Dispatch with Nonsmooth Fuel Cost and Emission Level Functions," *Electrical Power Components and Systems*, vol. 32, no. 2, pp. 163 – 173, 2010.

[111] U. K. Rout, R. K. Swain, A. K. Barisal and R. C. Prusty, "Clonal Selection Algorithm for Dynamic Economic Dispatch with Nonsmooth Cost Functions," *International Journal of Scientific and Engineering Research*, vol. 2, no. 12, pp. 1 – 5, 2011.

[112] H. Rudnick, R. Varela and W. Hogan, "Evaluation of Alternatives for Power System Coordination and Pooling in a Competitive Environment," *Transactions on Power Systems*, vol. 12, no. 2, pp. 605 – 613, IEEE, 1997.

[113] A. Al-Sunaidy and R. Green, "Electricity Deregulation in OECD (Organization for Economic Cooperation and Development) Countries," *Journal of Energy*, vol. 31, pp. 769 – 787, Elsevier, 2006.

[114] H. Y. Yamin, "Security-Constrained Price-Based Unit Commitment in the Deregulated Power Market," in *Proceedings of the Large Engineering Systems Conference on Power Engineering*, pp. 18 – 22, 2002.

[115] T. Li and M. Shahidehpour, "A Price-Based Unit Commitment: A Case of Lagrangian Relaxation versus Mixed Integer Programming," *Transactions on Power Systems*, vol. 20, no. 4, pp. 2015 – 2025, IEEE, 2005.

[116] G. B. Shrestha, B. K. Pokharel, T. T. Lie and S. E. Fleten, "Price-Based Unit Commitment for Bidding under Price Uncertainty," *IET Generation, Transmission and Distribution*, vol. 1, no. 4, pp. 663 – 669, 2007.

[117] D. K. Dimitroulas and P. S. Georgilakis, "A New Memetic Algorithm Approach for the Price-Based Unit Commitment Problem," *Journal of Applied Energy*, vol. 88, pp. 4687 – 4699, Elsevier, 2011.

[118] R. Leou, "A Price-Based Unit Commitment Model considering uncertainties with a Fuzzy Regression Model," *International Journal of Energy Science*, vol. 2, no. 2, pp. 51 – 58, 2012.

[119] G. C. Liao, "Bid-Based Economic Electrical Load Dispatch using Improved Genetic Algorithm," in *Proceedings of 8th Asian Control Conference*, pp. 1387 – 1392, 2011.

[120] C. W. Richter and G. B. Sheble, "Genetic Algorithm Evolution of Utility Bidding Strategies for the Competitive Marketplace," *Transactions on Power Systems*, vol. 13, no. 1, pp. 256 – 261, IEEE, 1998.

[121] W. M. Lin and S. J. Chen, "Bid-Based Dynamic Economic Dispatch with an Efficient Interior Point Algorithm," *Journal of Electrical Power and Energy Systems*, vol. 24, pp. 51 – 57, Elsevie, 2002.

[122] B. Zhao, C. Guo and Y. Cao, "Dynamic Economic Dispatch in Electricity Market using Particle Swarm Optimisation Algorithm," in *Proceedings of the 5th World Congress on Intelligent Control and Automation*, pp. 5050 – 5054, China, 2004.

[123] G. C. Liao and J. C. Lee, "Application Novel Immune Genetic Algorithm for Solving Bid-Based Dynamic Economic Power Load Dispatch," *International Conference on Power System Technology*, pp. 1 – 7, IEEE, 2010.

[124] C. K. Panigrahi, P. K. Chattopadhyay and U. Prasad, "Linear Programming Approach for Bidding of Generators in Restructured Power Industry," *International Electrical Engineering Journal*, vol. 3, no. 1, pp. 589 – 594, 2012.

[125] D. W. Ross and S. Kim, "Dynamic Economic Dispatch of Generation," *Transactions on Power Apparatus and Systems*, vol. 99, no. 6, pp. 2060 – 2068, IEEE, 1980.

[126] R. R. Shoults, S. V. Venkatesh, S. D. Helmick, G. L. Ward and M. J. Lollar, "A Dynamic Programming Based Method for Developing Dispatch Curves when Incremental Heat Rate Curves are Non-Monotonically Increasing," *Transactions on Power Systems*, vol. 1, no. 1, pp. 10 – 16, IEEE, 1886.

[127] Z. X. Liang and J. D. Glover, "A Zoom Feature for a Dynamic Programming Solution to Dispatch including Transmission Losses," *Transactions on Power Systems*, vol. 7, no. 2, pp. 544 – 550, IEEE, 1992.

[128] F. Zhuang and F. D. Galiana, "Unit Commitment by Simulated Annealing," *Transactions on Power Systems*, vol. 5, no. 1, pp. 311 – 318, IEEE, 1990.

[129] K. P. Wong and C. C. Fung, "Simulated Annealing Based Economic Dispatch Algorithm," in *Proceedings of Power Generation, Transmission and Distribution*, vol. 140, no. 6, IEEE, 1993.

[130] A. H. Mantawy, Y. L. Abdel-Magid and S. Z. Selim, "A Simulated Annealing Algorithm for Unit Commitment," Transactions on Power Systems, vol. 13, no. 1, pp. 197 – 204, IEEE, 1998.

[131] A. J. Gaul, E. Handschin, W. Hoffmann and C. Lehmköster, "Establishing a Rule Base for a Hybrid ES/XPS Approach to Load Management," *Transactions on Power Systems*, vol. 13, no. 1, pp. 86 – 93, IEEE, 1998.

[132] P. H. Chen and H. C. Chang, "Large-Scale Economic Dispatch by Genetic Algorithm," *Transactions on Power Systems*, vol. 10, no. 4, pp. 1919 – 1926, IEEE, 1995.

[133] S. A. Kazarlis, A. G. Bakirtzis and V. Petridis, "A Genetic Algorithm Solution to the Unit Commitment Problem," *Transactions on Power Systems*, vol. 11, no. 1, pp. 83 – 92, IEEE, 1996.

[134] T. T. Maifield and G. B. Sheblé, "Genetic-Based Unit Commitment Algorithm," *Transactions on Power Systems*, vol. 11, no. 3, pp. 1359 – 1370, IEEE, 1996.

[135] S. H. Hosseini and M. Kheradmandi, "Dynamic Economic Dispatch in Restructured Power Systems Considering Transmission Cost using Genetic Algorithm," in *Proceedings of Canadian Conference on Electrical and Computer Engineering*, vol. 3, pp. 1625 – 1628, IEEE, 2004.

[136] X. Ma and Y. Liu, "Dynamic Load Economic Dispatch in Electricity Market Using Improved Particle Swarm Optimisation Algorithm," in Proceedings of International Conference on Intelligent Computation Technology and Automation, vol. 2, pp. 165 – 168, IEEE, 2010.

[137] B. Rampriya, K. Mahadevan and S. Kannan, "Application of Differential Evolution to Dynamic Economic Dispatch Problem with Transmission Losses under various Bidding Strategies in Electricity Markets," *Journal of Electrical Engineering and Technology*, vol. 7, no. 5, pp. 681 – 688, 2012.

[138] M. Sniedovich, “A New Look at Bellman’s Principle of Optimality,” *Journal of Optimisation Theory and Application*, vol. 49, no. 1, pp. 161 – 176, 1986.

[139] A. K. David, “Competitive Bidding in Electricity Supply,” in *Proceedings of Generation, Transmission and Distrubution*, vol. 140, no. 3, pp. 421 – 426, IEEE, 1999.

[140] A. K. David and F. Wen, “Strategic Bidding in Competitive Electricity Markets: A Literature Survey,” in *Proceedings of Summer Power Meetings*, vol. 4, Seattle, USA, pp. 2168 – 2173, IEEE, 2000.

[141] A. Botterud, P. R. Thimmapuram and M. Yamakado. Simulating GENCO Bidding Strategies in Electricity Markets with an Agent-based Model. In *Proceedings of the 7th Annual IAEE European Energy Conference*, 2005.

[142] D. Zhang, Y. Wang and P. B. Luh, “Optimisation Based Bidding Strategies in the Deregulated Market,” *Transactions on Power Systems*, vol. 15, no. 3, pp. 981 – 986, IEEE, 2000.

[143] *List of Countries in the World by Population*, Available: http://en.wikipedia.org/wiki/List_of_countries_by_population, accessed: 14/01/2015.

[144] *CIA World Factbook*, Available: https://www.cia.gov/library/publications/the-world-factbook/, accessed: 14/01/2015.

[145] N. C. Maduekwe, “Unbundling and Privatisation of the Nigerian Electricity Sector: Reality or Myth?,” *Energy Centre*, University of Dundee, 2011.

[146] S. Fajana, “Strengthening Social Dialogue in the Utilities Sector in Nigeria,” *Publications of the International Labour Office*, Paper no. 272, Geneva, 2010.

[147] A. S. Sambo, “Matching Electricity Supply with Demand in Nigeria,” *International Association for Energy Economics*, Fourth Quarter, pp. 32 – 36, 2008.

[148] A. A. Akinwale, “The Menace of Inadequate Infrastructure in Nigeria,” *African Journal of Science, Technology, Innovation and Develepment*, vol. 2, no. 3, pp. 207 – 228, 2010.

[149] O. I. Okoro and E. Chikuni, “Power Sector Reforms in Nigeria: Opportunities and Chanllenges,” *Journal of Energy in South Africa*, vol. 18, no. 3, pp. 52 – 57, 2007.

[150] O. I. Okoro, E. Chikuni, P. O. Oluseyi and P. Govender, “Conventional Energy Sources in Nigeria: A Statistical Approach,” Available at: http://active.cput.ac.za/energy/past_papers/DUE/2008/PDF/Paper-OkoroO.pdf, Accessed: 06/01/2014.

[151] *World’s Electricity Consumption Per Capita (in kWh)*, Available: http://www.indexmundi.com/map/?v=81000, Accessed: 06/01/2014.

[152] *World Bank Data Indicators: Electric Power Consumption (in kWh)*, Available: http://data.worldbank.org/indicator/EG.USE.ELEC.KH.PC, Accessed: 06/01/2014.

[153] *Electricity Consumption Per Capita (in kWh) 2013 Country Ranks*, Available: http://www.photius.com/rankings/energy/electricity_consumption_per_capita_2013_0.html, Accessed: 06/01/2014.

[154] Felix Ayanruoh, “The Challenges of the Nigerian Electric Power Sector Reform,” in *Energy, Vanguard Newspaper*, February 26, 2013, available: http://www.vanguardngr.com/2013/02/the-challenges-of-the-nigerian-electric-power-sector-reform-1/, accessed: 06/01/2014.

[155] D. O. Adeyemo and A. Salami, “A Review of Privatisation and Public Enterprises Reform in Nigeria,” *Contemporary Management Research*, vol. 4, no. 4, pp. 401 – 418, 2008.

[156] *Public Enterprises (Privatisation and Commercialisation) Decree 1999*, Available: http://www.nlii.org/files/privatisation.html.pdf, Accessed: 08/01/2014

[157] EPSRA, "Electric Power SectorReform Act No. 6," *Federal Republic of Nigeria Official Gazette*, vol. 92, no. 77, pp. 77 – 130, 2005.

[158] *Roadmap for Power Sector Reform (a Customer-driven Sector-wide plan to achieve stable Power Supply)*, The Presidency, Federal Republic of Nigeria, 2010.

[159] K. O. Titus, A. J. Abdul-Ganiyu and D. A. Phillips, "The Curent and Future Challenges of Electricity Market in Nigeria in the face of Deregulation Process," *African Journal of Engineering Research*, vol. 1, no. 2, pp. 33 – 39, 2013.

[160] T. Yalcinoz and M. J. Short, "Large Scale Economic Dispatch using an Improved Hopfield Neural Network," in *Proceedings of Power Generation, Transmission and Distribution*, vol. 144, no. 2, pp. 181 – 185, IEEE, 1997.

[161] K. Krishnakumar, "Micro Genetic Algorithms for Stationary and Non-stationary Function Optimisation," in *Proceedings of SPIE Intelligent Control and Adaptive System*, vol. 1196, pp. 289 -296, 1989.

[162] S. O. Onohaebi and S. O. Igbinovia, "Development of A MATLAB Program to Compute Bus Voltages in the Nigeria Electrical Power System," *International Journal of Academic Research*, vol. 2, no. 4, pp. 125 – 132, 2010.

[163] S. O. Onohaebi, "Reduction of the High Technical Power Losses associated with the Nigerian 330kV Transmission Network," *International Journal of Electrica Power Engineering*, vol. 1, no. 4, pp. 421 – 431, 2007.

[164] Power World Co-operation, *Power World Simulator*, version 8 (1996 – 2000).

[165] E. C. Chuka, U. Nwuba and M. C. Ogonna, "Optimum Reliability and Cost of Power Distribution System: A Case of Power Holding Company of Nigeria,"

International Journal of Engineering Science and Technology, vol. 3, no. 8, pp. 6671 – 6683, 2011.

[166] A. S. Sambo, B. Garba, I. H. Zarma and M. M. Gaji, “Electricity Generation and the Present Challenges in the Nigerian Power Sector,” *Energy Commission of Nigeria*, available: http://89.206.150.89/documents/congresspapers/70.pdf, accessed: 06/01/2014.

[167] N. Sinha, R. Chakrabarti and P. K. Chattopadhyay, “Evolutionary Programming Techniques for Economic Load Dispatch,” *Transactions on Evolutionary Computation*, vol. 7, no. 1, IEEE, 2003.

[168] S. Hemamalini and S. P. Simon, “Economic Load Dispatch with valve point effect using Artificial Bee Colony Algorithm,” in *32nd National System Conference*, pp.525 – 530, 2008.

[169] K. T. Chaturvedi, M. Pandit and L. Srivastava, “Self-Organizing Hierarchical Particle Swarm Optimisation for Non-Convex Economic Dispatch,” *Transactions of Power System,* vol. 23, no. 3, pp. 1079 – 1087, IEEE, 2008.

[170] J. G.Vlachogiannis and K.Y. Lee, “Economic Load Dispatch – A Comparative Study on Heuristic Optimisation Techniques with an Improved Coordinated Aggregation-based PSO. *Transactions of Power System,* vol. 24, no. 2, pp. 991 – 1001, IEEE, 2009.

[171] A. Bhattacharya and P.K. Chattopadhyay, “Biogeography-Based Optimisation for Different Economic Load Dispatch Problems,” *Transactions of Power System,* vol. 25, no. 2, pp. 1064 – 1077, IEEE, 2010.

[172] V. R. Pandi, B. K. Panigrahi, R.C. Bansal, S. Das and A. Mohapatra, “Economic Load Dispatch using hybrid Swarm Intelligence-Based Harmony Search Algorithm, *Electric Power Components and Systems,* Vol. 39.pp.751–767, 2011.

[173] B. K. Panigrahi, V. R. Pandi, S. Das, Z. Cui and R. Sharma, “Economic Load Dispatch using Population Variance Harmony Search Algorithm,” *Transactions of the Institute of Measurement and Control*, vol. 34, no. 1, pp. 746 – 754, 2011.

[174] K. Zare, M.T. Haque and E. Davoodi, “Solving Non-Convex Economic Dispatch Problem with valve point effects using Modified Group Search Optimiser Method,” *Electric Power Systems Research*, vol. 84, pp. 83 – 89, 2012.

[175] K. K. Vishwakarma, H. M. Dubey, M. Pandit and B. K. Panigrahi, “Simulated Annealing Approach for Solving Economic Load Dispatch Problems with Valve Point Loading Effects,” *International Journal of Engineering, Science and Technology*, vol. 4, no. 4, pp. 60 – 72, MultiCraft, 2012.

[176] W. McCulloch and W. Pitts, “A Logical Calculus of the ideas immanent in Nervous Activity,” *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115 – 133, 1943. Reprinted in W. S. McCulloch, *Embodiments of Mind*, MIT Press, 1965 and 1988.

[177] M. Eusuff and K. Lansey, “Optimisation of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm,” *Journal of Water Resources Planning and Management*, vol. 129, no. 3, pp. 210 –225, 2003.

[178] A. E. Eiben and J. K. van der Hauw, “Solving 3-SAT by GAs Adapting Constraint Weights,” *In Proceedings of IEEE International Conference on Evolutionary Computation*, Indianapolis, USA, 13th – 16th April, 1997.

[179] P. Ross, D. Corne, and H. L. Fang, “Improving Evolutionary Timetabling with Delta Evaluation and Directed Mutation,” *Journal of Parallel Problem Solving from Nature*, pp. 556 – 565, Springer-Verlag, 1994.

[180] S. Rajoria, C. Soares, J. P. de Sousa and J. Dhar, “Predicting the Outcome of Mutation in Genetic Algorithms,” Source: http://epia2009.web.ua.pt/onlineEdition/087.pdf; accessed: 10/03/2014.

[181] J. Denies, H. B. Ahmed and B. Dehez, "Optimal Design of Electromagnetic Devices: Development of an Efficient Optimisation tool based on Smart Mutation Operations Implemented in a Genetic Algorithm," *Journal of Mathematics and Computers in Simulation*, vol. 90, pp. 244 –255, 2013.

[182] B. Bhanu and Y. Lin, "Synthesizing Feature Agents using Evolutionary Computation," *Journal of Pattern Recognition Letters*, vol. 25, pp. 1519 –1531, 2004.

[183] D. El Kateb, F. Fouquet, J. Bourcier and Y. Le Traon, "Sputnik: Elitist Artificial Mutation Hyper-heuristic for Runtime Usage of Multi-objective Evolutionary Algorithms," Source: http://arxiv.org/abs/1402.4442v1 [cs.SE], 2014; accessed: 10/03/2014.

[184] B. L. Miller and D. E. Goldberg, "Genetic Algorithms, Tournament Selection, and the Effects of Noise," *Complex Systems*, vol. 9, pp. 193 – 212, 1995.

[185] B. H. F. Hasan and M. S. M. Saleh, "Evaluating the Effectiveness of Mutation Operators on the Behaviour of Genetic Algorithms Applied to Non-deterministic Polynomial Problems," Informatica, vol. 35, pp. 513 – 518, 2011.

[186] I. Korejo, S. Yang, and C. Li, "A Comparative Study of Adaptive Mutation Operators for Genetic Algorithms," in *Proceedings of the 8th Metaheuristics International Conference*, Hamburg, Germany, July 13 –16, 2009.

[187] S. Yang, "Statistics-Based Adaptive Non-Uniform Mutation for Genetic Algorithms," in *Proceedings of the 2003 Genetic and Evolutionary Computation Conference (GECCO'03), July 9 – 11, 2003, Chicago, USA.*

[188] B. Julstrom, "What have you done for me lately? Adaptive Operator Probabilities in a Steady-State Genetic Algorithm," in *L. J. Eshelman (ed.), Proceedings of the 6th Conference on Genetic Algorithms,* 81 – 87. Morgan Kaufmann, San Mateo, CA, USA, 1995.

[189] D. Corne, P. Ross and H. L. Fang, "Genetic Algorithm Research Note 7: Fast Practical Evolutionary Timetabling," *Technical Report*, Department of Artificial Intelligence, University of Edinburgh, UK, 1994.

[190] T. Bäck, "Mutation Parameters," in T. Bäck, D. B. Fogel and Z. Michalewicz (eds.), Handbook of Evolutionary Computation, E1.2.1 – E1.2.7, Oxford University Press, 1997.

[191] J. Woodward and J. Swan, "The Automatic Generation of Mutation Operators for Genetic Algorithms," in *Proceedings of the 2012 Genetic and Evolutionary Computation Conference (GECCO'12)*, July 7 – 11, 2012, Philadelphia, USA.

[192] W. Tao, C. Xu, Q. Ding, R. Li, Y. Xiang, J. Chung and J. Zhong, "A Single Point Mutation in E2 Engances Hepatitis C Virus Infectivity and Alters Lipoprotein Association of Viral Particles," *Journal of Virology 395*, pp. 67 – 76, 2009.

[193] E. A. Feigenbaum, B. G. Buchanan and J. Lederberg, "On Generality and Problem Solving: A Case of using the DENDRAL Program," *in Machine Intelligence 6*, Edinburgh University Press, 1971.

[194] Q. Huang, P. Sue and X. Wang, "Fault Diagnosis Method of Power System Based on Rough Set and Bayesian Networks,"Advances in information Sciences and Service Sciences (AISS), vol. 5, no. 10, pp. 698 – 704, 2013

[195] Q. Shi, S. Liang, W. Fei, Y. Shi and R. Shi, "Study on Bayesian Network Parameters Learning of Power System Component Fault Diagnosis Based on Particle Swarm Optimization," *International Journal of Smart Grid and Clean Energy*, vol. 2, no. 1, pp. 132 – 137, 2013.

[196] N. Vapnik, Statistical Learning Theory, John Wiley and Sons, New York, 1998.

[197] N. Vijayalakshmi and K. Gayathri, "Optimising Power Flow Using Support Vector Machine," in *Proceedings of International Conference on Advancements in Electrical and Power Engineering*, 24-25 March 2012, Dubai, UAE, pp. 100 – 103.

[198] X. S. Yang, Nature-Inspired Metaheuristic Algorithms (Second Edition), Luniver Press, 2010

[199] I. Fister Jr, X. S. Yang, I. Fister, J. Brest and D. Fister, A Brief Review of Nature-Inspired Algorithms for Optimisation," *Elektrotehniˇ Ski Vestnik*, vol. 80, no. 3, pp. 1 – 7, 2013 (English Edition).

[200] S. Binitha and S. S. Sathya, "A Survey of Bio inspired Optimisation Algorithms," International Journal of Soft Computing and Engineering, vol. 2, no. 2, pp. 137 – 151, 2012

[201] P. Moscato, "On Evolution, Search, Optimisation, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms," *Caltech Concurrent Computation Program* (Report 826), 1989.

[202] L. J. Fogel, A. J. Owens, M. J. Walsh, *Artificial Intelligence through Simulated Evolution*, John Wiley, 1966.

[203] S. Luke *et al*, "ECJ 22: A Java-based Evolutionary Computation Research System," George Mason University, available at: http://cs.gmu.edu/~eclab/projects/ecj/, accessed: 04/04/2014.

[204] M. Hauschild and M. Pelikan, "An Introduction and Survey of Estimation of Distribution Algorithms," *Swarm and Evolutionary Computation*, vol. 1, pp. 111 – 128, 2011.

[205] K. A. Folly and S. P. Sheetekela, "Application of a Simple Estimation of Distribution Algorithm to Power System Controller Design," in *Proceedings of 45th International Universities Power Engineering Conference (UPEC)*, 31st August to 3rd September, 2010.

[206] J. Kennedy and R. Eberhart, "Particle Swarm Optimisation," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, pp. 1942 – 1948, 1995.

[207] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez and R. G. Harley, "Particle Swarm Optimisation: Basic Concepts, Variants and Applications in Power Systems," IEEE Transactions on Evolutionary Computation, vol. 12, no. 2, pp. 171 – 195, 2008.

[208] S. He, Q. H. Wu, "A Novel Group Search Optimiser Inspired by Animal Behavioural Ecology," in *Proceedings of IEEE Congress on Evolutionary Computation*, Vancouver, BC, Canada, July 16-21, 2006, pp. 4415 – 4421.

[209] I. D. Couzin, J. Krause, N. R. Franks and S. A. Levin, "Effective Leadership and Decision Making in Animal Groups on the move," *Nature*, vol. 433, pp. 513 – 516, 2005.

[210] D. Simon, "Biogeography-Based Optimisation," IEEE Transactions on Evolutionary Computation, vol. 12, no. 6, pp. 702 – 713, 2008.

[211] B. Khokhar, K. P. S. Parmar and S. Dahiya, "Application of Biogeography-based Optimization for Economic Dispatch Problems," *International Journal of Computer Applications*, vol. 47, no.13, pp. 25 – 30, 2012.

[212] A. Nazari and A. Hadidi, "Biogeography Based Optimisation Algorithm for Economic Load Dispatch of Power System" *American Journal of Advanced Scientific Research*, vol. 1, no. 3, pp. 99 – 105, 2012.

[213] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, 1983, "Optimisation by Simulated Annealing," *Science New Series*, vol. 220, pp. 671 – 680.

[214] V. Cerny, 1985, "A Thermodynamical Approach to the Travelling Salesman Problem: An Efficient Simulated Algorithm," *Journal of Optimisation, Theory and Applications*, vol. 45, pp. 41 – 51.

[215] Z. W. Geem, J. H. Kim and G. V. Loganathan, "A New Heuristic Optimisation Algorithm: Harmony Search," *Simulation*, vol. 76, no. 2, pp. 60 – 68, 2001.

[216] T. Ratniyomchai, A. Oonsivilai, P. Pao-La-Or and T. Kulworawanichpong, "Economic Load Dispatch Using Improved Harmony Search," *WSEAS Transactions on Systems and Control*, vol. 5, no. 4, pp. 248 – 257, 2010.