Cape Peninsula
University of Technology

DEVELOPMENT OF DECOMPOSITION METHODS FOR SOLUTION OF A MULTIAREA POWER DISPATCH OPTIMISATION PROBLEM

by

SENTHIL KRISHNAMURTHY

Thesis submitted in fulfilment of the requirements for the degree

Doctor of Technology: Electrical Engineering

in the Faculty of Engineering

at the Cape Peninsula University of Technology

Supervisor: Professor Raynitchka Tzoneva

Bellville
September 2013

DECLARATION

I, Senthil Krishnamurthy, declare that the contents of this thesis represent my own unaided work, and that the thesis has not previously been submitted for academic examination towards any qualification. Furthermore, it represents my own opinions and not necessarily those of the Cape Peninsula University of Technology.

Signed **Date**

ABSTRACT

The objective of the economic dispatch problem of electrical power generation is to schedule the committed generating unit outputs to meet the required load demand while satisfying the system equality and inequality constraints. The thesis formulates single area and multi-area Combined Economic Emission Dispatch (CEED) problem as single criterion, bi-criterion and multi-criteria optimisation problems based on fuel cost and emission criterion functions, constraints over the operational limits of the generator and the tie-lines, and requirements for a balance between the produced power and the system demand and power loss.

Various methods, algorithms and softwares are developed to find solution of the formulated problems in single area and multi-area power systems. The developed methods are based on the classical Lagrange's and on the meta-heuristic Particle Swarm Optimisation (PSO) techniques for a single criterion function. Transformation of the bi-criteria or multi-criteria dispatch problem to a single criterion one is done by some existing and two proposed in the thesis penalty factors.

The solution of the CEED problems is obtained through implementation of the developed software in a sequential way using a single computer, or in a data-parallel way in a Matlab Cluster of Computers (CC). The capabilities of the developed Lagrange's and PSO algorithms are compared on the basis of the obtained results. The conclusion is that the Lagrange's method and algorithm allows to receive better solution for less computation time. Data-parallel implementation of the developed software allows a lot of results to be obtained for the same problem using different values of some of the problem parameters.

According to the literature papers, there are many algorithms available to solve the CEED problem for the single area power systems using sequential methods of optimisation, but they consume more computation time to solve this problem. The thesis aim is to develop a decomposition-coordinating algorithm for solution of the Multi Area Economic Emission Dispatch (MAEED) problem of power systems. The MAEED problem deals with the optimal power dispatch inside and between the multiple areas and addresses the environmental issue during the economic dispatch. To ensure the system security, tie-line transfer limits between different areas are incorporated as a set of constraints in the optimisation problem. A decomposition coordinating method based on the Lagrange's algorithm is developed to derive a set

of optimal solutions to minimize the fuel cost and emissions of the multi-area power systems.

An augmented function of Lagrange is applied and its decomposition in interconnected sub problems is done using a new coordinating-vector. Task-parallel computing in a Matlab Cluster is used to solve the multi-area dispatch problem. The calculations and tasks allocation to the Cluster workers are based on a shared memory architecture. Implementation of the calculation algorithm using a Cluster of Computers allows quick and simpler solutions to the multi-area CEED problem.

The thesis applied the developed algorithms for the various problem formulation scenarios, i.e. fuel cost and emission function with and without valve point loading effect, quadratic and cubic fuel cost and emission functions. The various IEEE benchmark models are used to test the developed Lagrange's and PSO algorithms in the sequential, data-parallel, and task-parallel implementations.

Developed methods, algorithms and software programmes can be applied for solution of various energy management problems in the regional and national control centres, smart grid applications, and in education and research institutions.

ACKNOWLEDGEMENTS

I wish to thank:

- My family members Father Krishnamurthy, Mother Usha Rani, Brother Manikandan, Sister Shankar Devi , Junior Abirami , Uncle Raja, and Grand Father Natam for their support, patience endurance and understanding during the study.
- Professor Raynitchka Tzoneva for her invaluable help in my career. I consider myself fortunate to come across such a noble professor. She has shown a great interest in all phases of this project to ensure its success. But for her keen involvement, this thesis would not have been possible. I am thankful to her for assigning me one of the most important research projects in the field of Energy Management Systems and Real-Time Distributed Systems.
- Adjunct Professor Alexander Apostolov for his valuable suggestions on all phases of this project and the seminar on IEC61850 standard for Substation Automation Systems.
- The Centre for Real Time Distributed Systems and Substation Automation and Energy Management Systems at the Cape Peninsula University of Technology for their assistance in different areas of the research.

DEDICATION

For my Family Members, Junior Abirami and Grand Father Natam

TABLE OF CONTENTS

CHAPTER THREE: COMBINED ECONOMIC EMISSION DISPATCH PROBLEM USING LAGRANGE'S ALGORITHM

CHAPTER FOUR: COMBINED ECONOMIC EMISSION DISPATCH PROBLEM SOLUTION BASED ON THE PARTICLE SWARM OPTIMIZATION ALGORITHM

CHAPTER FIVE: PARALLEL COMPUTING AND ITS IMPLEMENTATION

CHAPTER SIX: INVESTIGATION OF THE SINGLE AREA COMBINED ECONOMIC EMISSION DISPATCH PROBLEM CAPABILITIES OF THE DEVELOPED ALGORITHMS USING PARALLEL COMPUTING

CHAPTER SEVEN: **MULTI AREA ECONOMIC EMISSION DISPATCH PROBLEM FORMULATION AND SOLUTION BY A DECOMPOSITION-COORDINATING METHOD**

CHAPTER EIGHT: CONCLUSION AND FUTURE RECOMMENDATIONS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

GLOSSARY

Terms/Acronyms/Abbreviations	Definition/Explanation
AC	Alternating Current
ACE	Area Control Error
AGC	Automatic Generation Control
Algorithm	An effective method expressed as a finite list of well-defined instructions for calculating a function
ANN	Artificial Neural Network
API	Application Programming Interface
ATC	Available Transfer Capability
BA	Bees Algorithm
BGO	Bi-criterion Global Optimization
CC	Cluster of Computers: A group of linked computers, working together closely thus in many respects forming a single computer. The components of a cluster are commonly, but not always, connected to each other through fast local area networks
CCCP	Combined Cycle Co-generation Plant
CEED	Combined Economic Emission Dispatch
CIWA	Chaotic Inertial Weight Approach
CM	Component Model
Complex System	A system composed of interconnected parts that as a whole exhibit one or more properties
Constraints	It is a subsystem of the objective function of the dispatch problem. It describes the limitations of the objective function to achieve its potential
CPU	Central Processing Unit
CT	Computational Time
CU	Control Unit
CUF	Commitment Utilization Factor
DC	Direct Current
DE	Differential Evolution
Decomposition	It divides the main problem into a many sub problems and solves them simultaneously to get the solution in short time.
DED	Dynamic Economic Dispatch
Distributed computing	Computing with distributed applications, running the application on several nodes simultaneously.

DMA	Distributed Memory Architecture
DOS	Disk Operating System
DP	Dynamic Programming
DS	Direct Search
DS	Data Stream
DWDP	Dantzing Wolfe Decomposition Principle
Dynamic System	It describes the time dependence of a point in a geometrical space.
ED	Economic Dispatch: The short-term determination of the optimal number of generators connected to the power system to meet the system load, at the lowest possible cost.
EJB	Enterprise Java Beans
Electric Power System	A network of electrical components used to supply, transmit and use electric power.
EM	E-constraint Method
EP	Evolutionary Programming
EPMCEED	Evolutionary Programming based Modified Combined Economic Emission Dispatch
ES	Expert System
FGTS	Fuzzy Guided Tabu Search
FL	Fuzzy Logic
FMEP	Fuzzy Mutated Evolutionary Programming
FMOPF	Fuzzy Multiobjective Optimal Power Flow
FORTAN	FORmula TRANslator
FPSO	Fuzzied Particle Swarm Optimization
Fuel Cost	The amount required to produce 1MW of power in 1 hour
GA	Genetic Algorithm
GAMS	General Algebraic Modeling System
GSA	Gravitational Search Algorithm
GSM	Grid Service Model
HA	Heuristic Approach
HNN	Hopfield Neural Network
HPPC	High Performance Parallel Computing
HS	Harmony Search
HVAC	High Voltage Direct Current

HVDC	High Voltage Alternating Current
IEEE	Institute of Electrical and Electronics Engineer
Interconnected Power System	An whole power system will be described as a separate sub areas to transfer the power from one area to another area with tie-power limits.
IPSO	Improved Particle Swarm Optimisation
IS	Instruction Stream
ISM	Interconnected System Response
ISO	Independent System Operator
JM	Jacobin Method
JNDI	Java Naming and Directory Interface
Job	The complete large – scale operation to perform in Matlab, composed of a set of tasks.
Job Manager	The Math works process that queues jobs and assigns tasks to workers. A third party process that performs this function is called scheduler. The general term “scheduler” can also refer to a job manager
KKT	Karush-Kuhn-Tucker
LA	Lagrange’s Algorithm
LI	Lambda Iteration
Linear System	It is a mathematical model of a system based on the use of a linear operator.
MAED	Multi Area Economic Dispatch: The whole dispatch problem has subdivided into one or more areas. The composition of all the areas is called multi area. The multi-area economic dispatch problem includes the fuel cost criterion functions of the multi-area power system
MAEED	Multi Area Economic Emission Dispatch: The multi-area economic emission dispatch problem includes the combined fuel cost and emission criterion functions of the multi-area power system
MACEED	Multi Area Combined Economic Emission Dispatch
MATLAB	MATrix LABoratory: It is a numerical computing environment and fourth-generation programming language. It allows matrix manipulations, plotting of functions and data, implementation of algorithms and creation of user interfaces.
MDCE	Matlab Distributed Computing Engine
MDCS	Matlab Distributed Computing Server
Method	The procedure or techniques arranged in the steps to accomplish the solution

MIMD	Multiple Instruction Multiple Data
MISD	Multiple Instruction Single Data
MJS	Matlab Job Scheduler
MM	Memory Module
Model	A usable knowledge based representation of the essential aspects of an existing system.
MOEP	Multi Objective Evolutionary Programming
MPI	Message Passing Interface
MPSO	Modified Particle Swarm Optimisation
Multicriterial	It is a discipline aimed at supporting decision makers faced with making numerous and sometimes conflicting evaluations.
NFM	Network Flow Modelling
NFP	Network Flow Programming
NLP	Non Linear Programming
Node	A computer that is part of a cluster.
NR	Newton Raphson
NUMA	Non Uniform Memory Access
OPF	Optimal Power Flow
PAPSO	Parallel Asynchronous Particle Swarm Optimization
Particle	A small localized object to which can be ascribed several physical properties such as volume or mass.
Partition	A partition is a division of a system or its constituting elements into distinct independent parts.
PCT	Parallel Computing Toolbox
Power System Demand	The consumers total load connected to the power system .
PS	Pattern Search
PSO	Particle Swarm Optimization: It is a computational method that optimise a problem solution by iteratively trying to improve a candidate solution with regard to a given measure of quality.
PSPSO	Parallel Synchronous Particle Swarm Optimization
PTDF	Power Transfer Distribution Factor
PU	Processing Unit
PWS	Power World Simulator
QP	Quadratic Programming

QPSO	Quantum-inspired Particle Swarm Optimisation
RCGA	Real Coded Genetic Algorithm
Reliability	The ability of a system or component to perform its required functions under stated conditions for a specified period of time.
RMI	Remote Method Invocation
RSCAD	Real time Simulator for Computer Aided Design
RTDS	Real Time Digital Simulator : The power systems simulation technology for fast, reliable, accurate and cost-effective study of power systems with complex High Voltage Alternating Current (HVAC) and High Voltage Direct Current (HVDC) networks. The RTDS Simulator is a fully digital electromagnetic transient power system simulator that operates in real time.
RTO	Regional Transmission Organization
SAMF	Sequential Approach with Matrix Framework
SAS	Simulated Annealing Search
Scheduler	The process, either third party or the MathWorks job manager, that queues jobs and assigns tasks to workers.
SDP	Semi Definite Programming
SIMD	Single Instruction Multiple Data
SIMLAB	SIMulation LABorartory
SISD	Single Instruction Single Data
SMA	Shared Memory Architecture
SPEA	Strength Pareto Evolutionary Algorithm
Task	One segment of a job to be evaluated by a worker.
Tie-line	A communication connection between extensions of one area to another area in the inter connected power system.
TSA	Tabu Search Algorithm
UCA	Unit commitment Approach
UMA	Uniform Memory Access
Velocity	The measurement of the rate and direction of change in the position of an object.
VP	Valve Point
VPEL	Valve Point Effect Loading
Worker	The Matlab process that performs the task computations. Also known as the Matlab worker
WSM	Weighted Sum Method

MATHEMATICAL NOTATIONS

Symbols / letters	Definition/Explanation
$\Delta\lambda$	Gradient procedure of λ
α_i, β_i	Valve point effect fuel cost coefficients of generating unit *i*
γ_i, δ_i	Valve point effect emission cost coefficients of generating unit *i*
$a_{SO_2i}, b_{SO_2i}, c_{SO_2i}, d_{SO_2i}$	Sulphurdioxide emission coefficients of generating unit *i*
$a_{CO_2i}, b_{CO_2i}, c_{CO_2i}, d_{CO_2i}$	Carbondioxide emission coefficients of generating unit *i*
$a_{NO_xi}, b_{NO_xi}, c_{NO_xi}, d_{NO_xi}$	Nitrogenoxide emission coefficients of generating unit *i*
E_{TSO_2}	Total sulphurdioxide emission
E_{TCO_2}	Total carbondioxide emission
E_{TNO_x}	Total nitrogendioxide emission
$F_C - SO_2$	Fuel cost of sulphurdioxide emission
$F_C - CO_2$	Fuel cost of carbondioxide emission
$F_C - NO_x$	Fuel cost of nitrogenoxide emission
F_{TSO_2}	Combined economic emission dispatch fuel cost of sulphurdioxide emission
F_{TCO_2}	Combined economic emission dispatch fuel cost of carbondioxide emission
F_{TNO_x}	Combined economic emission dispatch fuel cost of nitrogenoxide emission
h_{SO_2i}	Price penalty factor of sulphurdioxide emission
h_{CO_2i}	Price penalty factor of carbondioxide emission
h_{NO_xi}	Price penalty factor of nitrogendioxide emission
h_{AvgSO_2i}	Average price penalty factor of sulphurdioxide emission
h_{AvgCO_2i}	Average price penalty factor of carbondioxide emission
h_{AvgNO_xi}	Average price penalty factor of nitrogendioxide emission
h_{comSO_2i}	Common price penalty factor of sulphurdioxide emission
h_{comCO_2i}	Common price penalty factor of carbondioxide emission

$h_{comNO_x i}$	Common price penalty factor of nitrogendioxide emission
F_{TAvgSO_2}	Combined economic emission dispatch fuel cost using average price penalty factor of sulphurdioxide emission
F_{TAvgCO_2}	Combined economic emission dispatch fuel cost using average price penalty factor of carbondioxide emission
F_{TAvgNO_x}	Combined economic emission dispatch fuel cost using average price penalty factor of nitrogenoxide emission
F_{TcomSO_2}	Combined economic emission dispatch fuel cost using common price penalty factor of sulphurdioxide emission
F_{TcomCO_2}	Combined economic emission dispatch fuel cost using common price penalty factor of carbondioxide emission
F_{TcomNO_x}	Combined economic emission dispatch fuel cost using common price penalty factor of nitrogenoxide emission
L_{SO_2}	Lagrange's variable for the case of sulphurdioxide emission
V_{pi}	Initial velocity of the particle swarm
ω	Inertia weight
$\omega^{\min}, \omega^{\max}$	Minimum and maximum inertia weight
$V_{pi}^{\min}, V_{pi}^{\max}$	Minimum and maximum velocities
P_{pd}	Power produced by the slack bus generator
P_{pi}	Initial power produced by the generator for the unit i
$P_{pi}^{\min}, P_{pi}^{\max}$	Minimum and maximum real power limits of the generator for the unit i
P_{pi}^{best}	Best positions of the particles
G_{pi}^{best}	Global best positions of the particles
V_{pi}^{new}	New velocity of the particle swarm
P_{pi}^{new}	New power produced by the generator for the unit i
F_T^{new}	New fuel cost of the combined economic emission dispatch problem
F_T^{best}	Best fuel cost of the combined economic emission dispatch problem
$F_T^{newbest}$	New best fuel cost of the combined economic emission dispatch problem

F_{Cp}	Fuel cost of PSO algorithm, where $p=\overline{1,N_p}$
E_{Tp}	Emission cost of the PSO algorithm , where $p=\overline{1,N_p}$
F_{Tp}	Combined economic emission dispatch problem fuel cost of PSO algorithm, where $p=\overline{1,N_p}$
h_{pi}	Price penalty factor of the particle swarm, where $p=\overline{1,N_p}$
a_{mn}, b_{mn}, c_{mn}	Fuel cost coefficients of the n^{th} generator in the m^{th} area
P_{Tm}	Vector of the power transmission between the m^{th} area and all other area, where $m=\overline{1,M-1}$
P_{Tmj}	Tie-line power flow from area m to area j, *where* $j=\overline{m+1,M}$
P_{Tjm}	Tie-line power flow from area j to area m, *where* $j=\overline{m+1,M}$
$P_{mn,\min}, P_{mn,\max}$	Minimum and Maximum power that can be produced by the n^{th} generator in the m^{th} area, where $m=\overline{1,M}$, $n=\overline{1,N_m}$
P_{mn}	Real power produced by the n^{th} generator in the m^{th} area
$P_{Tmj,\min}, P_{Tmj,\max}$	Minimum and maximum active power sent through the tie-lines
$B_{mnr}, B_{m0n}, B_{m00}$	Transmission loss coefficients of an interconnected system
ρ_{jm}	Fractional loss rate from area j to area m
d_{mn}, e_{mn}, f_{mn}	Emission cost coefficients of the n^{th} generator in the m^{th} area
h_{mn}	Price penalty factor of the n^{th} generator in the m^{th} area
λ_m	Lagrange's variable of the m^{th} area
P_{Lm}	Transmission loss of the m^{th} area
q_{mj}	Transmission coefficient for the cost for transmission of the power from area m to area j
q_{jm}	Transmission coefficient for the cost for transmission of the power from area j to area m
e_{PTjm}	Tolerance value of the tie-line power flow from area m to area j
e_{PTmj}	Tolerance value of the tie-line power flow from area j to area m
$e_{m\lambda}$	Tolerance value of the Lagrange's variable λ of the m^{th} area
YSh_{ij}	Transmission line Shunt admittance of line i to j

$ZSER_{ij}$	Transmission line Series impedance of i to j
a_i, b_i, c_i	Fuel coefficients of generating for the unit i
B_{ij}, B_{0i}, B_{00}	Transmission loss coefficients
$c1, c2$	Acceleration constant
$CEED-CO_2$	Combined economic emission dispatch solution for the case of carbondioxide emission
$CEED-NO_x$	Combined economic emission dispatch solution for the case of nitrogenoxide emission
$CEED-SO_2$	Combined economic emission dispatch solution for the case of sulphurdioxide emission
$CEED-SO_2-NO_x-CO_2$	Combined economic emission dispatch solution for the case of all three pollutants (sulphurdioxide, carbondioxide and nitrogenoxide emissions)
CO_2	Carbondioxide
CT_B	Best computation time
d_i, e_i, f_i	Emission coefficients of generating unit i
E, D	Vector to calculate the real power of the generator for single area power system using Lagrange's
E_m, D_m	Vector to calculate the real power of the generator for multi-area power system using Lagrange's
E_T	Total Emission
E_{TVP}	Total emission with valve point effect loading
F_C	Fuel cost
F_{CVP}	Total fuel cost with valve point effect loading
F_{TVP}	Combined Economic Emission Dispatch fuel cost with valve point effect loading
h_i	Price Penalty factor
$Iter_{\max}$	Maximum iteration
k	Number of iterations
M	Number of the interconnected areas
n	Number of generating units
N_m	Number of generators belonging to area m

NO_x	Nitrogenoxide
N_p	Number of the particles in the swarm
p	processors
P	Vector of the real power produced in the whole power system
P_D	Power demand
P_i	Real power generation of a generator unit i
$P_{i,\min}, P_{i,\max}$	Minimum and maximum value of real power allowed at a generator i
P_L	Transmission line loss
P_m	Vector of the real power produced in the m^{th} area, where $m=\overline{1,M}$
P_T	Vector of the power transmission between all areas
Q_i	Reactive power generation of a generator unit i
$Q_{i,\min}, Q_{i,\max}$	Minimum and maximum value of reactive power allowed at a generator i
$rand()$	Random number
s	Speed-up of the parallelization program
S	Apparent power
SO_2	Sulphurdioxide
V_i	Bus voltage magnitude
α	Step size of the gradient procedure
δ_i	Phase angle of a bus
ε	Tolerance value of the gradient procedure
λ	Lagrange's multiplier

CHAPTER ONE

INTRODUCTION

1.1 Introduction

An electric power system consists of generation, transmission and distribution utilities to enchance electrical power to the consumers. Economic dispatch is the short term determination of the optimal output power of generators to meet the system load and operate the generators at the lowest fuel cost. Energy management system is used to monitor, control and optimize the performance of the generators used in the electric utility grid. Most of the reference papers address the economic dispatch problem for single area power system and few papers address the economic dispatch problem for multi-area power system but the optimisation problem is still solved in a sequential way. The thesis aim is to solve the multi-area economic dispatch problem using a developed Lagrange's decomposed method. In addition to that, it formulates the Combined Economic Emission Dispatch (CEED) problem by additionally considering the emissions of the thermal power plants in order to reduce the global warming. The CEED Problem is solved using the developed Lagrange's and Particle Swarm Optimization (PSO) algorithms. This chapter presents the awareness of the problem in part 1.2, problem statement is described in part 1.3, aim and objectives of the research are given in part 1.4, hypothesis, delimitations of the research and assumptions are given in parts 1.5 – 1.7 respectively, deliverables and thesis chapter break down are described in parts 1.8 – 1.9 respectively, and the conclusions is given in part 1.10.

1.2 Awareness of the problem

Large interconnected power systems are usually decomposed into areas or zones based on criteria, such as the size of the electric power system, network topology and geographical location. Multi Area Economic Emission Dispatch (MAEED) Problem is an optimisation task in power system operation for allocating amount of generation to the committed units within the areas. Its objective is to minimize the fuel cost subject to the power balance, generator limit, transmission line and tie-line constraints. The solution of the MAEED problem determines the amount of power that can be economically generated in the areas and transferred to other areas if it is needed without violating tie-line capacity constraints and the whole power network constraints. The benefits of the multi-area power systems are continuity of service, economy on operation using the optimum generation schedule (economic dispatch), small frequency deviations due to the fact that all the generators participate during the load of the large interconnected power system changes, and transportation of energy from one sub area to another is possible. These benefits are achieved through the interconnection of tie-lines between the sub areas as shown in Figure 1.1.

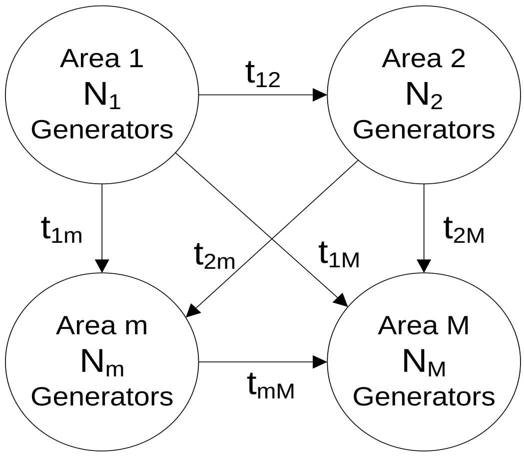


Figure 1.1: **Model of a Multi-area power system with tie-line power transfer**

where

N_m	Number of generators belonging to area *m*
M	Number of the interconnected areas
t_{1m}	Number of tie-lines from area1 to area *m*
t_M	Number of tie-lines of the interconnected areas

In addition to that, emissions of the thermal power plants which create pollution to the environment are considered. The cost minimum condition corresponds to minimum cost with considerable amount of emission. Similarly the emission minimum condition produces minimum emission with higher deviation from the minimum cost. These two conditions cannot be implemented simultaneously. Hence, the feasible optimum corresponds to a small deviation in cost with an allowable tolerance in emission taking into account of the emission constraints. This has been termed as Combined Economic Emission Dispatch (CEED) problem.

The investigations in the thesis concentrate first on developing methods for solution of the CEED problem and second on development methods for parallel solution of the MAEED problem on the basis of data-decomposition and task-decomposition approaches.

Single area CEED problem is solved to minimize the fuel cost and emission of the system within the area, but it is important to decompose the large complex power system problem into a set of sub-problems corresponding to the power system areas and the solution of the sub-problems has benefits which allows to transport energy from one area to other area through tie-lines and simultaneously minimize the fuel cost and emission of the interconnected power systems. The solution of the MAEED problem is still implemented in a sequential way. This thesis describes the importance of the parallel solution of the MAEED problem developing a Lagrange's decomposition-coordinating method in a Cluster of Computers. Most of the literature papers used either classical or heuristic method for solution of the CEED problem. The thesis developed two methods classical (Lagrange's) and heuristic (PSO) for the CEED problem and comparison of the results of the two methods is done. The methods characteristics such as solution type (global or local), number of iterations, computation time used by the two methods are compared.

The research investigations in this thesis are looking at the solution of the following research questions:

- Effect of quadratic and cubic fuel and emission cost functions and their impact on the solution of the CEED problem.
- Valve point loading effect of the thermal power system.
- Effect of pollutants such as SO_2, NO_x and CO_2 emissions of the thermal power plants and their impact on the solution of the dispatch problem.
- Role of price penalty factor used in the combined economic emission dispatch problem and their impact on the solution of the CEED problem.
- How to develop Lagrangian decomposition method to solve the multi-area economic dispatch problem in a parallel way using Cluster of Computers.
- To develop a Lagrangian and PSO methods to solve the CEED problem in a sequential way.
- How to implement data-parallel solutions of the sequential Lagrangian and PSO algorithms in a Matlab Cluster of Computers.
- How to apply the decomposition principle to decompose the large complex power system dispatch problem into a set of sub-problems corresponding to the power system areas.
- Investigate the use of Matlab parallel computing software in a Cluster of Computers.
- How to compare the developed Lagrange's (classical) and PSO (meta-heuristic) optimisation methods and their impacts on the solutions.

- How to compare the MAEED solution with the single area one.

The solutions of the MAEED problem in the conditions of deregulation are difficult, due to the model size, nonlinearity, and interconnections, and require intensive computations in real-time. Matlab parallel computing gives possibilities for reduction of the problem complexity and the time for calculation by the use of parallel processing for running advanced application programs efficiently, reliably and quickly. A cluster of computers working in Matlab software environment is used to implement the optimisation algorithms. Parallelization of the solution is done through decomposition of the MAEED problem according to the power system interconnected areas and coordination of the obtained solutions for every area by a coordinator. A classical (Lagrange) decomposition-coordinating method for parallel computing is developed and implemented using IEEE benchmark power system models. In addition to that softwares for Lagrange's and Particle Swarm Optimization (PSO) methods are developed for CEED problem in a sequential and parallel ways.

1.3 Statement of the Problem

The main problem is:

- To make the formulation of the dispatch optimization problem relevant to the new condition of deregulation when the power system is considered to consist of interconnected sub-systems.
- To solve the problems by a method that will be relevant to the conditions of deregulation of power system and will reduce complexity, time of the calculation and also communication of data between the interconnected sub-systems.

Two types of research sub-problems are solved in order to complete the investigations in the thesis.

- Design based sub-problems, and
- Implementation sub-problems

1.3.1 Design based sub-problems

Sub-problem 1: Analysis of existing methods used in single and multi-area economic dispatch problem.

Sub-problem 2: Formulation of the single area CEED problems using quadratic and cubic fuel cost functions with and without valve point loading effect.

Sub-problem 3: Development of Lagrange's method for the formulated single area CEED problems solution using various price penalty factors.

Sub-problem 4: Development of PSO method for the formulated CEED problems solution using various price penalty factors.

Sub-problem 5: Analysis of the parallel processing hardware architecture and Matlab parallel computing toolbox.

Sub-problem 6: Development of Lagrange's decomposition-coordinating method for MAEED problem parallel solution in a Cluster of Computers.

1.3.2 Sub-problems on implementation

Sub-problem 1: Development of software for implementation of the Lagrange's methods to solve the CEED problems using quadratic fuel cost and emission criterion functions.

Sub-problem 2: Development of software for implementation of the Lagrange's methods to solve the CEED problems using cubic fuel cost and emission criterion functions.

Sub-problem 3: Development of software for implementation of the Lagrange's methods to solve the CEED problems using quadratic fuel cost and emission criterion functions with valve point loading effect.

Sub-problem 4: Development of software for implementation of the PSO method to solve the single area CEED problem with quadratic fuel cost and emission criterion functions.

Sub-problem 5: Development of software for data-parallel implementation of the Lagrange's and PSO method to solve the single area CEED problem with quadratic fuel cost and emission criterion functions.

Sub-problem 6: Development of software for task-parallel implementation of the Lagrange's decomposition-coordinating method to solve the MAEED problem in a Cluster of Computers.

1.4 Research Aim and Objectives

Electric utility systems are interconnected to achieve the benefits of minimum production cost, maximum reliability and better operating conditions such as reserve sharing, improved stability, and operation under emergencies. The power system comprises of either single area or multi areas based on the complexity of the system. The literature review reported that economic dispatch problem has been solved for both single area and multi area electric power systems in a sequential way in a single computer so far. The present and future research directions focus on the multi area power system, where each area is located in the same or various geographical regions. The management of the Smart grid requires the information of MAED solution in real-time in order to take decisions for the behaviour of the power system and to improve the efficiency, reliability, economics, and sustainability of the production and distribution of electricity.

On the basis of the above it can be concluded that new formulation of the multi-area dispatch problems is needed and new methods to solve these problems paying attention to the interconnected structure of the power system have to be developed. These necessities determine the aim and objectives of the thesis.

1.4.1 Aim

To investigate existing and develop improved methods and algorithms for solution of the single area optimization dispatch problems. To develop software for solution of the problems in a single computer. To formulate the multi-area economic emission dispatch problem in a way it corresponds to the requirements of the deregulated power system structure and the future challenges of the smart grid. To develop a decomposition-coordinating method for solution of the multi-area dispatch problem using Lagrange's algorithm. To develop software for both data-parallel and task-parallel implementation of the single area and multi-area problem algorithms in a Cluster of Computers.

1.4.2 Objectives

i. To conduct literature review on the approaches, algorithms, and software programs on the exiting methods for solution of the single area and multi-area economic dispatch problem.

ii. To formulate the quadratic, cubic, and valve point loading effect criteria and investigate their impacts on the solution of the economic dispatch problem

iii. To develop single area sequential Lagrange's and PSO methods for solution of the CEED problems with various criterion functions.

iv. To formulate the MAEED problem based on quadratic fuel cost and emission criterion functions

v. To develop Lagrange’s decomposition- coordinating method for solution of the MAEED problem.

vi. To develop software for sequential implementation of the developed Lagrange's and PSO methods for the considered types of CEED problem.

vii. To develop software for data-parallel implementation of the developed Lagrange's and PSO methods for the considered types of CEED problem in a cluster of computers.

viii. To apply the developed software to standard IEEE benchmark models of power systems.

ix. To develop software for parallel calculation of the Lagrange’s MAEED problem solutions in a Cluster of Computers based on the developed decomposition-coordinating method.

x. To investigate the performance of the developed methods using Cluster of Computers in MATLAB software environment.

xi. To improve the existing parallel computing infrastructure with novel optimization techniques for better performance, and computing resource utilization.

1.5 Hypothesis

The hypothesis is based on the review investigation of methods and algorithms used in the referenced papers to solve the single and multi-area economic dispatch problem.

The review investigations state that various methods are used to solve the single and multi-area economic dispatch problem, each method has its own merits and demerits according to their characteristics, solution, computational time, accuracy and reliability.

1. The optimisation algorithms used in economic dispatch problem solution are either based on the classical or meta-heuristic approaches so far. The hypothesis is that the classical algorithms produce better values of the criterion functions. This hypothesis is proved by development of two optimisation algorithms for solution of the economic dispatch problem namely, Lagrange's a classical approach and PSO, a meta-heuristic one. The solution and computational time of the two methods to solve the economic dispatch problem are compared.
2. The fuel cost curves with quadratic and cubic cost function and with/ without valve point loading effect are considered. The hypothesis is that the impact of the different criterion functions on the solution of the economic dispatch problem can be analysed and evaluated by implementation of the developed Lagrange's and PSO algorithms.
3. Most of the reference papers use Max-Max price penalty factor for CEED problem solution. Thesis hypothesis is that a new price penalty factor called 'Min-Max' in addition to that 'Max-Max' one can be developed. This newly developed price penalty factor provides less value of the fuel cost and less emission value in comparison with Max-Max one.
4. The large interconnected power system is decomposed into one or more sub areas. The multi area economic dispatch problem is to schedule the generators within the area and to minimize the fuel cost of the generators that belong to that area. The MAED problem is still solved in a sequential way in a single computer so far. The thesis is based on the hypothesis that a decomposition-coordinating Lagrange's based method and algorithm for solution of the MAED problem in a task-parallel way using Matlab parallel computing toolbox in a Cluster of Computers is possible.

   The proof of the hypothesis is done by development of software implementing the method and providing various experiments with it.

1.6 Delimitation of research

The research project emphasizes on the characteristics of the various optimisation algorithms developed to solve the combined economic emission dispatch problems and multi-area economic emission dispatch problems.

- Lagrange's method is developed and implemented first for a single area economic emission dispatch problem and second for the multi-area one.

- PSO method is developed and implemented for the single area economic emission dispatch problem.
- Lagrange's decomposition method is implemented for the multi area economic emission dispatch problem using Matlab parallel computing toolbox in a Cluster of Computers.
- Finally the comparison of the Lagrange's and PSO methods are done. The investigation will not include the other existing optimization methods but will simply discuss them using the explanatory methods.

The following steps are carried out in order to implement the developed optimisation methods:

Step 1: Lagrange’s method is developed to solve the single area economic emission dispatch problem with quadratic or cubic cost functions.

Step 2: The PSO method is developed to solve the single area economic emission dispatch problem with quadratic cost function.

.Step 3: The Lagrange's method is developed on the bases of the structure of the multi-area power system through the process of decomposition and coordination.

Step 4: Software is developed for sequential and parallel solution of the single and multi-area economic emission dispatch problems and is implemented in Matlab environment.

Other optimisation problems are not considered.

1.7 Assumptions

The research is conducted on the basis of the assumptions used to solve the economic emission dispatch problem, they are:

i. For simplicity the power demand of the power system is considered as constant for some period of time for which the economic dispatch problem is solved. But in reality the power demand is changing in real-time in respect to the load consumed by the consumers.

ii. It is a common practice for including the effect of transmission losses in the economic dispatch problem to express the total transmission loss as a quadratic function of the generator power outputs given by Kron's formulae (Dhillon et al, 1994) but in reality the active power transmission losses are calculated using the power flow equations of the power system.

iii. In the multi-area power system, the maximum number of tie-lines is given as, $N_T = \frac{(M-1)M}{2}$. This formula can be calculated by induction if all tie-lines between all areas exist

where

M Total number of the area and

N_T Number of tie-lines.

iv. The tie-lines are not considered in the emission function of the interconnected power system because the transmission of power does not create chemical pollution.

v. Lagrange's method follows the gradient procedure to obtain the global solution.

vi. Lagrange's method needs the initial guess of the Lagrange's multiplier λ to start the search process

vii. PSO method needs the initial guess of acceleration factors, inertia constant and random numbers to start the search process.

viii. PSO does not require that the optimization problem be differentiable as like classic optimization methods such as Lagrange's, gradient descent and quasi-newton methods.

ix. Metaheuristic algorithm PSO do not guarantee an optimal solution is ever found since it does not use the gradient procedure of the problem being optimized.

x. The time for communication between the computers in the Cluster is smaller than the time for calculation of the sub-problems solved in them.

xi. At each iteration of the iterative optimization algorithms the computational cost and memory engagement is constant.

xii. Implementation and parallelization of the Lagrange's and PSO methods are possible.

1.8 The deliverables of the thesis

The main deliverables of the thesis can be grouped as follows:

- Comparative analysis of the existing methods for solution of the single area and multi-area economic emission dispatch problem.

- Mathematical formulation of the economic dispatch problems
  - Formulation of the CEED problem using quadratic fuel cost function.
  - Development of the 'Min-Max' and 'Max-Min' price penalty factors for CEED problem in addition to the other existing penalty factors.
  - Formulation of the CEED problem using quadratic fuel cost function with valve point loading effect.
  - Formulation of the CEED problem using cubic fuel cost function.
- Optimisation methods developed to solve the economic emission dispatch problems
  - Development of the Lagrange's methods for solution of the CEED problems.
  - Development of the PSO method for solution of the CEED problem.
  - Development of the Lagrange's decomposition-coordinating method for solution of the MAEED problem.
- Software developed and implemented for solution of the single and multi-area economic emission dispatch problems
  - Development of software based on Lagrange's and PSO methods and algorithms to solve the CEED problems in a sequential and a parallel ways.
  - Development of software based on the Lagrange's decomposition-coordinating method to solve the MAEED problem for a sequential and a parallel ways.
  - Application of the developed algorithms and software for solution of various types of dispatch problems.

Detail description of the thesis deliverables is given in chapter 8.

1.9 Chapter breakdown

This thesis has eight chapters as follows:

Chapter 1 presents the problem statement and background of the thesis. The thesis statement includes the research aim and objectives, statement of the problem, hypothesis, research delimitation, and assumptions.

Chapter 2 describes the literature review of single and multi-area economic dispatch problems. It analyses the specific characteristics of various classical, heuristic, metaheuristic and hybrid algorithms for solution of the economic emission dispatch problem.

Chapter 3 describes the developed Lagrange's algorithms for single area economic emission dispatch problem solution. The Lagrange's methods and algorithms are developed for the following cases:

- CEED problem using quadratic cost function
- CEED problem using quadratic cost function with valve point effect loading
- CEED problem using cubic cost function

Various IEEE benchmark models are used to test the algorithms in Matlab environment.

Chapter 4 presents introduction to the Particle Swarm Optimisation (PSO) algorithm and describes the developed PSO algorithm for the CEED problem. The various IEEE benchmark models are used to test the algorithm in Matlab environment.

Chapter 5 firstly elaborates on the reason why there is a need in applying parallel computing in Economic Dispatch problem solution. It also explains the concept of parallel computing and gives a brief description of the parallel Matlab environment. Matlab Parallel Computing Toolbox and Matlab Distributed Computing Server and their concepts are introduced.

Chapter 6 describes the implementation of the solution of the single area CEED problem in a data-parallel way using the two developed Lagrange's and PSO methods and algorithms, which are described in Chapter 3 and Chapter 4 respectively.

Chapter 7 describes the introduction of the multi-area economic dispatch problem and its necessity. The Lagrange's decomposition-coordinating method and algorithm is developed for the large interconnected power system. The software is developed for the calculation of the multi-area dispatch problem in decomposition way and is tested with two IEEE models: i) Four area forty generator power system and ii) four area three generator power systems.

Chapter 8 presents the conclusion, and the deliverables of the thesis. The applications of the thesis results, the future work, and the list of author publications complete the chapter.

1.10 Conclusion

This chapter presents the problem statement, gives the background of the project. The project statement includes the research aim, problem statement, objectives, and assumptions. Outline and deliverables of the thesis are stated.

In order to understand the depth of the problem it is critical to review literature specifically looking at the papers that propose methods and algorithms for solution of single or multi-area economic dispatch problems using classical, heuristic or meta heuristic optimisation methods.

Chapter 2 outlines the existing literature on methods used for economic emission dispatch problem solution.

CHAPTER TWO

INVESTIGATION OF THE METHODS FOR SINGLE AREA AND MULTI AREA OPTIMISATION OF A POWER SYSTEM DISPATCH PROBLEM

2.1 Introduction

In operating the power system for any load condition the contribution from each generating plant and from each unit within a plant must be determined so that the cost of the delivered power is a minimum. Any plant may contain different units such as hydro, thermal, gas etc. These plants have different characteristic which gives different generating cost at any load. So there should be a proper scheduling of plants for the minimization of cost of the operation. The cost characteristic of each generating unit is also non-linear function of the produced active power. So the problem of achieving the minimum cost becomes a non-linear optimisation problem which is difficult to be solved. The optimum load dispatch problem involves the solution of two different problems. The first of these is the unit commitment or pre-dispatch problem wherein it is required to select the optimal power of the available generating sources in order to meet the expected load and provide a specified margin of operating reserve over a specified period of time .The second aspect of economic dispatch is the real-time economic dispatch whereas it is required to distribute load among the generating units actually in parallel with the system in such a manner as to minimize the total cost of supplying the minute to minute demand of the system.

The objective of this chapter is to review the single area and multi-area economic dispatch problems and the existing methods and algorithms for solution of these dispatch problems. The review divides the economic dispatch problem into single area and multi-area as shown in Figure 1, and is done for a time period of 55 years from 1958 to 2013. The total number of 182 publications is considered. The number of publications year wise and algorithm wise is shown in the Table 2.1 and its graphical representation is shown in Figure 2.1 and Figure 2.2 respectively.

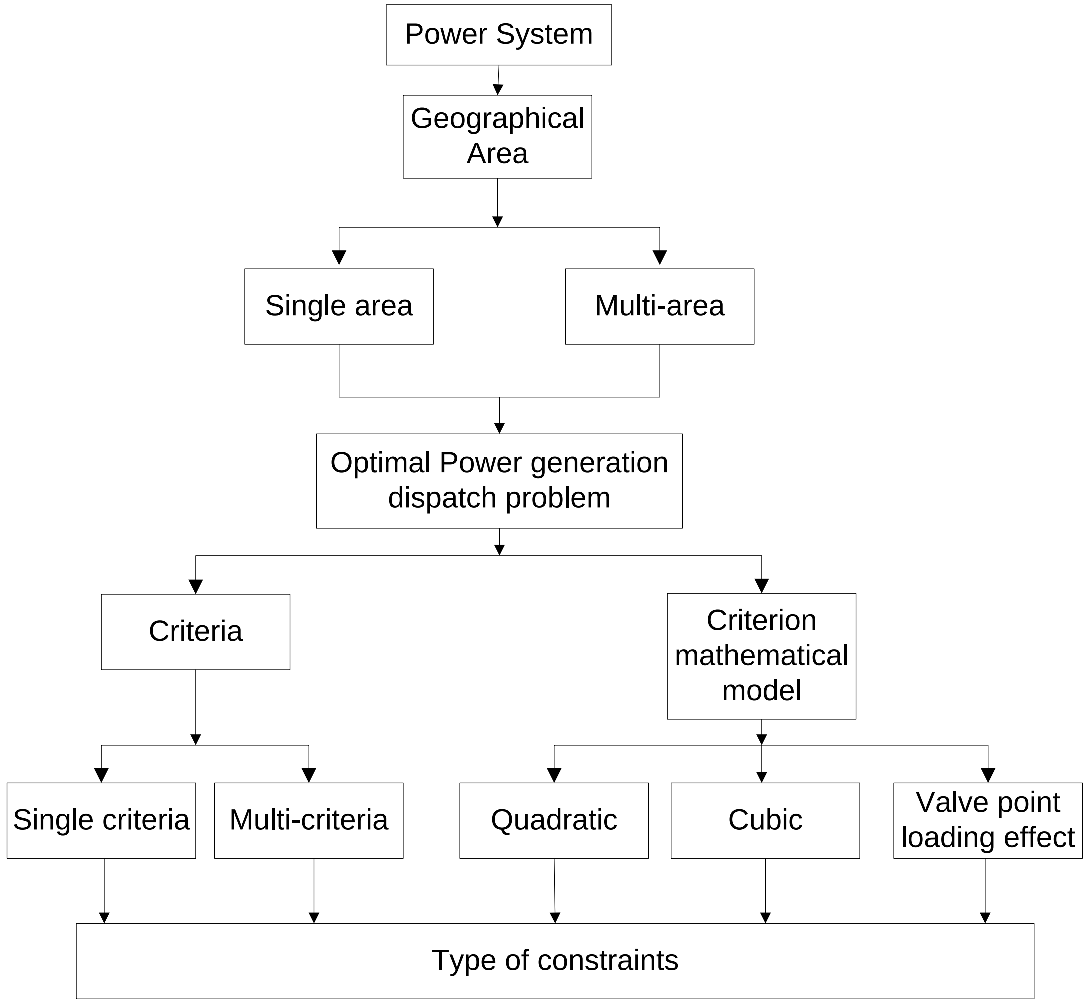


Figure 2.1: Classification of Single area and Multi-area power system dispatch Problem

Table 2.1: Number of publications year wise

NUMBER OF PUBLICATIONS - YEAR WISE		
Reference	**Year of publication**	**No. of publication**
(Gerald and Aaron, 1958) and (Leon, 1958)	1958	2
(Amdahl, 1967)	1967	1
(Billinton and Jain, 1972)	1972	1
(Happ,1977)	1977	1
(Quintana et al., 1981)	1981	1
(Jamshidian et al., 1983)	1983	1
(Billinton and Allan, 1984)	1984	1
(Helmick and Shoults, 1985)	1985	1
(Billiton and Chowdhury, 1988), and (Gustafson,1988)	1988	2
(Gottlieb and Almasi, 1989)	1989	1

(Chowdhury and Rahman, 1990), and (Chawdhury and Billinton, 1990)	1990	2
(Ouyang and Shahidehpour, 1991) and (Srikrishna and Palanichamy,1991)	1991	2
(Lee and Feng, 1992), (Lin et al.,1992), (Message-passing interface report,1992) ,and (Wang and Shahidehpour, 1992)	1992	4
(Dhillon et al., 1993), and (Wong et al, 1993)	1993	2
(Dhillon et al, 1994) and (Nanda et al, 1994)	1994	2
(Derose et al., 1995), (Eberhant and Kennedy,1995), (Kennedy and Eberhart, 1995), (Moler, 1995), (Okada and Asano,1995), (Rong-Mow and Nanming, 1995), and (Steriffert, 1995),	1995	7
(Available Transfer Capability, NERC Report, 1996), (Hollingsworth, et al., 1996), (Storn and Price, 1996) , and (Wood and Wollenberg, 1996)	1996	4
(David and Singh, 1997) and (Kennedy and Eberhart, 1997)	1997	2
(Lagarias et al., 1988) and (Tseng et al., 1998)	1998	2
(Ching-Tzong, et al., 1999), (David, et al., 1999), and (Hadi Saadat, 1999)	1999	3
(John, 2000) , (Kulkarni,et al., 2000), and (Poli et al., 2000)	2000	3
(Chen and Chen, 2001), (El-Gallad et al., 2001), (Ray and Liew, 2001), (Shi and Eberhart, 2001), (Yalcionoz et al.,2001), and (Zhu and Momoh, 2001)	2001	6
(Behrooz, 2002), (Changhun, 2002), (Hu and Eberhart, 2002), and (Parsopoulos and Vrahatis, 2002)	2002	4
(Gaing, 2003), (Nidul et al, 2003), and (Venkatesh et al, 2003)	2003	3
(Gilat,2004), and (Guerrero,2004), (Gnanadass et al., 2004), (Kannan and Veilumuthu, 2004), and (Kumar et al., 2004)	2004	5
(Chiang,2005), (Danaraj and Gajendran, 2005) , (Gnanadass, 2005), (Jong-Bae et al, 2005), (Lai et al., 2005), (Nithiyananthan and Ramachandran, 2005), (Park et al., 2005), and (Venayagamoorthy,2005)	2005	8
(Abhijit and Suntia, 2006), (Abido, 2006), (Adhinarayanan and Sydulu, 2006), (Chaturvedi et al., 2006), (Coelho and Mariani,2006), (Hong-Shan et al., 2006), (Jeyakumar et al., 2006), (Koh et al., 2006), (Mishra et al., 2006), (Panuganti et al.,2006), (Rani et al., 2006), (Thakur, 2006), (Wang and Singh, 2006), and (Zwe-Lee and Rung-Fang, 2006)	2006	14
(Adhinarayanan, 2007), (Balamurugan and Subramanian, 2007), (Chayakulkeree and Ongsakul, 2007), (Deb et al., 2007), (Dos Santos and Mariani, 2007), (Immanuel and Thanushkodi, 2007), (Jeyakumar et al., 2007), (Kim et al.,2007), (Lee et al., 2007), (Panta, 2007), (Ramesh and Ramachandran, 2007), (Ruchir et al, 2007), (Saber et al., 2007), (Yang et al., 2007), and (Zarei et al., 2007)	2007	15
(Al-Sumait et al., 2008), (Baskar and Mohan, 2008), (Duncan et al.,2008), (Hemamalini and Simon, 2008), (Jayabarathi et al., 2008), (Manikandan et al 2008b), (Manikandan et al., 2008a), (Manoharan et al., 2008), (Nasr Azadani et al., 2008), (Palanichamy and Babu,2008), (Senjyu et al., 2008), (Shubham et al., 2008), (Sugsakaran and Damrongkulkamjorn, 2008), (Venkatesh and Lee, 2008), and (Yingvivatanapong et al., 2008)	2008	15
(Amita et al.,2009), (Anurag and Noel,2009), (Chen and Wang, 2009), (Hemamalini and Simon, 2009), (Karthikeyan et al., 2009), (Mahor et al, 2009), (Muneender and Kumar, 2009), (Ozyon et al., 2009), (Prasanna and Somasundaram, 2009), (Ravikumar et al, 2009), (Roman and Peter, 2009), (Tao and Jin-ding, 2009), (Wang and Singh, 2009), and (YU, et al., 2009)	2009	14

(Adhinarayanan and Sydulu, 2010), (Alsumait et al., 2010), (Christos and George, 2010), (Ganesan and Subramanian, 2010), (Hemamalini and Simon, 2010), (Jong-Bae et al., 2010), (Khamsawang et al., 2010), (Kothari and Dhillon, 2010), (Meng et al., 2010), (Ming et al., 2010), (Pankaj, 2010), (Park et al., 2010), (Ratniyomchai et al., 2010), (Sharma et al., 2010) , (Subramanian and Ganesan, 2010),(Tiacharoen et al., 2010), (Tsai and Yen, 2010), (Yee Ming and Wen-Shiang, 2010), and (Yu and Hang, 2010)	2010	19
(Ahmad and Mortazavi 2011), (Barzegari et al, 2011), (Demirovic and Tesnjak, 2011), (Hosseinnezhad et al., 2011), (Krishnamurthy and Tzoneva, 2011a), (Krishnamurthy and Tzoneva, 2011b), (Ganesan and Subramanian, 2011), (MATLAB® Distributed Computing Server™ 5 Installation Guide, 2011), (Rani and Afshin, 2011),(Somasundaram and Jothi Swaroopan, 2011), and (Yu and Chung, 2011)	2011	11
(Ciornei and Kyriakides, 2012), (David and John, 2012), (Fragomeni, 2012), (Jabr, 2012), (Krishnamutrhy and Tzoneva, 2012a), (Krishnamutrhy and Tzoneva, 2012b), (Krishnamutrhy and Tzoneva, 2012c), (Krishnamutrhy and Tzoneva, 2012d), (Krishnamutrhy and Tzoneva, 2012e), and (Rashuraman etal., 2012)	2012	10
(Abimbola et al., 2013), (Ali, 2013), (Arul et al., 2013),(Barisal, 2013),(Basu, 2013), (Chuco,2013), (Hosseinnezhad and Ebrahim, 2013), (Krishnamurthy and Tzoneva, 2013a), (Krishnamurthy and Tzoneva, 2013b), (Mondal, etal., 2013), (Ozyon and Aydin, 2013), (Wang and Li, 2013), and (Xingwen et al., 2013).	2013	13

Table 2.2: Most used methods and algorithms for Economic Dispatch Problem

Most used methods and algorithms		Reference paper	No. of publication
Algorithm	Abbreviation		
Artificial Neural Network	ANN	(Basu, 2013), (Chaturvedi etal., 2006), (Kulkarni etal., 2000), (Mishra, et al., 2006) and (Ozyon and Aydin, 2013)	5
Bees Algorithm	BA	(Tiacharoen et al., 2010)	1
Bi-criterion Global Optimization	BGO	(Wong et al., 1993)	1
Component Model	CM	(Kannan and Veilumuthu, 2004)	1
Classical Technique	CT	(Nanda et al., 1994)	1
Commitment Utilization Factor	CUF	(Lee and Feng, 1992)	1
Differential Evolution	DE	(Coelho and Mariani, 2006), (Guerrero, 2004), (Jayabarathi et al., 2008), (Storn and Price, 1996), and (Xingwen, et al., 2013)	5
Dynamic Programming	DP	(Balamurugan and Subramanian, 2008), and (Barisal, 2013)	2
Direct Search	DS	(Chen an Chen, 2001),(Fragomeni,2012), (Ganesan and Subraminan, 2011), and (Zarei et al., 2007)	4
Dantzing Wolfe Decomposition Principle	DWDP	(Quintana et al., 1981)	1
E-constraint Method	EM	(Dhillon et al., 19994)	1
Evolutionary Programming	EP	(Abido, 2006) , (Deb et al., 2007), (Gnanadass et al., 2004), (Jeyakumar et al., 2007), (Manoharan et al., 2008), (Nidul and Chattopadhyay, 2003), (Prasanna and Somasundaram, 2009), (Sharma et al., 2010), (Venkatesh and Lee, 2008) , and (Venkatesh, et al., 2003)	10

Expert System	ES	(Wang and Shahidehpour, 1992)	1
Fuzzy Logic	FL	(Ali, 2013) , (Chayakulkeree and Ongsakul, 2007) and (Shubham, et al.,2008)	3
Genetic Algorithm	GA	(Chiang, 2005), (Ciornei and Kyriakides, 2012), (Ozyon et al., 2009), and (Yalcionoz et al., 2001)	4
General Algebraic Modeling System	GAMS	(Demirovic and Tesnjak, 2011)	1
Gravitational Search Algorithm	GSA	(Mondal, 2013)	1
Grid Service Model	GSM	(Ramesh and Ramachandran, 2007)	1
Heuristic Approach	HA	(Ouyang and Shahidehpour, 1991)	1
Harmony Search	HS	(Arul, etal., 2013), (Ravikumar Pandi et al., 2009), and (Wang and Li, 2013)	3
Jacobin Method	JM	(Jamshidian et al., 1983)	1
Lagrange's Algorithm	LA	(Danaraj and Gajendran, 2005), (Hemamalini and Simon, 2009), (Hemamalini and Simon, 2010), (Krishnamurthy and Tzoneva, 2011a), (Krishnamurthy and Tzoneva, 2011b), (Krishnamurthy and Tzoneva, 2012a), (Krishnamurthy and Tzoneva, 2013), (Okada and Asano, 1995), (Yingvivatanapong et al., 2008), and (Yu and Hang, 2010)	10
Lambda Iteration	LI	(Palanichamy and Babu, 2008), and (Panta et al., 2007)	2
Network Flow Modeling	NFM	(Streiffert, 1995), and (Zhu and Momoh, 2001)	2
Netwon Raphson	NR	(Lin, et al., 1992), and (Rong-Mow and Nanming, 1995)	2
Pattern Search	PS	(Al-Sumait, et al., 2010), and (Al-Sumait, et al., 2008)	2
Particle Swarm Optimization	PSO	(Adhinarayanan and Sydulu, 2006), (Amita, etal., 2009), (Barzegari et al., 2011), (Baskar and Mohan, 2008), (Dos Santos and Mariani, 2007), (Eberhant, and Kennedy, 1995) , (El-Gallad et al, 2001),(Gaing, 2003),(Hemamalini and Sishaj , 2008), (Hosseinnezhad and Ebrahim, 2013),(Hosseinnezhad et al., 2011), (Hu and Eberhart, 2002), (Immanuel and Thanushkodi, 2007), (Jeyakumar et al., 2006), (Jong-Bae, et al., 2005), (Jong-Bae, et al.,2010),(Kennedy and Ebehart, 1997),(Khamsawang et al., 2010), (Kim et al., 2007), (Koh et al., 2006),(Kothari, and Dhillon, 2010), (Krishnamurth and Tzoneva, 2012), (Lai et al., 2005), (Lee et al., 2007), (Mahor et al., 2009), (Meng et al., 2010), (Ming et al., 2010), (Muneender and Kumar, 2009), (Nasr Azadani, 2008), (Park et al., 2005), (Park et al., 2010), (Parsopoulos and Vrahatis, 2002),(Poli et al., 2000), (Rani al., 2006), (Rani and Afshin, 2011), (Ray and Liew, 2001),(Saber et al., 2007), (Senjyu et al., 2008), (Shi and Eberhart, 2001),(Shubham, et al.,2008), (Somasundaram and Jothi , 2011), (Sugsakarn and Damrongkulkamjorn, 2008), (Tao and Jin-ding, 2009), (Thakur et al., 2006), (Tsai and Yen, 2010),(Wang and Singh, 2006), (Wang and Singh, 2009), (Yang et al., 2007), (Yee and Wen-Shiang, 2010), and (Yu and Chung, 2011)	51
Power Transfer Distribution Factor	PTDF	(Manikandan et al., 2008a), and (Manikandan et al., 2008b)	2

Quadratic Programming	QP	(Danaraj and Gajendran, 2005)	1
Remote Method Invocation	RMI	(Nithiyananthan and Ramachandran, 2005)	1
Sequential Approach with Matrix Framework	SAMF	(Subramanian and Ganesan, 2010)	1
Semi Definite Programming	SDP	(Abimbola, et al., 2013) and (Jabr, 2012)	2
Tabu Search	TS	(Prasanna and Somasundaram, 2009)	1
Unit commitment Approach	UCA	(Karthikeyan et al., 2009)	1

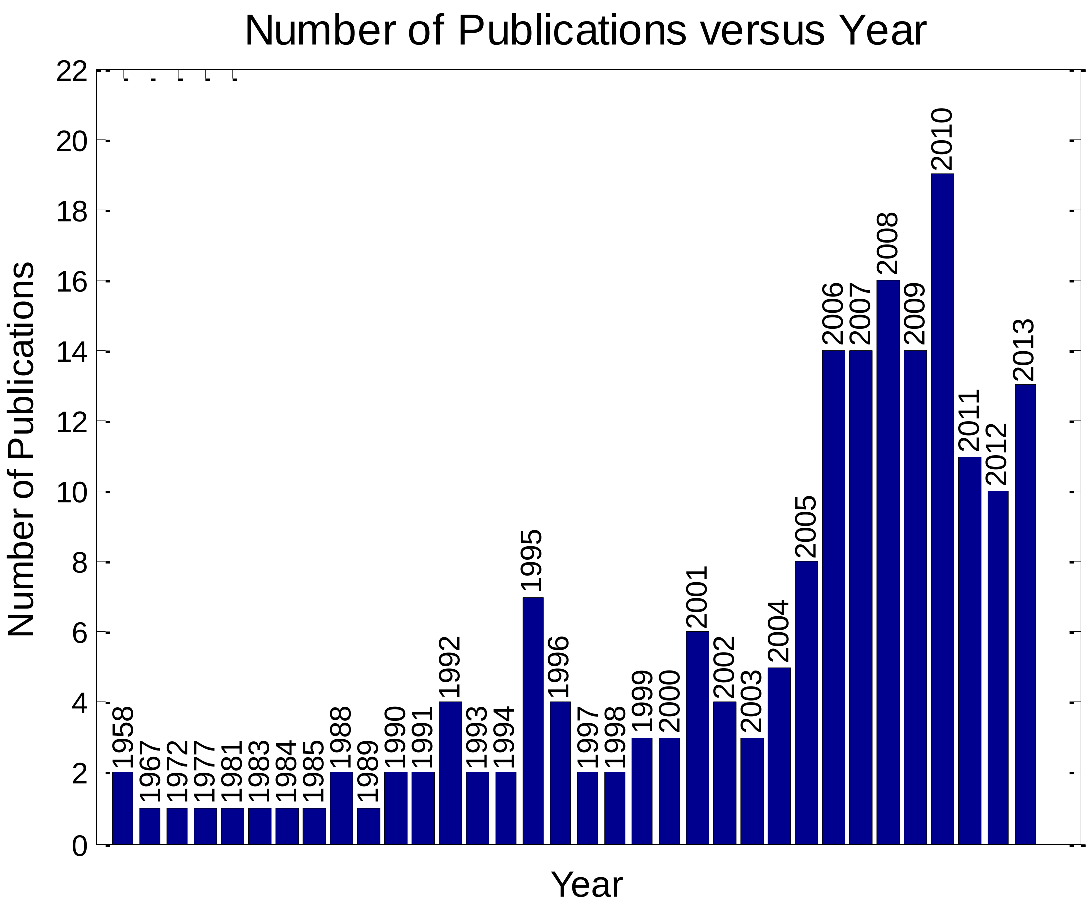


Figure 2.2: **Number of Publications Year Wise**

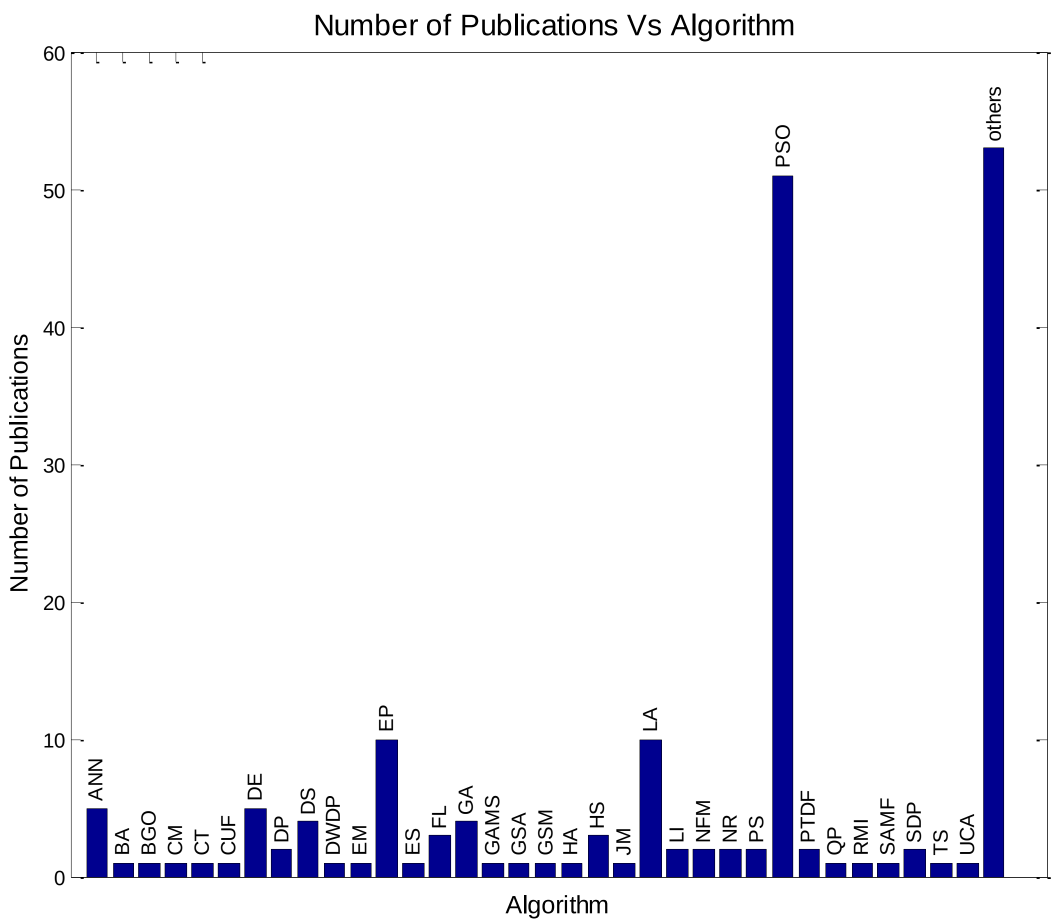


Figure 2.3: **Number of Publications Algorithm Wise**

2.2 Economic power dispatch problem formulation

The objective of solving the economic dispatch problem in electric power system is to determine the generation levels for all on-line units which minimize the total fuel cost and minimize the emission level of the system, while satisfying a set of constraints.

In this chapter the economic dispatch problem formulation is classified into two types as follows:

1. Single area economic dispatch problem
2. Multi-area dispatch problem

Every type can be formulated using a single criterion or multi-criteria and various types of constraints.

2.2.1 Single area economic dispatch problem

2.2.1.1 Single criterion dispatch problem

The objective of economic dispatch is to simultaneously minimize the generation cost and to meet the load demand of a power system over some appropriate period of time while satisfying various operating constraints. The objective function of an economic dispatch problem can be formulated as, (Happ, 1977)

Minimize

$$F_C = \sum_{i=1}^{n} F_i(P_i) = \sum_{i=1}^{n} \left(a_i P_i^2 + b_i P_i + c_i \right) \qquad [\$/h] \qquad (2.1)$$

Where

F_C	Total Fuel Cost
$F_i(P_i)$	Fuel cost of the i^{th} generator
P_i	Real power generation of unit i
a_i , b_i, c_i	Cost coefficients of generating unit i
n	Number of generating units

Under the Constraints

1) Power balance constraint

$$\sum_{i=1}^{n} P_i = P_G = P_D + P_L \qquad [MW] \qquad (2.2)$$

Where

P_G	Total power generation of the system
P_D	Total demand of the system
P_L	Total transmission loss of the system

The transmission loss can be expressed as, (Kothari and Dhillon, 2010)

$$P_L = \sum_{i=1}^{n} \sum_{j=1}^{n} P_i B_{ij} P_j + \sum_{i=1}^{n} B_{0i} P_i + B_{00} \qquad [MW] \qquad (2.3)$$

where

P_j	Real power generation of the generation unit j
B_{ij}, B_{0i}, B_{oo}	Transmission loss coefficients

2) Generator operational constraints

$$P_{i,\min} \le P_i \le P_{i,\max} \qquad [MW] \qquad (2.4)$$

Where

$P_{i,min}$ Minimum value of the real power allowed at generator *i*

$P_{i,max}$ Maximum value of the real power allowed at generator *i*

2.2.1.2 Multi-criteria dispatch problem

The various pollutants like sulphur dioxide, nitrogen oxide and carbon dioxide are released as a result of operation of the thermal power plants. Reduction of these pollutants is compulsory for every generating unit. To achieve this goal new criteria are included in formulation of the economic dispatch problem as follows:(Chayakulkeree and Ongsakul, 2007)

$$E_{SO_2} = \sum_{i=1}^{n} \left(a_{SO_2 i} P_i^2 + b_{SO_2 i} P_i + c_{SO_2 i} \right) \quad \text{[Kg/h]} \tag{2.5}$$

$$E_{NO_x} = \sum_{i=1}^{n} \left(a_{NO_x i} P_i^2 + b_{NO_x i} P_i + c_{NO_x i} \right) \quad \text{[Kg/h]} \tag{2.6}$$

$$E_{CO_2} = \sum_{i=1}^{n} \left(a_{CO_2 i} P_i^2 + b_{CO_2 i} P_i + c_{CO_2 i} \right) \quad \text{[Kg/h]} \tag{2.7}$$

$$E_T = E_{SO_2} + E_{NO_x} + E_{CO_2} \quad \text{[Kg/h]} \tag{2.8}$$

For a single pollutant it can be written as

$$E_T = \sum_{i=1}^{n} \left(d_i P_i^2 + e_i P_i + f_i \right) \quad \text{[Kg/h]} \tag{2.9}$$

Where

$E_{SO2}, E_{NOx}, E_{CO2}$ SO_2, NO_x, CO_2 emissions respectively

E_T Total emission

d_i, e_i, f_i Emission coefficients of generating unit *i*

a_{SO2i}, b_{SO2i}, c_{SO2i} SO_2 Emission coefficients of generating unit *i*

a_{NOxi}, b_{NOxi}, c_{NOxi} NO_x Emission coefficients of generating unit *i*

a_{CO2i}, b_{CO2i}, c_{CO2i} CO_2 Emission coefficients of generating unit *i*

A multi-objective optimization is converted into a single objective optimization one called Combined Economic Emission Dispatch (CEED) problem by introducing price penalty factor h_i to the various pollutants (Chayakulkeree, 2011). Three bi-criterion problems can be formulated for every pollutant separately as follows

$$F_{TSO_2} = \sum_{i=1}^{n} \left[(a_i P_i^2 + b_i P_i + c_i) + h_{SO_2 i} \left(a_{so_2 i} P_i^2 + b_{so_2 i} P_i + c_{so_2 i} \right) \right] \quad \text{[\$/h]} \tag{2.10}$$

$$F_{TNOx} = \sum_{i=1}^{n} \left[(a_i P_i^2 + b_i P_i + c_i) + h_{NOxi} \left(a_{NOxi} P_i^2 + b_{NOxi} P_i + c_{NOxi} \right) \right] \quad \text{[\$/h]} \tag{2.11}$$

$$F_{TCO_2} = \sum_{i=1}^{n}\left[(a_iP_i{}^2+b_iP_i+c_i)+h_{CO_2i}\left(a_{Co_2i}P_i^2+b_{cO_{2i}}P_i+c_{co_2i}\right)\right] \quad [\$/h] \qquad (2.12)$$

Where

F_{TSO2} CEED's fuel cost of SO_2 emission

F_{TNOx} CEED's fuel cost of NO_x emission

F_{TCO2} CEED's fuel cost of CO_2 emission

In the cases where the impact of all 3 pollutants is important, the multi-objective dispatch problem can be solved for all pollutants. Then the CEED fuel cost for the SO_2, NOx, and CO_2 emissions is given by the equation (2.9)

$$F_T = \sum_{i=1}^{n}\left[\begin{array}{l}(a_iP_i^2+b_iP_i+c_i)+h_{SO_2i}\left(a_{so_2i}P_i+b_{so_2i}P_i+c_{so_2i}\right)+\\ +h_{NOxi}\left(a_{NOxi}P_i^2+b_{_{NOxi}}P_i+c_{NOxi}\right)+h_{CO_2i}\left(a_{Co_2i}P_i^2+b_{cO_{2i}}P_i+c_{co_2i}\right)\end{array}\right] [\$/h] \quad (2.13)$$

The role of price penalty factor is to transfer the physical meaning of emission criterion from weight of the emission to the fuel cost for the emission. The difference between these penalty factors is in the weight of the fuel cost for emission in the final optimal fuel cost for generation and emission. The price penalty factor for multi-objective dispatch problem (Subramanian and Ganesan, 2010) is formulated taking the ratio between the maximum fuel cost and maximum emission of the corresponding generators as follows:

$$h_{SO_2i} = \frac{(a_iP_{i\,\max}^2+b_iP_{i\max}+c_i)}{(a_{so_2i}P_{i\max}^2+b_{_{so_2i}}P_{i\max}+c_{so_2i})} \quad [\$/kg] \qquad (2.14)$$

$$h_{NOxi} = \frac{(a_iP_{i\max}^2+b_iP_{i\max}+c_i)}{(a_{NOxi}P_{i\max}^2+b_{NOxi}P_{i\max}+c_{NOxi})} \quad [\$/kg] \qquad (2.15)$$

$$h_{CO_2i} = \frac{(a_iP_{i\max}^2+b_iP_{i\max}+c_i)}{(a_{Co_2i}P_{i\max}^2+b_{CO_2i}P_{i\max}+c_{Co_2i})} \quad [\$/kg] \qquad (2.16)$$

Where

h_{SO_2i} Max-Max price penalty factor of SO_2 emission

h_{NOxi} Max-Max price penalty factor of NO_x emission

h_{CO_2i} Max-Max price penalty factor of CO_2 emission

(Balamurugan and Subramanian, 2008) used "Min-Min" and "Max-Max" to calculate the average and common price penalty factors to solve the multi-objective economic emission dispatch problem which is given by the Equations (2.17) to (2.20).

$$h_{(\max)} = \frac{(a_i P_{i\max}^2 + b_i P_{i\max} + c_i)}{(d_i P_{i\max}^2 + e_i P_{i\max} + f_i)}, \quad i = 1,2,..,n \qquad (2.17)$$

$$h_{(\min)} = \frac{(a_i P_{i\min}^2 + b_i P_{i\min} + c_i)}{(d_i P_{i\min}^2 + e_i P_{i\min} + f_i)}, \quad i = 1,2,..,n \qquad (2.18)$$

$$h_{(avg)} = \frac{h_{(\max)} + h_{(\min)}}{2}, \quad i = 1,2,..,n \qquad (2.19)$$

$$h_{com} = \frac{\sum_{i=1}^{n} h_{(avg)}}{n} \qquad (2.20)$$

The problem (2.1), (2.9), (2.13), subject to the constraints (2.2), (2.3), (2.4) and various penalty factors (2.17) to (2.20) are solved in the literature by different optimisation methods.

2.2.2 Multi-area dispatch problem

2.2.2.1 Introduction of Multi-area Economic Emission dispatch problem

The multi-area dispatch problem dispatched the generator power within multiple areas. Every area has its own set of generators (Chen and Wang, 2010). The areas are interconnected by tie-lines as shown in Figure 2.4

Single criterion or multicriteria dispatch problems can be formulated for every area and for the whole system. They determine the optimal power to be produced by the generators in every area and the optimal values of the power transferred between the areas through the tie-lines.

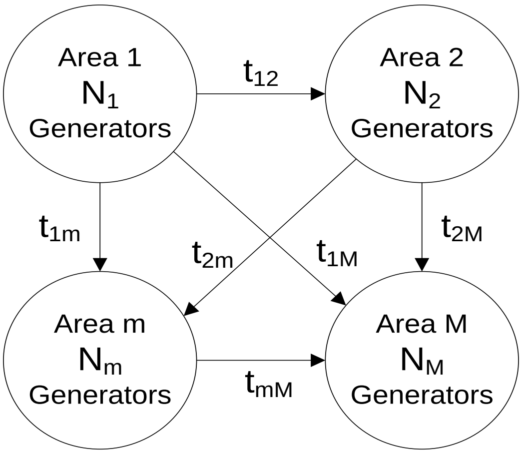


Figure 2.4: **Model of a Multi-area power system with tie-line power transfer**

2.2.2.2 Optimal power generation and transmission cost criterion for every area

The objective of multi-area dispatch is to determine the generation levels in every area and the interchange power between the areas that minimize both fuel and emission costs while satisfying a set of constraints as: (Chen and Wang, 2010) and (Wang and Singh, 2009)

$$\min \sum_{m=1}^{M}\left[F_m^1, F_m^2\right] = \begin{bmatrix} \sum_{m=1}^{M}\sum_{i=1}^{n}\left(a_{mi}P_{mi}^2 + b_{mi}P_{mi} + c_{mi}\right) + \\ + \sum_{m=1}^{M}\left[h_m \sum_{i=1}^{n}\left(\alpha_{mi}P_{mi}^2 + \beta_{mi}P_{mi} + \gamma_{mi}\right)\right] \end{bmatrix} \tag{2.21}$$

Where

F_m^1	Expected fuel cost for the m^{th} area
F_m^2	Expected *NOx* emission for the m^{th} area
n	Number of on-line generators for the area m in a M area system
a_{mi}, b_{mi}, c_{mi}	Fuel cost coefficients for the i^{th} generator in the m^{th} area
$\alpha_{mi}, \beta_{mi}, \gamma_{mi}$	Emission coefficients of the generator i in the m^{th} area
P_{mi}	Power output of the generator i in the area m
h_m	Price penalty factor in the area m

The constraints involved with the problem formulation are as follows: (Chen and Wang, 2010)

a) Area demand balance

In the area m, the total power generation must cover the local area demand $P_{demand,m}$ and the transmission loss $P_{loss,m}$ with the consideration of imported and exported power. This relationship can be expressed as: (Chen and Wang, 2010)

$$\sum_{i=1}^{n}P_{mi} - \sum_{\substack{m=1\\ m\neq k}}^{M}\left[t_{mk} - \left(1-\rho_{km}\right)t_{km}\right] - P_{demand,m} - P_{loss,m} = 0 \tag{2.22}$$

b) Area generation capacity

$$P_{mi}^{\min} \le P_{mi} \le P_{mi}^{\max}, \ m = \overline{1,M} \tag{2.23}$$

c) Tie-line capacity limits

$$t_{km}^{\min} \le t_{km} \le t_{km}^{\max}\ ,\ \ m = \overline{1,M}\ ,\ k = \overline{1,M} \quad k \ne m \qquad (2.24)$$

Where

t_{km}	Economic tie-line transfer real power
ρ_{km}	Tie-line transfer loss ratio from area k to area m
$P_{demand,m}$	Local demand in the area m
$P_{loss,m}$	Transmission loss for the area m
$t_{km}^{\min}$ and $t_{km}^{\max}$	Tie-line minimum and maximum capacity limits from the area k to the area m
$P_{mi}^{\min}$ and $P_{mi}^{\max}$	Minimum and maximum power output of the generator i in the area m

2.3 Investigation of the single area power system dispatch problem methods for solution

2.3.1 Single criterion Problem

This section investigates the single area power system dispatch problem according to various optimization techniques used, such as Lagrange's Algorithm (LA), Improved Harmony Search Algorithm (IHSA), Bees Algorithm (BA), General Algebraic Modeling Systems (GAMS), Artificial Neural Network (ANN), Evolutionary Algorithm (EA), and Particle Swarm Optimization (PSO).

(Happ, 1977) represented a comprehensive survey of papers on optimal dispatch problem covering the period of 1920-1977. In that, the existing literature of the economic dispatch into the following main groups: classical single area, multi area economic dispatch, and optimal load flow.

Classic economic dispatch of real power treats the network approximately and optimizes the generator powers when all voltages are considered constant with objective being the minimization of production costs. The classic economic dispatch was compared with the rigorous power allocation method, and it was concluded that there were no economic incentives in terms of dollars saved in order to switch to a more rigorous economic dispatch procedure.

Rigorous procedures compute the incremental losses directly from the Jacobian of the Newton Raphson load flow. The work on valve point loading and environmental dispatch is also considered in his review.

According to survey done by (Chowdhury and Rahman, 1990) for the period 1977-1988, the economic dispatch is divided into four areas. They are optimal power flow, economic dispatch in relation to Automatic Generation Control (AGC), dynamic dispatch, and economic dispatch with non-conventional generation sources.

The optimal power flow procedure consists of methods utilizing load flow techniques for the purpose of economic dispatch. Some authors used AC load flow model and other used DC load flow model. The role of Automatic Generation Control (AGC) is to maintain desired megawatt output of a generator unit and control the system frequency. The AGC loop maintains control only during normal (small and slow) changes in load and frequency. Adequate control is not possible during emergency situations when large imbalance occurs. A dynamic economic dispatch is one that considers change related costs. With the use of steady-state operating costs in the static operation, poor transient behavior results when these solutions are incorporated in the feedback control of dynamic electric power networks. Non-conventional generation sources such as solar, photovoltaic, solar thermal, wind, geothermal, storage battery etc. are considered. The frequent weather changes may translate into extremely high variations in the power generations from the Non-conventional plants. If the plant is constantly connected to the distribution system, this causes operational problems like load following, spinning reserve requirements, load frequency excursions, system stability etc., which the conventional AGC is unable to handle.

Maclaurin series based Lagrangian method is used to solve the Dynamic Economic Dispatch (DED) problem with valve-point loading effect (Hemamalini and Simon, 2009). The sine term is approximated in (Hemamalini and Simon, 2010) so there will be an approximation error and the solution may not converge to an optimal value. So initialization factor is used to compensate the Maclaurin sine series expansion approximation and to minimize the error.

Harmony search (Wang and Li, 2013) and Improved Harmony Search (IHS) (Pandi et al., 2009) is a new meta-heuristic optimization algorithm which imitates the music improvisation process applied by musicians. The parameters of the IHS algorithm are the Harmony Memory Size (HMS), Harmony Memory Considering Rate (HMCR), Pitch Adjusting Rate (PAR), and the Number of Improvisations (NI). Some efficient meta-heuristic search methods (Ratniyomchai et al., 2010) like Genetic Algorithm, Evolutionary Programming, Adaptive Tabu Search, Particle Swarm Optimization and Improved Harmony Search are briefed and summarized. The improved harmony search method proves that it can find a place among some efficient meta-heuristic search methods in order to find a near global solution of the economic load dispatch problems.

In Bees algorithm, the colony of artificial bees contains two groups of bees: scout and employed bees. The scout bees have the responsibility to find a new food source. The responsibility of employed bees is to determine a food source within the neighborhood of the food source in their memory and share their information with other bees within the hive. Bees algorithm has been applied to solve the economic dispatch problem with prohibited, operating zones, ramp-rate limits, and non-smooth cost functions in (Tiacharoen, 2010).

The problems of economic dispatching and multi-periodic power flows are formulated and modeled using General Algebraic Modeling Systems (GAMS). GAMS language (Demirovic and Tesnjak, 2011) offers a series of algorithms that optimize both linear and nonlinear or mixed integer problems. GAMS program is used in order to solve problems, both in the domain of power systems optimization and in other engineering fields.

Artificial neural network are often used to find solution of the optimal power dispatch problem. The architecture of the feed-forward neural network trained by Levenberg-Marquardt back propagation algorithm (Chaturvedi et al., 2006) and (Basu, 2013), is shown in Figure 2.5. A large number of loading patterns were generated in wide range.

The total power demand in the power system is taken as the input variable for the proposed neural network. The output of the neural network provides the optimal value of the incremental cost of generation λ and the generation allocation at different generators so that the total cost of generation is minimum.

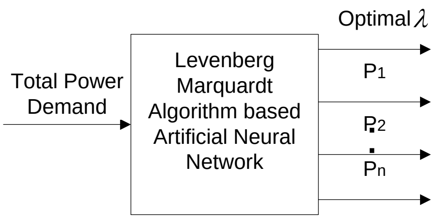


Figure 2.5: **ANN Architecture (Chaturvedi et al., 2006)**

Various architectures of three-layered feed-forward Artificial Neural Network (ANN) models having different number of hidden nodes have been trained for the same error goal and the optimal structure has been selected on the basis of the least training time. The trained neural network has been tested for the testing patterns to evaluate its performance. According to (Panta et al., 2007) the ANN with three layers, i.e. input layer, hidden layer and output layer shown in Figure 2.6 can be used for successful solution of the dispatch problem. The input layer has only one neuron which is the total power demand. The output layer has '*n*' neurons representing the generated powers of each generator.

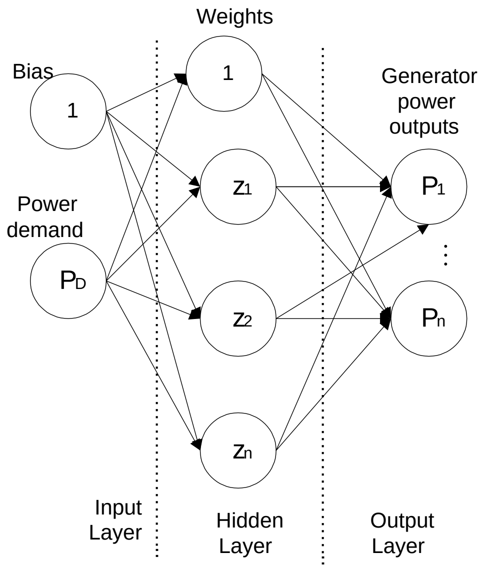


Figure 2.6: **ANN structure for economic power dispatch (Panta et al., 2007)**

The output of the input layer's neuron is connected to all neurons in the hidden layer through a weight. Also a bias signal is coupled to all the neurons through a weight. All the layers of a Neural Network have a hyper tangent sigmoid transfer function. The optimal power generated by each generator is calculated by using lambda iteration method. The neural network was trained using a MATLAB program.

Evolutionary algorithm is applied to find solution for a power network of combined cycle plants. In a Combined Cycle Co-generation Plant (CCCP), gas and steam turbines work in combination to generate electric power. In (Gnanadass et al., 2004), the economic dispatch problem is formulated using quadratic fuel cost function as in Equation (2.1) and CCCP fuel cost function is described below:

$$\sum_{i=1}^{n} F_i(P_i)_{CCCP} = \sum_{i=1}^{n}\left(b_i P_i + c_i\right)_{Linear} \quad [\$/hr] \tag{2.25}$$

Where

b_i and c_i Constants in [$/hr]

The main stages of Evolutionary Algorithm (EA) are initialization, creation of offspring vectors by mutation and competition and selection of best vectors to evaluate best fitness solution. Suppose the plant has *n* generators, the minimum solution was obtained by assuming the quadratic fuel cost characteristics for *(n-1)* generators and any one of the generator with CCCP. The fuel cost obtained due to CCCP effect is less than the quadratic fuel cost function. The transmission losses were computed using Newton-Raphson method in (Gnanadass et al., 2004).

Particle Swarm Optimization (PSO) was first introduced by Kennedy and Eberhart in 1995 (Kennedy and Eberhart, 1995). In (Manoharan et al., 2008), Evolutionary Algorithm (EA) such as the Real-Coded Genetic Algorithm (RCGA), PSO and Differential Evolution (DE) are considered. A constraint-handling method is employed which does not require any penalty parameter. In penalty parameter-less constraint-handling scheme, all feasible solutions have zero constraint violation and all infeasible solutions are evaluated according to their constraint violations alone. Hence, both the objective function value and constraint violation are not combined in any solution to the population. In general, there is no guarantee for the EA-based results, which are close to an optimal solution. This is a difficult question to answer for solving any arbitrary problem, not only using an evolutionary optimization technique, but also using any other optimization technique. A verification method based on Karush–Kuhn–Tucker (KKT) conditions to validate the solutions obtained by EAs is implemented in (Deb et al., 2007).

In (Lagarias, 1998), the economic dispatch problem formulation considers both valve-point and multiple fuel options. The optimal results obtained using various EAs are compared with Nelder–Mead simplex method. It reveals that PSO performs better in terms of solution quality and consistency in comparison to RCGA & DE. But DE performs better in terms of mean computation time in comparison to RCGA & PSO. Hybrid Differential Evolution approach is a simple population based stochastic function method and has been extended from the original algorithm of differential evolution (Storn and Price, 1996).

The Hybrid Differential Evolution (HDE) technique (Jayabarathi et al., 2008) is applied to various kinds of Economic Dispatch (ED) problems such as those including prohibited zones, emission dispatch, multiple fuels, and multiple areas. This method is used to solve unconstrained nonlinear, non-smooth, and non-differentiable optimization problems. The basic operations of HDE are: representation and initialization, mutation, crossover operation, selection and evaluation, accelerated operation, and migration operation. It can be observed that initially the PSO method appears to converge faster but after less than 20 iterations HDE performs better. The acceleration factor has reduced the convergence time while the introduction of the migration factor has improved the optimal result by avoiding the local optima.

A survey of particle swarm optimization applications in electric power systems to solve electric power optimization problems such as optimal power flow, economic dispatch, reactive power dispatch, unit commitment, generation and transmission planning, maintenance scheduling, state estimation, model identification, load forecasting, and control system is done in (Yang et al., 2007). PSO is classified into three types, as shown in Figure 2.7.

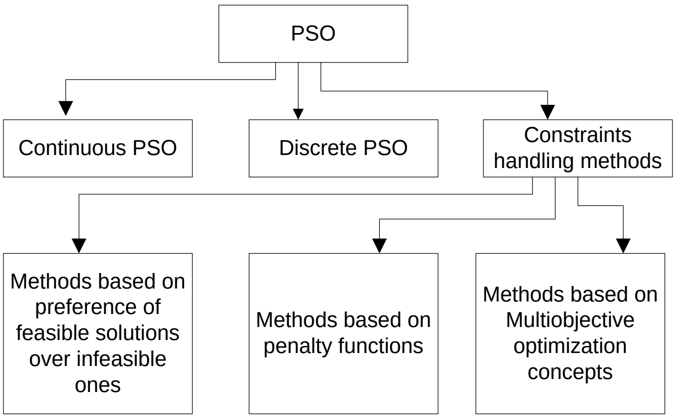


Figure 2.7: **Classification of PSO methods (Yang et al., 2007)**

Continuous particle swarm optimization is formulated as (Yang et al., 2007).

$$v_{id} = w\, v_{id} + c_1 rand().(p_{id} - x_{id}) + c_2 rand().(p_{nd} - x_{id}) \quad (2.26)$$

$$x_{id} = x_{id} + v_{id} \quad (2.27)$$

Where

v_{id}	New velocity
p_{id}	Particle location
p_{nd}	Best particles location among the neighbors
x_{id}	Particle position
w	Inertia weight
c_1 and c_2	Learning factors
rand ( )	Random numbers

Discrete particle swarm optimization (Kennedy and Eberhart, 1997), can be used to solve combinatorial problems such as unit commitment and generation scheduling. It is formulated as

$$v_{id} = w\, v_{id} + c_1 rand().(p_{id} - x_{id}) + c_2 rand().(p_{nd} - x_{id}) \quad (2.28)$$

$$\begin{aligned} &if\ (random() < S(v_{id})\ then\ x_{id} = 1 \\ &else\ x_{id} = 0 \end{aligned} \quad (2.29)$$

The values of

$p_{id,} p_{nd,} x_{id} \in (0,1)$

$v_{id} \in (0.0,1.0)$

rand ( ) Quasi random number in $\in (0.0,1.0)$

$S(v_{id})$ Sigmoid limiting transformation function

The constraints handling methods are used for many optimization problems in electric power systems. The Modification on basic PSO can be used to solve the constrained optimization problems. Method based on preference of feasible solutions over infeasible ones can be dealt in two ways. When a particle is outside the feasible space (El-Gallad et al., 2001), it will reset to the last best value found. When updating the memories, all particles keep feasible solutions in their memory and during the initialization process also all particles start with the feasible solutions (Hu and Eberhart, 2002). The initial feasible solution set is hard to be found and it is also difficult to deal with equity constraints.

The penalty function (Parsopoulos and Vrahatis, 2002) is used to convert the constrained optimization problem into unconstrained one. To find the optimal value of penalty function coefficients in optimization problems is difficult task. The inappropriate penalty coefficients can cause slow convergence or premature convergences of the algorithm.

A concept in multi-objective optimization (Ray and Leiw, 2001) is to generate a Better Performer List (BPL) based on the constraint matrix. An individual that is not in the BPL improves its performance by deriving information from its closest neighbor in the BPL.

A literature review of economic dispatch using PSO in (Mahor et al., 2009), pays an attention to local optimum solution, initial variables, premature and slow convergence, and dimensionality. It also reports that further improvements of PSO algorithms are required. The present versions of PSO have slower convergence at later stage and also are not able to provide optimal solution for real time scheduling problems. PSO method for solving the economic dispatch (ED) problem with nonlinear characteristics of the generator, such as ramp rate limits, prohibited operating zone, and non-smooth cost functions are considered in (Gaing, 2003). The solution based on the PSO method is compared with an elitist GA search method (Yalcionoz et al., 2001). It shows that PSO method avoids the shortcoming of premature convergence of GA method. PSO to solve economic dispatch of units with non-smooth input-output characteristic functions is developed in (Lai et al., 2005). The results of PSO solution is compared with Evolutionary Programming (EP) and it is concluded that increasing of the population size in an adaptive PSO algorithm can improve algorithm convergence characteristics.

Cubic cost function model (Adhinarayanan and Sydulu, 2006) represents the actual response of thermal generators more accurately. PSO algorithm is used to solve a dispatch problem with cubic fuel cost function

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 + d_i P_i^3 \tag{2.30}$$

Where

a_i, b_i, c_i, d_i Fuel cost coefficients of unit ***i***

The modified PSO (MPSO) algorithm is used to solve various types of economic dispatch problems in (Park et al., 2005), (Saber et al., 2007), (Baskar and Mohan, 2008), (Park et al., 2010). MPSO is applied to smooth cost functions and non-smooth cost functions considering valve-point effects and multi-fuel problems in (Park et al., 2005). The position of each individual (2.27) is modified by the Equation (2.31). The resulting position of an individual is not always guaranteed to satisfy the inequality constraints due to over/under velocity. If any element of an individual violates its inequality constraint due to over/under speed then the position of the individual is fixed to its maximum/minimum operating point. Therefore, this can be formulated as follows: (Park et al., 2005)

$$P_{id}{}^{k+1} = \begin{cases} P_{id}{}^{k} + v_{ij}{}^{k+1}, \text{if } P_{id,Min} \le \left(P_{id}{}^{k} + v_{id}{}^{k+1}\right) \le P_{id,Max} \\ P_{id,Min}, \text{if } \left(P_{id}{}^{k} + v_{id}{}^{k+1}\right) < P_{id,Min} \\ P_{id,Max}, \text{if } \left(P_{id}{}^{k} + v_{id}{}^{k+1}\right) > P_{id,Max} \end{cases} \tag{2.31}$$

In MPSO the position adjustment strategy is incorporated in the PSO framework in order to provide the solutions satisfying the inequality constraints. The dynamic search-space reduction strategy is applied to accelerate the convergence speed. In (Saber et al., 2007), Modified PSO algorithm consists of problem dependent variable, number of promising values in velocity vector, unit vector, error-iteration and step length. It reliably and accurately traces a continuously changing solution of the complex cost function and no extra concentration/effort is needed for more complex higher order cost polynomial in Economic Load Dispatch (ELD).

In the Improved Particle Swarm Optimization (IPSO) method (Park et al., 2010), the performance of the conventional PSO is greatly improved by using a new velocity strategy equation, which is suitable for a large system. The Constriction Factor Approach (CFA) is incorporated into this velocity equation. Equation (2.32) is used and the new velocity strategy equation is formulated where the maximum and minimum velocity limits of each individual are calculated as follows:

$$v_d^{\max} = \left(\frac{P_d^{\max} - P_d^{\min}}{2} \right) \beta$$
$$v_d^{\min} = -\left(\frac{P_d^{\max} - P_d^{\min}}{2} \right) \beta \qquad (2.32)$$

Where

$$P_d^{\max} = \sum_{i=1}^{n} P_i^{\max}$$

$$P_d^{\min} = \sum_{i=1}^{n} P_i^{\min}$$

β=0.01

This approach also considers the security constraints such as line flow constraints and bus voltage.

Chaotic PSO is used in Park et al., 2010), (Coelho and Mariani, 2007), (Tao and Jin-ding, 2009) and (Mondal, 2013) to optimize the economic dispatch problems, since the conventional PSO has drawbacks such as local optimal trapping due to premature convergence (i.e., exploration problem), insufficient capability to find nearby extreme points (i.e., exploitation problem), and lack of efficient mechanism to treat the constraints (i.e., constraint handling problem). To overcome these drawbacks an improved PSO framework employing chaotic sequences combined with the conventional linearly decreasing inertia weights and adopting a crossover operation scheme to increase both exploration and exploitation capability of the PSO is developed in (Park et al., 2010). Chaotic Inertial Weight Approach (CIWA), is defined as follows

$$c\omega^k = \omega^k \gamma k \qquad (2.33)$$

Where

$c\omega^k$ Chaotic weight at k^{th} iteration

ω^k Weight factor of IWA

γk Chaotic parameter

The chaotic PSO approach hybridized with an Implicit Filtering (IF) technique to optimize the performance of economic dispatch problems is proposed in (Coelho and Mariani, 2007). The chaotic PSO with chaos sequences is the global optimizer and the IF is used to fine-tune the chaotic PSO run in sequential manner. The IF explores the search space quickly with a gradient direction and guarantees a good local solution. A modified Tent-map-based Chaotic PSO (TCPSO) to solve the economic load dispatch problem is proposed in (Tao and Jin-ding, 2009). The new tent-map-based chaotic PSO can be used to find the initial value for each particle in order to improve the global convergence, and reduce the influence caused by the particle's initial position. There after the PSO algorithm is used to find the optimal solution.

The Quantum-inspired Particle Swarm Optimization (QPSO) is described in (Meng et al., 2010), (Hosseinnezhad et al., 2011). The state of a particle is depicted by quantum bit and angle, instead of particle position and velocity in the classical PSO. The QPSO has stronger search ability and quicker convergence speed based on introducing quantum computing theory. It also has two special implementations such as self-adaptive probability selection and chaotic sequences mutation.

A development of an educational simulator for particle swarm optimization is done in (Lee et al., 2007). It solves the mathematical test functions as well as ED problems with non-smooth cost functions. In the simulator, instructors and students can select the test functions for simulation and set the parameters that have an influence on the PSO performance. Through the visualization process of each particle and the variation of the value of objective function, the simulator is particularly effective in providing users with an intuitive feel for the PSO algorithm. This educational simulator was developed in MATLAB 6.5, which is run in an interpreter mode on a wide variety of operating systems.

The hybrid PSO is used to solve the economic dispatch problems (Muneender and Kumar, 2009), (Shi and Eberhart, 2001), (Kumar et al., 2004). The optimal re-dispatch of transactions for congestion management is formulated as a Non-Linear Programming (NLP). The Adaptive Fuzzy Particle Swarm Optimization based Optimal Power Flow (AFPSO-OPF) is introduced in (Muneender and Kumar, 2009) for Congestion Management problem or multi congestion case to solve the NLP. In this method, the inertia weight is dynamically adjusted using adaptive fuzzy IF/THEN rules (Shi and Eberhart, 2001), (Kumar et al., 2004) to increase the balance between global and local searching abilities. The selection of generators from the most sensitive cluster/zone uses two distribution factors that are Real and Reactive Power Transmission Congestion Distribution Factors (PTCDFs and QTCDFs) (Muneender and Kumar, 2009) and (Kumar et al., 2004) to minimize the number of readjustments for the congestion management. A hybrid method that integrates particle swarm optimization (PSO) with sequential quadratic programming (SQP) is used to solve the non-smooth cost functions in (Sugsakarn and Damrongkulkamjorn, 2008). PSO is the main optimizer to find the optimal global region while SQP is used as a fine tuning to determine the optimal solution at the final stage. In (Khamsawang et al., 2010), four scenarios of mutation operators are used for improving diversity exploration of the standard PSO. The scenarios of mutation operators are expressed as the following equations:

1) Scenario 1 (PSO-DE1)

$$v_i^{(k+1)} = SC\left[(x_j^{(k)} - x_i^{(k)}) - (x_q^{(k)} - x_i^{(k)})\right] \tag{2.34}$$

2) Scenario 2 (PSO-DE2)

$$v_i^{(k+1)} = SC\left[(x_j^{(k-\beta)} - x_i^{(k)}) - (x_q^{(k-\beta)} - x_i^{(k)})\right] \tag{2.35}$$

3) Scenario 3 (PSO-DE3)

$$v_i^{(k+1)} = SC\left[(x_j^{(k)} - x_i^{(k)}) - (x_q^{(k)} - x_i^{(k)}) - (x_r^{(k)} - x_i^{(k)})\right] \tag{2.36}$$

4) Scenario 4 (PSO-DE4)

$$v_i^{(k+1)} = SC\left[(x_j^{(k-\beta)} - x_i^{(k)}) - (x_q^{(k-\beta)} - x_i^{(k)}) - (x_r^{(k-\beta)} - x_i^{(k)})\right] \tag{2.37}$$

where

SC	Scaling factor between 0.1 and 2
β	Value from the previous iteration defined by the user
k, q and r	Random indexes of the particles and $(k \neq q \neq r)$

The Table 2.3 gives the review overview of the single area economic dispatch problem. The objective function, equality and inequality constraints, algorithms, software, power system considered, and real-life implementation are used as criteria for comparison of the reviewed papers.

Most of the authors used simulation tools such as Fortan, C language and MATLAB to validate the economic dispatch solutions and there is no sufficient proof that the model is implementated in real-time for both single and multi-criteria economic dispatch problem. The reason might be the copy right issues to publish the data of the power plants in publications.

Table 2.3: **Review of single area, single criterion economic dispatch problem methods of solution**

Reference Paper	*Objective Function*	*Constraints*		*Algorithm*	*Software*	*Power System considered*	*Real life Implementation*
		Equality Constraints	*Inequality Constraints*				
(Adhinarayanan and Sydulu, 2006)	Cubic fuel cost function	Real power balance constraint	Nil	Particle Swarm Optimization	Not mentioned	i. 3 generator system with quadratic cost function ii. 3 generator system with cubic cost function iii. 5 generator system with cubic cost function iv. 26 generator system with cubic cost function	No
(Baskar and Mohan, 2008)	Fuel cost function	i. Real power balance constraint ii. Network power flow equation	i. Real power generation limit ii. Bus voltage limit constraint iii. Power limit on transmission line	Improved Particle Swarm Optimization	MATLAB	i. IEEE 14 bus system ii. 66 bus Indian utility system	No
(Chaturvedi et al., 2006)	Fuel cost function	i. Real power balance constraint ii. Transmission Loss constraint	Real power generation limit	Levenberg-Marquardt Back Propagation Algorithm	ANN tool box	i. 3 generating unit system ii. 6 generating unit system	No
(Demirovic and Tesnjak, 2011)	i. Fuel cost function ii. Multi-periodic optimal power flow	i. Real power balance constraint ii. Transmission Loss constraint	i. Spinning reserve ii. Transmission capacity of lines	Generalized algebraic Modeling system (GAMS) Algorithm	GAMS software	Four nodes and three generators system	No
(Dos santos Coelho and	Fuel cost function with valve point effect	Real power balance constraint	Real power generation limit	Chaotic PSO approach hybridized with an	MATLAB	13 generator system with valve point effect	No

Mariani, 2007)				Implicit Filtering (IF)			
(Gaing, 2003)	Fuel cost function	Real power balance constraint	i. Real power generation limit ii. Ramp Rate Limits iii. Prohibited Operating Zone iv. Line flow constraints	Particle Swarm Optimization	MATLAB	i. 6 thermal units, 26 buses, and 46 transmission lines. ii. 15 thermal units with prohibited operating zones and transmission loss coefficients iii. 40 units of the Tai power mixed generating system of coal, oil , gas and diesel	No
(Gnandass et al., 2004)	i. Fuel cost quadratic function ii. Fuel cost function of iii. Combined Cycle Cogeneration Plant (CCCP)	Real power balance constraint	Real power generation limit	Evolutionary programming	MATLAB	i. IEEE 14 bus systems ii. IEEE 30 bus systems (Fuel cost characteristic of slack bus generator is of CCCP nature and all other generators having quadratic fuel cost function)	No
(Hemamalini and Simon, 2010)	Fuel cost function with valve point effect	i. Real power balance constraint ii. Transmission Loss constraint	Real power generation limit	Maclaurin series based Lagrangian method	MATLAB	i. 40 generator system with Transmission loss neglected ii. 5 generator system with transmission loss considered	No
(Hemamalini and Simon, 2009)	Fuel cost function with valve point effect	i. Real power balance constraint ii. Transmission Loss constraint	Real power generation limit	Maclaurin series based Lagrangian method	MATLAB	i. 3 generator system with Transmission loss neglected ii. IEEE 30 Bus of 6 generator system with Transmission considered. iii. 3 generator system	No

						with Transmission loss neglected iv. 40 generator system with Transmission loss neglected	
(Hosseinnezhad et al., 2011)	Fuel cost function with valve point effect	Real power balance constraint	Real power generation limit	Quantum PSO	Not Mentioned	i. 13 generator system with valve point effect ii. 40 generator system with valve point effect	No
(Jayabarathi et al., 2008)	i. Fuel cost function with prohibited operating zones ii. Multiple fuel cost function iii. Fuel cost function for Multi Area Economic Dispatch iv. Fuel cost function for combined economic emission dispatch problem	Real power balance constraint	i. Real power generation limit ii. Prohibited Operating Zone iii. Tie-line limit constraint	Hybrid Differential Evolution technique (HDE)	MATLAB	i. 15 generating unit system with prohibited zones considered. ii. 10 generating unit systems with multiple fuel cost considered. iii. 6 generating unit systems with emission coefficient considered. iv. 4 area with 4 generating units interconnected by six tie lines	No
(Khamsawang, et al., 2010)	Fuel cost function	i. Real power balance constraint ii. Network transmission loss	i. Real power generation limit ii. Prohibited operating zones	PSO with Differential Evolution (PSO-DE)	MATLAB	6 generator system	No
(Lai et al., 2005)	Fuel cost function with valve point effect	Real and reactive power flow equations	Power generation limit	Particle Swarm Optimization	MATLAB	IEEE 30-bus system with 6 generator system	No
(Lee et al., 2007)	1. Economic dispatch a) Fuel cost function b) Fuel cost function	Real power balance constraint	Real power generation limit	PSO	Educational simulator for PSO using	i. 3 generator system with valve point effect ii. 40 generator system with	No

	with valve point effect 2. Mathematical function a) Sphere function b) Rosenbrock function c) Ackley's function d) generalized Rastrigin function e) Generalized Griewank function				JAVA	valve point effect	
(Manoharan et al., 2008)	i. Fuel cost function ii. Fuel cost function with valve point effect iii. Multiple fuel cost function	Real power balance constraint	Real power generation limit	i. Real-coded genetic algorithm ii. Particle swarm optimization iii. Differential evolution	MATLAB	i. 10 generator system with non-smooth cost functions and multiple fuel options. ii. 10 generator system with Non-smooth cost functions considering both valve-point effects and multiple fuel options	No
(Meng et al., 2010)	Fuel cost function with valve point effect	Real power balance constraint	Real power generation limit	Quantum-Inspired PSO	Not Mentioned	i. 3 generator system ii. 13 generator system with valve point effect iii. 40 generator system with valve point effect	No
(Pandi et al., 2009)	Fuel cost function with cost of tie line flow	i. Area Power Balance Constraints ii. Transmission Loss constraint	iii. Generator Constraints iv. Prohibited Operating Zone v. Tie Line constraints	Improved Harmony Search	MATLAB	i. single area six unit system with loss coefficients ii. multi area economic load dispatch having four areas along with tie-line constraints iii. single area six unit system with loss	No

						coefficients and prohibited operating zones.	
(Panta et al., 2007)	Fuel cost function but the fuel cost is not estimated	Real power balance constraint	Real power generation limit	i. Lambda iteration ii. Artificial Neural Network (ANN)	MATLAB	i. 3 generating unit system ii. 10 generating unit system iii. 20 generating unit system	No
(Park et al., 2010)	i. Fuel cost function with valve point effect ii. Multiple fuel cost function with valve point effect	i. Real power balance constraint ii. Network transmission loss	i. Real power generation limit ii. Ramp rate limits iii. Prohibited operating zones	Improved PSO	Not Mentioned	i. 40-unit system with valve-point effects ii. 15-unit system with prohibited operating zones, ramp rate limits, and transmission network losses iii. 10 unit system considering multiple fuels with valve-point effect iv. 140-unit Korean power system with valve-point effects, prohibited operating zones, and ramp rate limits.	No
(Park et al., 2005)	i. Fuel cost function with valve point effect ii. Multiple fuel cost function	Real power balance constraint	Power generation limit	Modified Particle Swarm Optimization	Not Mentioned	i. 3 generator system with valve point effect ii. 40 generator system with valve point effect iii. 10 generator system with multiple fuel cost function	No
(Ratniyomchai et al., 2010)	Fuel cost function with valve point effect	Real power balance constraint	Real power generation limit	i. Genetic Algorithm ii. Evolutionary Algorithm iii. Adaptive Tabu	MATLAB	IEEE 30 bus system with six thermal power generating plant	No

				Search iv. Particle Swarm Optimization v. Improved Harmony Search			
(Saber et al., 2007)	i. Fuel cost function ii. Multiple fuel cost function iii. Emission function iv. Maintenance cost function	Real power balance constraint	i. Power generation limit ii. Spinning reserve iii. Ramp rate iv. Prohibited operating zones v. Transmission loss constraints vi. Network losses	Modified Particle Swarm Optimization	Turbo C/C++	6 generator system	No
(Sugsakaran and Damrongkulkamjorn, 2008)	i. Fuel cost function ii. Fuel cost function with valve point effect iii. Multiple fuel cost function iv. Multiple fuel cost function with valve point effect	Real power balance constraint	Real power generation limit	PSO with Sequential Quadratic Programming (PSO-SQP)	MATLAB	i. 10 generator system with multiple fuel cost ii. 10 generator system considers multiple fuel cost with valve point effect	No
(Tao and Jin-ding, 2009)	Fuel cost function with valve point effect	Real power balance constraint	Real power generation limit	Tent-map based chaotic PSO (TCPSO)	MATLAB	i. 10 generator system ii. 3 generator system with valve point effect iii. 13 generator system with valve point effect	No
(Tiacharoen et al., 2010)	i. Fuel cost function ii. Fuel cost function with valve point effect iii. Multiple fuel cost function	i. Real power balance constraint ii. Transmission Loss constraint	i. Ramp Rate Limits ii. Prohibited Operating Zone	Bees algorithm	MATLAB	i. 15 thermal unit system with transmission loss considered ii. 40-unit system with valve-point effects considered. iii. 10-unit system with multiple-fuel effects considered.	No

2.4 Investigation of a single area multi-criteria power system dispatch problem

Energy management has to perform complicated and timely restricted system control function to operate a large power system reliably and effectively. In electric power system operation, the objective is to achieve the most economical generation policy that could supply the local demands without violating constraints. Thermal power plants play a major role in power production and they burn fossil fuels that generate toxic gases. This creates pollution for the environment. There are two objectives i) minimum cost and ii) minimum emission, converted into a single objective function determining the so called emission constrained economic dispatch (Nanda et al., 1994). It is compulsory for the electric utilities to reduce the pollution level by reducing sulphur dioxide, nitrogen oxide and carbon dioxide gases. The multi-criteria power system dispatch problem is solved using two methods in the existing literature. They are Weighted Sum Method (WSM) and price penalty factor method. According to (Rani et al., 2006), (Jeyakumar et al., 2006), (Jeyakumar et al., 2007), (Wong et al., 1993), (Yu et al., 2011), and (Ozyon et al., 2009) the bi-criteria dispatch problem is solved using WSM method where the criterion function is expressed as (Wong et al., 1993)

$$F_T = \omega\sum_{i=1}^{n}(a_iP_i^{\,2}+b_iP_i+c_i)+(1-\omega)\sum_{i=1}^{n}(d_iP_i^2+e_iP_i+f_i)\ \text{[\$/h]} \qquad (2.38)$$

Where

F_T Combined Economic Emission Dispatch (CEED) fuel cost value,

ω Weighing coefficient,

n Number of generators,

a_i, b_i, c_i Cost coefficients, and

d_i, e_i, f_i Emission coefficients for the i^{th} generator.

When ω=1, only the fuel cost objective is considered,

When ω=0, only the emission value objective is accounted for.

In the papers (Dhillon et al., 1994), (Tsai and Yen, 2010), (Hemamalini and simon, 2009), (Thakur et al., 2006), and (Ming et al., 2010) the multi-criteria dispatch problem is solved using Max-Max price penalty factor method. The price penalty factor h_i is the ratio between maximum fuel cost and maximum emission of corresponding generator and the objective function is expressed as

$$F_T = \sum_{i=1}^{n}\left[(a_iP_i^{\,2}+b_iP_i+c_i)+h_i\left(d_iP_i^2+e_iP_i+f_i\right)\right] \quad \text{[\$/h]} \qquad (2.39)$$

Where

h_i is the Max-Max price penalty factor and is given in Equation (2.17).

The solution of the bi-criterion global optimization is described in (Rani et al., 2006) and (Jeyakumar et al., 2006) using Particle Swarm Optimization (PSO) technique. According to (Jeyakumar et al., 2007) Multi-Objective Evolutionary Programming (MOEP) method is used to solve the Combined Economic Emission Dispatch (CEED) problem. A bi-criterion global optimization problem to determine the most appropriate generation dispatch solution of the fuel cost, the environmental cost and operation security of power networks is solved in (Wong et al., 1993). The approach is based on the simulated annealing technique.

The Maclaurin series-based Lagrangian method is used in (Srikishna and Palanichamy, 1991) to solve complicated, non-convex and non-linear economic dispatch problems. In this method, the rectified sinusoid function is represented by Maclaurin sine series expansion. The problem is solved using the Lagrangian method iteratively for calculations of the dual variable. In (Subramanian and Ganesan, 2010) a sequential approach with matrix framework is used to solve the CEED problem using Max-Max price penalty factor. According to (Ming et al., 2010), PSO algorithm is applied to find the suboptimal solution using the Distance Index (DI) method to select the candidate members and exact global optimal solution is obtained by Direct Approach (DA) method. The weighting factors are used to get the best compromise solution for both emission and economic cost dispatch. The environmental economic power dispatch problem in hydrothermal power systems (Ozyon et al., 2009) is solved by using genetic algorithm. The CEED problem is solved using Max-Max penalty factor by PSO algorithm in (Thakur et al., 2006). Solutions of the multi-objective problems are grouped into two methods namely non-interactive and interactive methods (Dhillon et al., 1994). In the non-interactive type methods a global preference function of the objectives is identified and optimized with respect to the constraints. In the interactive methods, a local preference function or trade-off among objectives is identified by interacting with the decision maker, and the solution process proceeds gradually toward the globally satisfactory solution. It incorporates a sensitivity measure into multi-objective optimization, which generates a non-inferior optimal solution with respect to the objective function and the sensitivity index. The index provides useful information about the distribution of the optimal solution in the presence of variations in the model parameters defining the problem. The decision maker is able to analyze the sensitivity information conveniently. The sensitivity index is a scalar-valued quantity, regardless of the number of objectives. The most important characteristic of the sensitivity index is that a sensitivity trade-off is calculated at each non inferior point. This allows the decision maker to know the trade-offs between objective levels and parameter sensitivity.

The environmentally constrained economic dispatch problem is a multi-objective nonlinear optimization problem with constraints. In (Thakur et al., 2006) an efficient and reliable Particle Swarm Optimization (PSO) algorithm based technique for solving the emission and economic dispatch problem by using Max-Max Price penalty factor is considered. It mainly focuses on nitrogen oxides and sulfur dioxide emissions of thermal power plants. In (Tsai and Yen, 2010) an Improved Particle Swarm Optimization (IPSO) is presented to solve the economic dispatch problems considering fuel cost, environmental issue, and valve point effect loading. The IPSO is developed in such a way that PSO with Constriction Factor (PSO-CF) is applied as a basic level search, which can give a good direction to the optimal global region. The IPSO introduces two operators, "Random Particles" and "Fine-Tuning" into the PSO-CF algorithm for increasing the search ability. The process of "Random Particles" will add a proper random particle into the group of particles when the solution is searched in each generation. The procedure of "Fine-Tuning" will regulate the best position of the group from the past generation of PSO-CF algorithm. Table 2.4 gives the review of single area, multi-criteria power dispatch problem. The objective function, multi-objective criteria, equality and inequality constraints, algorithm, software, and sample system used in (Palanichamy and Basu, 2008), (Ming et al., 2010), (Krishnamurthy and Tzoneva, 2011) and few other papers are given.

2.5 Investigation of a multi-area power system dispatch problem methods for solution

This section reviews the earlier work on Multi Area Economic Emission Dispatch (MAEED) problem, MAEED with tie line constraints and transmission capacity constraints. Fuzzy and evolutionary programming techniques of MAEED problem, multi-area unit commitment problem, component model, Remote Method Invocation (RMI) based distributed model and grid service model for Multi-area Economic Dispatch (MAED) without considering the emissions of the plant is termed as MAED problem, parallel and sequential PSO algorithm of MAEED problem are considered.

Table 2.4: **Review of single area, multi-criteria power dispatch problem methods of solution**

Reference Paper	Objective function	Multi-objective Criteria formulation	Constraints		Algorithm	Software	Power System considered	Real life Implementation
			Equality Constraints	Inequality Constraints				
(Balamurugan and Subramanian, 2008)	i. Fuel cost function ii. Emission function	i. Max – Max ii. Min – Min iii. Average [(Max-Max)+(Min-Min)]/2 iv. Common (Average/n)	Real power balance constraint	Real power generation limit	Dynamic programming technique	Not mentioned	i. 11 generator systems ii. 6 generator system with various price penalty factors	No
(Danaraj and Gajendran, 2005)	i. Fuel cost function ii. Emission function	Max – Max price penalty factor	Real power balance constraint	Real power generation limit	Quadratic programming	Not mentioned	6 generator system	No
(Dhillon et al., 1994)	i. Fuel cost function ii. Emission function	ε – constraint method	i. Real power balance constraint ii. Network transmission loss	i. Real power generation limit ii. Area power balance iii. Tie-line constraint iv. Prohibited operating zones	ε – constraint method	Not mentioned	3 generator system	No
Jeyakumar et al., 2007)	i. Fuel cost function ii. Emission function	Weighted sum method	Real power balance constraint	Real power generation limit	Multi-Objective Evolutionary Programming (MOEP)	MATLAB	i. 3 generator system ii. 6 generator system	No
(Jeyakumar et al., 2006)	i. Fuel cost function ii. Emission function	Weighted sum method	i. Real power balance constraint i. Network transmission loss	i. Real power generation limit ii. Area power balance iii. Tie-line constraint iv. Prohibited	PSO	MATLAB	i. 4 area system interconnected by six tie Lines ii. 10 generating units each with three types of fuel	No

				operating zones			iii. 6 generating units with fuel cost coefficients, NOx emission co-efficient, and transmission loss coefficients iv. 15 unit power system with four of the units having up to three prohibited operating zones.	
(Karthikeyan et al., 2009)	i. Fuel cost function ii. Emission function	Max – Max price penalty factor	i. Real power balance constraint ii. Power flow equations	i. Real power generation limit ii. System spinning reserve iii. Minimum up time iv. Minimum down time v. Ramp rate limits vi. Power flow inequality constraints	Unit commitment approach	MATLAB	i. IEEE 14 bus system ii. IEEE 30 bus system iii. IEEE 57 bus system iv. IEEE 118 bus system v. Indian utility 75 bus system	No
(Krishnamurthy and Tzoneva, 2011)	i. Fuel cost function ii.Emission function iii. CEED function	i. Min – Max price penalty factor ii. Max – Max price penalty factor	Real power balance constraint including Transmission loss	Real power generation limit	Lagrange's	MATLAB	IEEE 30 bus with 6 generators including transmission loss and emission coefficients	No
(Krishnamurthy and Tzoneva, 2012)	i. Fuel cost function ii. Emission function iii. CEED function	i. Min – Max price penalty factor ii. Max – Max price penalty factor	Real power balance constraint	Real power generation limit	PSO	MATLAB	IEEE 30 bus with 6 generators including emission coefficients	No

(Kulkarni et al., 2000)	i. Fuel cost function ii. Emission function	Max – Max price penalty factor	Real power balance constraint	Real power generation limit	Improved Back Propagation Neural Network (IBPNN)	MATLAB	6 generator system	No
(Ming et al., 2010)	i. Energy saving function ii. Fuel cost function iii. Emission function with valve point effect	i. Max – Max price penalty factor ii. Weighted sum method	Real power balance constraint	Real power generation limit	PSO	MATLAB	IEEE 30 bus with 6 generators including NOx emission coefficients	No
(Ozyon et al., 2009)	i. Fuel cost function ii. Emission function	Weighted sum method	i. Real power balance constraint ii. Transmission loss	Real power generation limit	Genetic Algorithm	Not mentioned	16 buses with 4 thermal generators and 4 hydro generators	No
(Palanichamy and Babu, 2008)	i. Fuel cost function ii. Emission function	Max – Max price penalty factor	ii. Real power balance constraint iii. Network transmission loss	Real power generation limit	Lambda approach	Not mentioned	6 generator system	No
(Rani et al., 2006)	i. Fuel cost function ii. Emission function	Weighted sum method	Real power balance constraint	Real power generation limit	PSO	MATLAB	6 generator system	No
(Subramanian and Ganesan, 2010)	i. Fuel cost function ii. Multiple fuel cost function iii. Emission function	Max – Max price penalty factor	i. Real power balance constraint ii. Transmission loss	i. Real power generation limit ii. Ramp rate limits iii. Prohibited operating zones iv. Area power balance	Sequential approach with matrix framework	MATLAB	i. 40 unit Taipower system mixed - generating system ii. 15 generator system with cost and loss coefficients, ramp rates, prohibited	No

				v. Tie line limits			operating zones iii. 6 generator system with cost, emission and loss coefficients iv. 2 area with 40 thermal units v. 10 generator system with multiple fuels	
(Thankur et al., 2006)	i. Fuel cost function ii. Emission function	Max – Max price penalty factor	i. Real power balance constraint ii. Transmission loss	Real power generation limit	PSO	Not mentioned	3 generator system with NO_x and SO_2 emission coefficients	No
(Tsai and Yen, 2010)	i. Fuel cost function with valve point effect ii. Emission function with valve point effect	Separate solution for economic dispatch and emission dispatch problems	Real power balance constraint	i. Real power generation limit ii. Transmission line loss constraint	Improved PSO	MATLAB	i. 40 generating system ii. 6 generating system iii. 10 generating system	No
(Wong et al., 1993)	i. Fuel cost function ii. Emission function	Separate solution for economic and emission dispatch problems	Real power balance constraint	i. Real power generation limit ii. Tie–line constraint	bi-criterion global optimization approach	Not mentioned	3 area with 6 generators, 3 types of fuels (coal, oil and gas) and 3 types of emissions (NO_x, SO_2 and CO_2) are considered	No
(Yu and Chung, 2011)	i. Fuel cost function ii. Emission function	Separate solution for economic dispatch and emission dispatch problems	i. Real power balance constraint ii. Transmission loss	i. Real power generation limit ii. System spinning reserve constraints	Multiple PSO	Not mentioned	6 generator system	No

MAEED problem with linearization of the fuel cost curves, line active-power flows, spinning reserves and network balance equations are solved using Dantzig-Wolfe decomposition principle with the revised simplex method which is iterated together with a fast decoupled load flow algorithm in (Quintana et al., 1981).

A decomposition approach to multi-area generation scheduling problem is considered in (Wang and Shahidehpour, 1992). It uses a two-level decomposition to solve the problem as shown in Figure 2.8. In the first decomposition, the problem is divided into several sub problems during the study period. The information that the master problem sends to each sub problem is the load demands of all areas at the corresponding hour and the output of the sub problem is the system operation cost at that time. The coordination factor of this layer of decomposition is the minimum operation cost of the system in the given period. The second layer of decomposition divides the previous sub problems further according to the control areas in the power pool. The sub problem for each area receives system λ and returns the area λ, where λ is the tie-line coordinator variables vector.

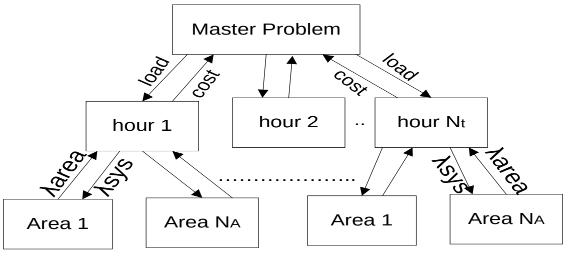


Figure 2.8: **Two level decomposition problem (Wang and Shahidehpour, 1992)**

In (Jamshidian et al., 1983) a Jacobian based algorithm is used to calculate penalty factors for a multi-area power system. It includes calculation of the penalty factor at the interchange boundary of an area. It appears to be superior to the classical B-coefficient technique in comparison to its speed and accuracy. B-coefficient method is not suitable for multi-area dispatch problems. In (Helmick and Shoults, 1985) an Area Control Error (ACE) signal is calculated for three operating companies in the Texas Utilities (T.U.) System. The difference between the desired generation and the actual generation for each area is called ACE. This feature is in sorted-table approach to economic dispatch and calculates the ACE for every five seconds.

Two types of margin time are important; (i) time to satisfy system frequency and dynamic stability and (ii) time to satisfy loss of generation or other facilities. These margin times are normally of the order of one minute and five minutes for hydro plants and are considered in (Chawdhury and Billinton, 1990). The committed units in the system should be dispatched in a way that the response risk at the single system level, designated as the Single System Response Risk (SSRR), should be less than or equal to a specified level. Once the response risk criteria at the single system level are satisfied in each system, all response assistances are considered at the interconnected level. All response assistances to a given system in the interconnected configuration are added to form an equivalent response model and the response risk, designated as the Interconnected System Response Risk (ISM), must satisfy a specified risk level presented in (Chawdhury and Billinton, 1990).

Spinning reserve location in multi-interconnected power systems can be considered using the multi-area techniques given in (Billinton and Jain, 1972). The allocation of spinning reserve amongst the committed units can be done by selecting a suitable risk level. The response risk evaluation technique for a single system is discussed in detail in (Billinton and Allan, 1984). An interconnected system is required to satisfy a Single System Risk (SSR) in which possible assistance from its neighbors is not taken into account. In addition, the interconnected system is required to satisfy its Interconnected System Risk (ISR) in which assisting from its neighbors is considered in (Billinton and Chowdhury, 1988).

Multi-area economic load dispatching control with a combination of spot pricing as demand side option and economic power interchange as a supply side option to keep demand and supply balance is considered in (Okada and Asano, 1995). A region that has a generation capacity larger than its load demand, and can supply interchange power to the interconnected power network, is called an “interchange power selling region”. Wheeling is the transmission of electrical power from the selling region to the buying region through a transmission network owned by an intermediate region. As shown in Figure 2.9, a region that can sell wheeling power to the buying region through an intermediate region which directly connects to the buying region is called “wheeling power selling region”.

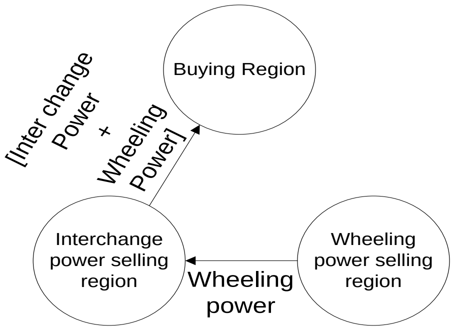


Figure 2.9: **Concept of buying and selling power (Okada and Asano, 1995)**

Multi-Area Economic Dispatch problem using an Incremental Network Flow Programming (INFP) algorithm is presented in (Streiffert, 1995). The INFP formulation offers significant advantages over both the lambda iteration and linear programming method in terms of computation time and convergence properties. In INFP each iteration is feasible with respect to all modeled constraints and convergence issues only relate to cost improvement.

Novel Nonlinear Convex Network Flow Programming (NLCNFP) was developed in (Zhu and Momoh, 2001) in order to solve the secure economic dispatch problem. The line security constraints in each area and tie-line capacity constraints are considered, the buying and selling contract in a multi-area environment is also introduced. The solution is a new nonlinear convex network flow programming, which is a combined method of Quadratic Programming (QP) and Network Flow Programming (NFP). The concept of maximum basis in a network flow graph was introduced so that the constrained model was changed into an unconstrained QP model, and is solved by the reduced gradient method. In (Chen and Chen, 2001), Direct Search Method (DSM) is used to handle the constraints and minimize the total generation cost in the multi-area reserve constrained economic dispatch problem. Enhanced Direct Search Method (EDSM) for two area power system is used in (Zarei et al., 2007). The capacity limit acts as an important role in two area power systems to have minimum total generation cost; system tends to use the all capacity of cheap area. It is an idea to tie-line compensation on economic dispatch during the total demanded load variation.

In (NERC report, 1996) Available Transfer Capability (ATC) is a measure of the transfer capability remaining in the physical transmission network for further commercial activity over the already committed uses. In (Manikandan et al., 2008), the evaluation of multi-area Available Transfer Capability (ATC) using AC Power Transfer Distribution Factors (ACPTDF) and Participation Factors (PF) is used in Combined Economic Emission Dispatch (CEED) problem.

ATC = TTC - Existing Transmission Commitments.

where, the Total Transfer Capability (TTC) is defined as the amount of electric power that can be transferred over the interconnected transmission network.

Bilateral forward contracts give participants the obligation to sell (or buy) a specific amount of power, at a specified price, within a specified period of time.

Reliability Must-Run (RMR) contracts enter between an Independent System Operator or Regional Transmission Organization (ISO/RTO) and generation owners.

The “call” option gives the buyer the right to buy and the “put” option gives the right to sell. In (Yingvivatanapong et al., 2008), an approach to incorporate power contracts, which include call and put options, forward contracts, and reliability must-run contracts, into multi-area unit commitment and economic dispatch solutions is developed.

PSO and DE based evolutionary strategies for multi-area economic emission dispatch (MAEED) are used in (Sharma et al., 2010). Evolutionary methods do not suffer from convexity assumptions and achieve fast solutions even for complex non-linear, non-convex, multi-modal optimization problems. Fuzzified Particle Swarm Optimization (FPSO) algorithm is used in (Somasundaram and Jothi Swaroopan, 2011), for solving the security-constrained multi-area economic dispatch of an interconnected power system. An adaptive inertia weight of PSO algorithm can be obtained from the fuzzy logic strategy, thereby leading to an improved PSO technique called the FPSO. Combined application of Fuzzy logic strategy incorporated in both Evolutionary Programming (EP) and Tabu-Search (TS) algorithms are used in (Prasanna and Somasundaram, 2009) to solve the MAEED problem. An adaptive scaling factor or the variance of the Evolutionary Programming (EP) can be obtained from the fuzzy logic strategy and is termed as Fuzzy Mutated Evolutionary Programming (FMEP). In Tabu search algorithm the fuzzy logic is applied to Mutation and Recombination process. Thus leading to the term called Fuzzy Guided Tabu Search (FGTS).

The multi area unit commitment problem is solved using Lagrangian relaxation and dynamic programming methods in (Ouyang and Shahidehpour, 1991), (Lee and Feng, 1992), and (Tseng et al., 1998).

The component model architecture for economic load dispatch of multi-area power system is developed in (Kannan and Veilumuthu, 2004) and is shown in Figure 2.10. A component which is based on a single-server serving multiple clients has been proposed which enables all neighboring power systems to have simultaneous access to the remote economic load dispatch server for obtaining continuous load dispatch solutions. An EJB (Enterprise Java Beans) based, distributed environment has been implemented in such a way that each power system client can access the remote economic load dispatch EJB server through JNDI (Java Naming and Directory Interface) naming service with its power system data.

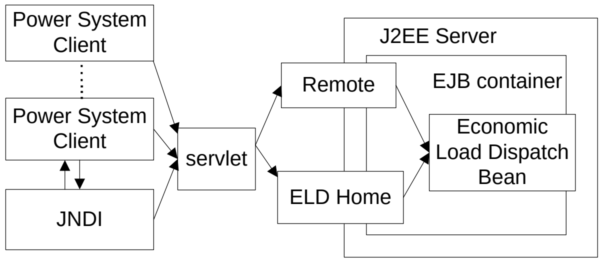


Figure 2.10: **Component model for MAED problem (Kannan and Veilumuthu,2004)**

The server computes the economic load dispatch and it provides the continuous automated load dispatch solutions to all the registered power system clients.

Remote Method Invocation (RMI) based distributed environment has been implemented in (Nithiyananthan and Ramachandran, 2005) in such a way that for every specific period of time, the remote server obtains the system data simultaneously from the neighboring power systems which are the clients registered with the remote ELD server and the optimized economic load dispatch solutions with power loss from the server are sent back to the respective clients.

A distributed model using grid environment through which the economic load dispatch solutions of multi-area power systems can be obtained continuously is developed in (Ramesh and Ramachandran, 2007). Grid computing is a viable solution in order to exploit the enormous amount of computing power available across Internet to solve large interconnected power system problems. This model is designed in such a way that any node in the grid can provide the economic load dispatch (ELD) solution. The node which serves as a server node at a specific instance in the grid can obtain the power system data from other client grid nodes and responds with economic load dispatch solutions.

An overview of PSO algorithm is done in (Poli et al., 2000). The existing parallel implementations are done by Parallel Synchronous Particle Swarm Optimization tasks, but they do not make efficient use of computational resources when a load imbalance exists. In (Koh and George et al., 2006), a Parallel Asynchronous PSO (PAPSO) algorithm is used to enhance computational efficiency. The performance of the PAPSO algorithm was compared to that of a PSPSO algorithm in homogeneous and heterogeneous computing environments for small- to medium-scale analytical test problems and a medium-scale biomechanical test problem. The results state that parallel performance of PAPSO was significantly better than that of PSPSO for heterogeneous computing environments or heterogeneous computational tasks. In (Kim et al., 2007), parallel PSO algorithm based on PC-cluster system is applied to the Optimal Power Flow (OPF) problem. The parallel PSO algorithm can divide the population of the PSO into several sub populations to share the burden of calculating the load flow. The results show that computing time of the parallel PSO algorithm is improved in comparison to the sequential PSO.

Multi-population binary clustered particle swarm optimization (BCPSO) algorithm is used to solve short term thermal generation scheduling problem in (Senjyu et al., 2008). This algorithm provides a way to explore larger search space and reduces the probability of local trapping. In (Jeyakumar et al., 2006), (Wang and Singh, 2006), (Azadani et al., 2008), and (Wang and Singh, 2009) the sequential PSO algorithm is used to solve the MAED problem. The Multi-area Environmental/Economic Dispatch (MAEED) problem addresses the environmental issue during the ED in (Wang and Singh, 2006).

The MAEED problem is first formulated and then an improved Multi-Objective Particle Swarm Optimization (MOPSO) algorithm is developed to derive a set of Pareto-optimal solutions to the MAEED problem.

Table 2.5 gives the review of the method for solution of the multi-area power dispatch problem. The objective function, multi-objective criteria, tie-line limits, equality constraints and inequality constraints, algorithm, software, and power system used are given.

Table 2.5: **Review of multi-area, single and multi-criteria power dispatch problem methods of solution**

Reference Paper	***Objective function***	***Multi-area Constraints***	***Constraints***		***Algorithm***	***Software***	***Power System considered***	***Real life Implementation***
			Equality Constraints	***Inequality Constraints***				
(Chawdhury and Billinton, 1990)	Incremental running cost	Tie–line active power flow	i. Single system risk ii. Interconnected system risk	Spinning reserve	Numerical calculation	Not mentioned	10 unit system with 3 and 4 state model	No
(Chen, 2001)	Fuel cost function	Tie–line active power flow	Real power balance constraint	i. Real power generation limit ii. Spinning reserve iii. Area spinning reserve iv. Transmission capacity constraints	Direct search procedure	Pentium III-500 Mhz PC (Software not mentioned)	3 area Taiwan power system with 87 thermal units	No
(Helmick and Shoults, 1985)	Area control error signal	Area control error	Real power balance constraint	Inequality constraints are not considered	User defined Area control error algorithm using computational iterative process	SCADA programmed with 16-Bit mini-computer in PASCAL language	3 area such as Texas Power and Light (TP&L), Texas Electric Service Company (TESCO) and Dallas Power and Light (DP&L), with a total aggregate system peak load of over 14,000 MW and a generation capability nearly 18,000 MW	Yes, Texas Utilities System
(Jamshidian et al., 1983)	Optimum real power generation dispatch	i. Load distribution	iii. Real power balance constraint iv. Transmission	Inequality constraints are not considered	Jacobian method	Not mentioned	Modified IEEE 30 bus system with 2 area and 3 area	No

		vector ii. Penalty factor	loss					
(Jeyakumar et al., 2006)	System operation cost	Tie-line limits	Area power balance constraint	Real power generation limit constraint	PSO	MATLAB 6.5	4 area interconnected by 6 tie-lines	No
(Kannan and velimuthu, 2004)	Total generation cost	Tie-line power	Power generation in each bus	Inequality constraints not considered	Enterprise Java Beans (EJB)	Component model architecture using JAVA	6,9,10, and 13 bus sample system	No
(Kim et al., 2007)	Fuel cost of the generator	Multi-area constraint not considered	Real and reactive power flow equations	i. Transmission line loadings ii. Load bus voltages, iii. Reactive power iv. generations of generator v. Active power generation of slack generator. vi. Active power output Voltage of generators Transformers tap ratio Shunt capacitors	Parallel PSO algorithm	PC cluster system with 6 Intel Pentium IV 2GHz processors using Matlab software	IEEE 30-bus system	No
(Koh et al., 2006)	i. Analytical test problems ii. Biomechanical test problem	Multi-area constraint not considered	Equality constraints not considered	Inequality constraints not considered	parallel Asynchronous Particle Swarm Optimization (PAPSO)	i. Homogeneo-us Linux cluster of 20 identical machines using Matlab	i. Analytical test problems ii. Biomechanical test problem	No

						software ii. 20 heterogeneo-us machines from several Linux clusters using Matlab software		
(Lee and Feng, 1992)	Fuel cost of the interconnected system	Tie-line power flow limit	Area power balance constraint	i. Real power generation limit constraint ii. Spinning reserve in each area iii. Minimum up time and minimum down time iv. Must Run and Must Off constraints	Commitment utilization factor	16 MHz INTEL 80386 based computer using C program	Three operating areas of Midwestern utility system	No
(Manikandan et al., 2008)	i. Fuel cost function ii. Emission cost function	Available Transfer Capability (ATC)	Real power balance constraint	i. Real power generation limit ii. Network power flow iii. Voltage constraint iv. Power limit on transmission line	multi-area ATC using AC Power Transfer Distribution Factors (ACPTDF) and Participation Factors(PF)	Power world Simulator	i. IEEE 30 bus system with 2 area ii. IEEE 118 bus system with 3 area	No

(Nithiyananthan and Ramachandran, 2005)	Total generation cost	Tie-line power	Power generation in each bus	Inequality constraints not considered	RMI used in built Java security mechanism	Remote Method Invocation (RMI) architecture using JAVA software	6,9,10, and 13 bus sample system	No
(Okada and Asano, 1995)	Fuel cost function	Tie-line active power flow	Real power balance constraint	Real power generation limit	Lagrange's method	Not mentioned	3 area with 5 generators in each area with tie-line and electricity price on each area is considered.	No
(Ouyang and Shahidehpour, 1991)	Fuel cost of the interconnected system	Tie-line power flow limit	Area power balance constraint	i. Real power generation limit constraint ii. Spinning reserve in each area iii. Minimum up time and minimum down time	Heuristic approach	C programming language in IBM-PC machine	4 area each having 26 thermal units	No
(Prasanna and somasundaram, 2009)	Fuel cost of the interconnected system	Tie-line power flow limit	Area power balance constraint	Real power generation limit constraint	i. Fuzzy Mutated Evolutionary Programming (FMEP) ii. Fuzzy Guided Tabu-Search(FGTS)	Intel Pentium IV 2.5-GHz processor (Software not mentioned)	3 area, 32 bus interconnected system. (Extended IEEE 30 bus system)	No
(Quintana et., al,	Fuel cost function	Tie –line	i. Real power balance constraint	i. Spinning reserves ii. Network power	Dantzing-Wolfe decompositi	UNIVAC-1100 computer,	i. IEEE 14 node system, 3 area with spinning	No

1981)		active power flow	ii. Line active power flows	balance equations	on principle	Fortran IV language.	reserves and tie-line active power flows are considered ii. 23 nodes, 24 generators , 2 area with spinning reserves and line flows are considered . iii. Mexican power network of 82 nodes, 32 generators , 3 area with spinning reserves and line flows are considered	
(Ramesh and Ramachandran, 2007)	Total generation cost	Multi-area constraint not considered	Power generation in each bus	Inequality constraints not considered	XML (eXtensible Mark-up Language) and web Services	Grid service model using JAVA software	3, 5 and 20 bus sample system	No
(Senjyu et al., 2008)	Total production cost	Multi-area constraint not considered	System real power balance	i. System reserve requirements ii. Power generation limit iii. Minimum up/down time iv. Ramp rate constraint v. Transmission line constraint	Multi-Population Binary Clustered Particle Swarm Optimization (BCPSO) algorithm	C++ Pentium IV machine 512 MB RAM	10, 20, 40 unit system and Tai-power 38 unit system	No
(Somasundaram and Jothi Swaroopan, 2011)	Fuel cost of the interconnected system	Tie-line power flow limit	Area power balance constraint	Real power generation limit constraint	Fuzzified particle swarm optimization algorithm	Intel Pentium IV 2.5-GHz processor (Software not	3 area, 32 bus interconnected system. (Extended IEEE 30 bus system)	No

						mentioned)		
(Streiffert, 1995)	Fuel cost function	Tie–line active power flow	Real power balance constraint	Real power generation limit	Incremental network flow programming algorithm	Not mentioned	4 area with 4 generators in each area	No
(Tseng et al., 1998)	Total generation cost	Transmission constraints	Multi-area demand constraints	i. Multi-area spinning reserve constraints ii. Area interchange restriction iii. Local unit constraints iv. Minimum up time and minimum down time	Lagrangian relaxation algorithm	FORTRAN on HP 700 workstation.	i. IEEE test system which has 24 buses, 34 transmission lines and 32 generating units. ii. Sample test system of 2500 lines, 2200 buses and 79 Generating units.	No
(Wang and Shahidehpour, 1992)	Fuel cost function	Tie–line active power flow	i. System power balance ii. Interchange transaction balance iii. Transmission loss	i. Area spinning reserve requirements ii. Area generation limits	Expert systems	Not mentioned	4 area system with 26 units in each area	No
(Wang and Singh, 2006)	i. Fuel cost function ii. Emission function	Tie-line limits	Area power balance constraint	Real power generation limit constraint Spinning reserve requirements	Enhanced Multi-objective Particle Swarm Optimization (MOPSO)	MATLAB	4 area test system with 4 generators in each area	No
(wang and Singh,	i. Fuel cost function	Tie-line	Area power balance constraint	i. Real power generation limit	Improved Multi-	MATLAB	4 area with 4 generators in each area with different	No

2009)	ii. Emission function	transfer limit		constraint ii. Area Spinning reserve requirements iii. System security constraints	objective Particle Swarm Optimization (MOPSO)		fuel and emission characteristics	
(Yingvivatanapong et al., 20080)	The total operating cost (Fuel costs, operation , maintenance costs (OMC), and startup costs)	Tie-line capacity constraints	Power balance constraint	i. Spinning reserve constraint ii. Area import/export constraints iii. Tie line capacity constraints iv. Initial condition of unit v. Generation output limits vi. Maximum ramp-up and ramp-down vii. Minimum up-time and down-time viii. Unit restrictions (must-run and must-off) ix. Crew constraints	Adaptive Lagrangian relaxation algorithm	C #	IEEE modified system of 3 area(2 sub area each) , 6 sub area , 3 tie-lines of bidirectional	No

2.6. Discussion on the findings of the review

The review considered the time period between 1958 and 2013 with thirty four different algorithms used in both single area and multi-area economic dispatch problems.

2.6.1 Review findings on the problem solution of single area, single criterion, multi-criteria, and multi-area dispatch problems

The review finds that various objective functions, and constraints are used in both single area and multi-area dispatch problems as follows:

The most used objective functions are:

1. Second order or cubic fuel cost function with or without valve point loading effect.
2. Second order or cubic emission function with or without valve point loading effect.
3. Second order or cubic combined fuel cost and emission functions with or without valve point loading effect.

The most used constraints are:

Real power balance constraint, transmission loss constraint, real power operation limits, real and reactive power flows, voltage limits, area power balance constraint, and tie-line power transfer limits constraint.

It is discussed that the solution of the dispatch problem is more accurate by considering valve point effect for both fuel cost and emission functions.

It is necessary to consider the transmission loss of the power system in order to achieve accurate real power balance constraint.

2.6.2. Review findings on the penalty factor of Multi-criteria, single area and multi-area dispatch problems

The bi-objective function (fuel cost and emission) is converted into single objective function by the following penalty factors: Weighted sum method in (Rani et al., 2006), (Jeyakumar et al., 2006), (Jeyakumar et al., 2007), (Wong et al., 19993), and (Ozyon et al., 2009); Max-Max price penalty factor in (Dhillon et al., 1994), (Tsai and Yen 2010), (Hemamalini and Simon, 2009), (Thankur et al., 2006), and (Ming et al., 2010); Min-Max price penalty factor (Krishnamurthy and Tzoneva, 2011); Average price penalty factor, and Common price penalty factor in (Balamurugan and Subramanian, 2008).

Weighted sum method is the simplest multi-criteria decision analysis for evluating a number of alternatives in terms of a number of decision criteria. But it is applicable only when all the data are expressed in same unit. It is easy to prove that the cost of the CEED problem solution is minimum and it depends on the user choosen appropriate weights. This method gives an idea of the shape of the fuel cost pareto surface and provides the user with more information about the trade-off among the various objectives.

The Max-Max penalty factor is commonly used in a multi-criteria dispatch problem to combine both cost and emission functions. The Min-Max penalty factor (Krishnamurthy and Tzoneva, 2011), yields better solution for the multi-criteria dispatch problem in comparison to other penalty factors. The average and common penalty factor (Balamurugan and Subramanian, 2008) are also used in the multi-criteria dispatch problem.

The penalty factor approach for conversion of the multi-criteria dispatch problem in a single criterion one yields a better objective function value and calculation time for combined economic emission dispatch problem in comparison with the weighted sum approach. Therefore the global solution of the multi-criteria economic dispatch problem depends on the penalty factors used.

2.6.3. Review findings on the Algorithms used in a single area and a multi-area dispatch problems

The findings on the algorithms used in the single area and multi-area dispatch problems are as follows:

Almost 51 references using PSO algorithm have been reviewed from the total of 180 references. The other most used algorithms are Evolutionary programming and Lagrange's method.

Nowadays the researchers are moving towards the heuristic algorithms such as (PSO-DE) Particle Swarm Optimization – Differential Evolution (Khamsawang et al., 2010), (HA) Heuristic Approach (Ouyang and Shahidehpour, 1991), (HDE) Hybrid Differential Evolution (Jayabharathi et al., 2008), (PSO-SQP) PSO with Sequential Quadratic Programming (Sugsakarn and Damrongkulkamjorn, 2008), (TCPSO) Tent-map based Chaotic PSO (Tao and Jin-ding, 2009)],(FPSO) Fuzzified Particle Swarm Optimization (Somasundaram and Swaroopan, 2011) and (Basu, 2013), (FMEP) Fuzzy Mutated Evolutionary Programming and (FGTS) Fuzzy Guided Tabu Search (Prasanna and Somasundaram, 2009) to obtain the global optimum solution and to reduce the computation time required to solve the dispatch problems.

2.7 Conclusion

This chapter reviews the existing methods and algorithms for solution of the single area and the multi-area dispatch problem using single criteria and multi-criteria formulations of the problems. The review investigates that different objective functions used in the single area and multi-area economic emission dispatch problems to evaluate the fuel cost, emission production and their combination.

Constraints such as real power balance, transmission loss, voltage limits, real power operation limits, area power balance, tie-line power transfer limits, real and reactive power flows are mostly used. The multi-area economic dispatch problem is formulated with tie-line real power coordinator. Its objective function is described either as single criteria or multi-criteria.

The review also analyzed the computation time and convergence properties obtained in the single area and multi-area power system economic dispatch problem solutions through the various optimization algorithms.

The future direction of research in the field of economic and combined economic dispatch problems can be identified as :

i. Multi-area problem formulation and solution by decomposition methods and algorithms.
ii. Innovative ways of application of the heuristics algorithms.
iii. Using of parallel computation to solve the non-linear high dimension dispatch problems in real-time.

In chapter 3, methods and algorithms for solution of the Combined Economic Emission Dispatch (CEED) problem using Lagrange's approach are proposed for various types of fuel cost and emission functions. The impact of seven different penalty factors over the optimisation solution and convergence of the calculation procedure for the CEED problem is evaluated.

CHAPTER THREE

COMBINED ECONOMIC EMISSION DISPATCH PROBLEM USING LAGRANGE'S ALGORITHM

3.1 Introduction

The objective of the economic dispatch problems of electrical power generation is to schedule the committed generating units output to meet the required load demand while satisfying the system equality and inequality constraints. Thermal power plants play a major role in power production. The resulting from the power production, pollutants such as sulphur dioxide (SO_2), Nitrogen oxide (NO_x) and carbon dioxide (CO_2) affect the environmental conditions. Three types of fuel cost and emission criterion functions are considered in this chapter. They are: i) quadratic model of the fuel cost and emission criterion functions without considering a valve point loading effect ii) quadratic model of the fuel cost and emission criterion functions with a valve point loading effect, and iii) cubic functions for both the fuel cost and emission criterion functions. The constraints such as generator limits, load balance and transmission loss are considered for all the three types of criterion functions described above. Lagrange's methods, algorithms, and softwares for the solution of the considered three types of Combined Economic Emission Dispatch (CEED) problems are developed in this Chapter.

The chapter breaks down into three sections: Section 3.2 formulates the CEED problem with the quadratic fuel cost and emission function, Section 3.3 formulates the CEED Problem with the valve point loading effect, Section 3.4 formulates the CEED problem with the cubic fuel cost and emission criterion functions.

3.2 Formulation and solution of the CEED problem for the case of quadratic functions of the fuel cost and emission

3.2.1 Economic dispatch problem with a quadratic fuel cost objective function

The formulation of an economic dispatch optimization problem can be written in the following way:

Minimize

$$F_C = \sum_{i=1}^{n} F_i\left(P_i\right) = \sum_{i=1}^{n} (a_i P_i{}^2 + b_i P_i + c_i) \qquad [\$/h] \qquad (3.1)$$

Where

F_C Total Fuel Cost

$F_i(P_i)$ Fuel cost of the i^{th} generator

P_i	Real power generation of a generator unit *i*
a_i , b_i, c_i	Cost coefficients of generating for the unit *i in* [\$/MW^{2}h], [\$/MWh] and [\$/h] respectively
n	Number of generating units

Under the constraints

i. Power balance constraint

$$\sum_{i=1}^{n} P_i = P_G = P_D + P_L \qquad \text{[MW]} \qquad (3.2)$$

Where

P_G	Total power generation of the system
P_D	Total demand of the system
P_L	Total transmission loss of the system

The transmission loss can be expressed as, (Dhillon et al, 1994)

$$P_L = \sum_{i=1}^{n}\sum_{j=1}^{n} P_i B_{ij} P_j + \sum_{i=1}^{n} B_{0i} P_i + B_{00} \qquad \text{[MW]} \qquad (3.3)$$

Where

P_i	Active power generation of unit *i*
P_j	Active power generation of unit *j*
B_{ij}, B_{0i}, B_{oo}	Transmission loss coefficients

ii. Generator operational constraints

$$P_{i,\min} \le P_i \le P_{i,\max}, i = \overline{1,n} \qquad \text{[MW]} \qquad (3.4)$$

Where

$P_{i,min}$	Minimum value of real power allowed at a generator *i*
$P_{i,max}$	Maximum value of real power allowed at a generator *i*

3.2.2 Bi-Criteria Dispatch Problem with quadratic fuel cost and emission functions

The various pollutants like sulphur dioxide, nitrogen oxide and carbon dioxide are the major waste emissions from the thermal power plants. The problem for minimization of the quantity of the emissions is formulated by including the reduction of waste emissions as an additional objective function by the following equation

$$E_T = \sum_{i=1}^{n}\left(d_i P_i^2 + e_i P_i + f_i\right) \qquad \text{[kg/h]} \qquad (3.5)$$

Where

E_T Total emission

d_i ,e_i, f_i Emission coefficients of generating unit i *in* [kg/MW^{2}h], [kg/MWh] and [kg/h] respectively

Then the CEED problem is determined by the objective functions (3.1) and (3.5).

A bi-objective optimization is converted into a single objective optimization (CEED) problem by introducing a price penalty factor h_i as follows:

$$F_T = \sum_{i=1}^{n}\left[(a_i P_i^{\,2} + b_i P_i + c_i) + h_i\left(d_i P_i^2 + e_i P_i + f_i\right)\right] \qquad \text{[\$/h]} \qquad (3.6)$$

Where

F_T CEED's fuel cost

The CEED problem is formulated by the criterion (3.6) and constraints (3.2) to (3.4).

3.2.3 Formulation of various price penalty factors for CEED problem

The authors (Su etal,1986) proposed a price penalty factor method for combined fuel cost and transmission line loss as follows:

$$F_C = \sum_{i=1}^{n} F_i\left(P_i\right) = \sum_{i=1}^{n}\left[(a_i P_i^{\,2} + b_i P_i + c_i) + h_i \sum_{m=1}^{M} R_m T_m^2\right]$$

where

R_m Resistance of the line

T_m Transmitted power of the line

M Total number of transmission lines

Subject to the constraints (i) Power balance and (ii) Generator operational limits given in Equation (3.2) and (3.4) respectively.

The procedure to calculate h_i is described by (Su et al,1986) as follows:

h_i is determined by evaluating the average cost of each generator at its maximum output (P_i,*max*)

$$h_i = \frac{F_C\left(P_{i,\max}\right)}{P_{i,\max}} = \frac{(a_i P_{i,\max}^2 + b_i P_{i,\max} + c_i)}{P_{i,\max}} \qquad i = \overline{1,n} \qquad \text{[\$/MWh]}$$

h_i is then arranged in ascending order. The maximum capacity of each unit $P_{i,max}$ is added, one at a time, starting from the smallest h_i unit, until $\sum_{i=1}^{n} P_i = P_G = P_D$. At this stage, the h_i associated with the last unit in the process is h.

The economic dispatch and emission dispatch are two different problems. Emission dispatch can be included in economic dispatch problem by the addition of emission cost to the normal dispatch. The bi-objective Combined Economic Emission Dispatch (CEED) problem can be converted into single objective optimization problem by introducing a price penalty factor h as follows:

$$F_{T_i} = w_1 F_{C_i} + w_2\ h_i\ E_{T_i}$$

$$if\ w_1 = 1\ and\ w_2 = 1,\ \ then$$

$$\mathrm{F}_{\mathrm{T}_i} = \mathrm{F}_{\mathrm{C}_i} + \mathrm{h}_\mathrm{i}\ \mathrm{E}_{\mathrm{T}_i}$$

where w_1 and w_2 are the weight factors, described as

(i) w_1=1 and w_2=0 for the pure economic dispatch problem

(ii) $w1$=0 and w_2=1 for the pure emission dispatch problem

(iii) w_1=1 and w_2=1 for the Combined Economic Emission Dispatch (CEED) problem

The authors (Palanichamy and Srikrishna, 1991) and (Kulkarni et al, 2000) followed the procedure proposed by (Su et al,1986) to find the price penalty factor for the CEED problem, the procedure of the method is given below:

step 1: Evaluate the average cost of each generator at its maximum output as follows:

$$\frac{F_{C_i}\left(P_{i,\max}\right)}{P_{i,\max}} = \frac{a_i P_{i,\max}^2 + b_i P_{i,\max} + c_i}{P_{i,\max}} \qquad [\$/MWh]$$

step 2: Evaluate the average emission cost of each generator at its maximum output as follows:

$$\frac{E_{T_i}\left(P_{i,\max}\right)}{P_{i,\max}} = \frac{d_i P_{i,\max}^2 + e_i P_{i,\max} + f_i}{P_{i,\max}} \qquad [kg/MWh]$$

step 3: Divide the average cost of each generator by its average emission as follows:

$$h_i = \frac{a_i P_{i,\max}^2 + b_i P_{i,\max} + c_i}{d_i P_{i,\max}^2 + e_i P_{i,\max} + f_i} \qquad [\$/kg]$$

step 4: Arrange h_i (i=1,2,….n) in ascending order.

step 5: Add the maximum capacity of each unit ($P_{i,max}$). one at a time, starting from smallest h_i unit, until $\sum_{i=1}^{n} P_{i,\max} \geq P_D$.

step 6: At this stage h_i , associated with the last unit in the process is the price penalty factor h [$/kg] for the given load.

The procedure given by (Palanichamy and Srikrishna, 1991) gives the approximate value of the price penalty factor for the particular load demand. Hence a modified price penalty factor is introduced by (Venkatesh et al, 2003) to give value of h_i for the particular load demand. The procedure for calculating the modified price penalty factor as follows:

Steps 1 to 4: The first four steps of h_i calculation remain the same as in the procedure proposed by (Palanichamy and Srikrishna, 1991), to calculate the modified price penalty factor.

step 5: The ascending order values of h_i calculated in step 4, are used to calculate the generator real power P_i for the particular load demand.

The modified price penalty factor procedure proposed by (Venkatesh et al, 2003) gives minimum fuel cost and CEED cost values in comparison with the procedure proposed by (Palanichamy and Srikrishna, 1991) but the transmission line loss is bigger.

The described procedure above, calculate the price penalty factor h_i using non-optimized generator real power in which the calculation is taken place outside the iterative process loop. While this Chapter proposed a new procedure to calculate the price penalty factor h_i for every iteration using the optimized generator real power in order to minimize the fuel cost, CEED fuel cost and transmission line losses.

The procedure to calculate the new price penalty factor is as follows:

Steps 1 to 3: The first three steps of h_i calculation remain the same as in the procedure proposed by (Palanichamy and Srikrishna, 1991), to calculate the new price penalty factor for CEED problem.

step 4: The h_i values calculated in step 3 are used to find the unknown values of the generator real power P_i for the particular load demand.

This method of calculating new price penalty factor gives minimum transmission loss, fuel cost and CEED fuel cost values in comparison with the modified price penalty factor proposed by (Venkatesh et al, 2003).

The price penalty factor is used to convert bi-objective function into a single objective function in CEED problem. The comparative study of various price penalty factors such as Max-Max, Min-Min, Average and Common is proposed by (Balamurugan and Subramanian, 2008). In addition to that ,this chapter proposes Min-Max and Max-Min price penalty factors.

The impact of four types of price penalty factors such as Min-Max, Max-Max, Min-Min and Max-Min are considered in this chapter to solve the CEED problem.

The ratio between the minimum fuel cost and maximum emission is called “Min-Max” penalty factor is described as:

$$h_i = \frac{(a_i P_{i,\min}^2 + b_i P_{i,\min} + c_i)}{\left(d_i\, P_{i,\max}^2 + e_i\, P_{i,\max} + f_i\right)} \qquad [\$/kg] \qquad (3.7)$$

The second price penalty factor called “Max-Max” is described as the ratio between the maximum fuel cost and maximum emission of the corresponding generator units as:

$$h_i = \frac{(a_i P_{i,\max}^2 + b_i P_{i,\max} + c_i)}{\left(d_i\, P_{i,\max}^2 + e_i\, P_{i,\max} + f_i\right)} \qquad [\$/kg] \qquad (3.8)$$

The third price penalty factor called “Min-Min” is described as the ratio between the minimum fuel cost and minimum emission of the corresponding generator units as:

$$h_i = \frac{(a_i P_{i,\min}^2 + b_i P_{i,\min} + c_i)}{\left(d_i\, P_{i,\min}^2 + e_i\, P_{i,\min} + f_i\right)} \qquad [\$/kg] \qquad (3.9)$$

The last price penalty factor called “Max-Min” is described as the ratio between the maximum fuel cost and minimum emission of the corresponding generator units as:

$$h_i = \frac{(a_i P_{i,\max}^2 + b_i P_{i,\max} + c_i)}{\left(d_i\, P_{i,\min}^2 + e_i\, P_{i,\min} + f_i\right)} \qquad [\$/kg] \qquad (3.10)$$

The role of all penalty factors is to transfer the physical meaning of the emission criterion from weight of the emission to the fuel cost for the emission. The difference between these penalty factors is in the weight of the fuel cost for emission in the final optimal fuel cost for generation and emission, criterion 3.6.

Comparison between the influences of the separate price factors over the optimization problem solution is done through application of the Lagrange’s method for calculation of this solution.

3.2.4 Lagrange's method for solution of the combined economic emission dispatch problem

In mathematical optimization, the method of Lagrange's multipliers (named after Joseph Louis Lagrange) provides a strategy for finding the local maxima or minima of a function subject to the equality or inequality constraints.

Method of Lagrange's is used to obtain the solution for the CEED problem with criterion given by the equation (3.6), subject to the constraints (3.2), (3.3) and (3.4). The problem is solved by introduction of a function of Lagrange based on a Lagrange's multiplier λ, as follows:

$$L = F_T + \lambda\left(P_D + P_L - \sum_{i=1}^{n} P_i\right) \tag{3.11}$$

$$L = \sum_{i=1}^{n}\left[(a_i P_i^{\,2} + b_i P_i + c_i) + h_i\left(d_i P_i^2 + e_i P_i + f_i\right)\right] + \lambda\left(P_D + \sum_{i=1}^{n}\sum_{j=1}^{n} P_i B_{ij} P_j + \sum_{i=1}^{n} B_{0i} P_i + B_{00} - \sum_{i=1}^{n} P_i\right) \tag{3.12}$$

The physical meaning of the Lagrange's multiplier λ is of a cost for fuel production. Then the physical meaning of the function of Lagrange is a cost for fuel production.

The optimization problem (3.6), subject to the constraints (3.2), (3.3) and (3.4) is transferred to the problem for minimization of L according to $P_{i,}\ i = \overline{1,n},$ and maximization of L according λ, under the constraints (3.4).

Necessary conditions for optimality for solution of the problem (3.4), (3.11) are, i.e:

According to P_i, $\dfrac{\partial L}{\partial P_i} = 0, i = \overline{1,n}$ (3.13)

According to λ, $\dfrac{\partial L}{\partial \lambda} = 0$ (3.14)

Derivation of the condition (3.13) is as follows:

$$\frac{\partial L}{\partial P_i} = 2a_i P_i + b_i + h_i\left(2d_i P_i + e_i\right) + \lambda\left(2\sum_{j=1}^{n} B_{ij} P_j + B_{0i} - 1\right) = 0, i = \overline{1,n} \tag{3.15}$$

$$\frac{\partial L}{\partial P_i} = \left(2a_i + 2h_i d_i\right)P_i + 2\lambda B_{ii} P_i + b_i + h_i e_i + \lambda\left(2\sum_{\substack{j=1 \\ j\neq i}}^{n} B_{ij} P_j + B_{0i} - 1\right) = 0, i = \overline{1,n} \tag{3.16}$$

$$\frac{\partial L}{\partial P_i} = \left(\frac{a_i + h_i d_i}{\lambda} + B_{ii} \right) P_i + \sum_{\substack{j=1 \\ j \neq i}}^{n} B_{ij} P_j + \frac{1}{2} \left(\frac{b_i + h_i e_i}{\lambda} + B_{0i} - 1 \right) = 0, \; i = \overline{1,n} \tag{3.17}$$

The last equation can be written as

$$\left(\frac{a_i + h_i d_i}{\lambda} + B_{ii} \right) P_i + \sum_{\substack{j=1 \\ j \neq i}}^{n} B_{ij} P_j + \frac{1}{2} \left(1 - \frac{b_i + h_i e_i}{\lambda} - B_{0i} \right), \; i = \overline{1,n} \tag{3.18}$$

Equation (3.18) can be written in a matrix –vector form as follows

$$\begin{bmatrix} \frac{a_1 + h_1 d_1}{\lambda} + B_{11} & B_{12} & B_{1n} \\ B_{21} & \frac{a_2 + h_2 d_2}{\lambda} + B_{22} & B_{2n} \\ B_{n1} & B_{n2} & \frac{a_{1n} + h_n d_n}{\lambda} + B_{nn} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 - \left(\frac{b_1 + h_1 e_1}{\lambda} \right) - B_{01} \\ 1 - \left(\frac{b_1 + h_1 e_1}{\lambda} \right) - B_{02} \\ 1 - \left(\frac{b_1 + h_1 e_1}{\lambda} \right) - B_{0n} \end{bmatrix} \tag{3.19}$$

Its short form is

$EP=D$ (3.20)

If the value of the Lagrange's multiplier λ is known the equation (3.19), or (3.20) can be solved according to the unknown vector P, using the Matlab command

$P=E\backslash D$ (3.21)

The value of λ is unknown and has to be found from the necessary condition for optimality $\frac{\partial L}{\partial \lambda} = 0$.

This condition is:

$$\frac{\partial L}{\partial \lambda} = \left(P_D + \sum_{i=1}^{n} \sum_{j=1}^{n} P_i B_{ij} P_j + \sum_{i=1}^{n} B_{0i} P_i + B_{00} - \sum_{i=1}^{n} P_i \right) = 0 = \Delta\lambda \tag{3.22}$$

Equation (3.22) is not a function of λ, but represents the gradient of L according to λ. At the optimal solution this gradient has to be equal to zero. As analytical solution for λ is not possible a gradient procedure for calculation of λ has to be developed as follows:

$$\lambda^{(k+1)} = \lambda^k + \alpha \Delta\lambda^{(k)}, \lambda \neq 0 \tag{3.23}$$

Where $\Delta\lambda^{(k)}$ is determined by the Equation (3.22) and α is the step of the gradient procedure. This procedure will start with some given initial value of the Lagrange's variable $\lambda^{(0)}$. When during the iterations $\Delta\lambda = 0$, the optimal solution for the Lagrange's variable is obtained. It will determine the optimal solution for the energy that has to be produced by the generators as a solution of (3.21).

The obtained solutions for $P_i,\ i=\overline{1,n}$ have to belong to the constraint domain (3.4). That is why for every step of the gradient procedure, the obtained solution is fit to the constraint domain following the procedure

$$P_i^{(k)} = \begin{pmatrix} P_{i,\min}, & if \ \ P_i^{(k)} \langle P_{i,\min} \\ P_i^{(k)}, & if \ \ P_{i.\min} \le P_i^{(k)} \langle P_{i,\max} \\ P_{i,\max}, & if \ \ P_i^{(k)} \rangle P_{i,\max} \end{pmatrix} \tag{3.24}$$

The condition for end of the iterations is

$$\Delta\lambda^{(k)} \le \varepsilon, \text{ or } \quad k = Iter_{Max} \tag{3.25}$$

Where $\varepsilon \rangle 0$ is a small number, and $Iter_{Max}$ is the given maximum number of iterations.

The algorithm of the method is:

1) Initial value of the Lagrange's multiplier is guessed: $\lambda^{(0)}$, and the value of the condition for optimality ε is given.
2) In Equation (3.19) matrix $E^{(0)}$ and $D^{(0)}$ are formed
3) Equation (21) is solved and $P^{(0)} = E^{(0)} \backslash D^{(0)}$ is determined.
4) The obtained vector $P^{(0)}$ is fit to the constraint domain (3.24).
5) $\Delta\lambda^{(0)}$ is calculated using Equation (3.22) where $P^{(0)}$ is substituted.
6) The condition (3.25) is checked. If it is fulfilled the calculations stop, if not, improved value of $\lambda \to \lambda^{(1)}$ is calculated using Equation (3.23)
7) Calculations of the improved values of $P_i \to P_i^{(1)}$ is done as in Equation (3.21) and so on. Iterations continue until condition (3.25) is satisfied or the maximum number of iterations is reached.
8) The optimal solution is used to calculate the total cost, using Equation (3.6).

The flowchart of the algorithm is shown in Figure 3.1.

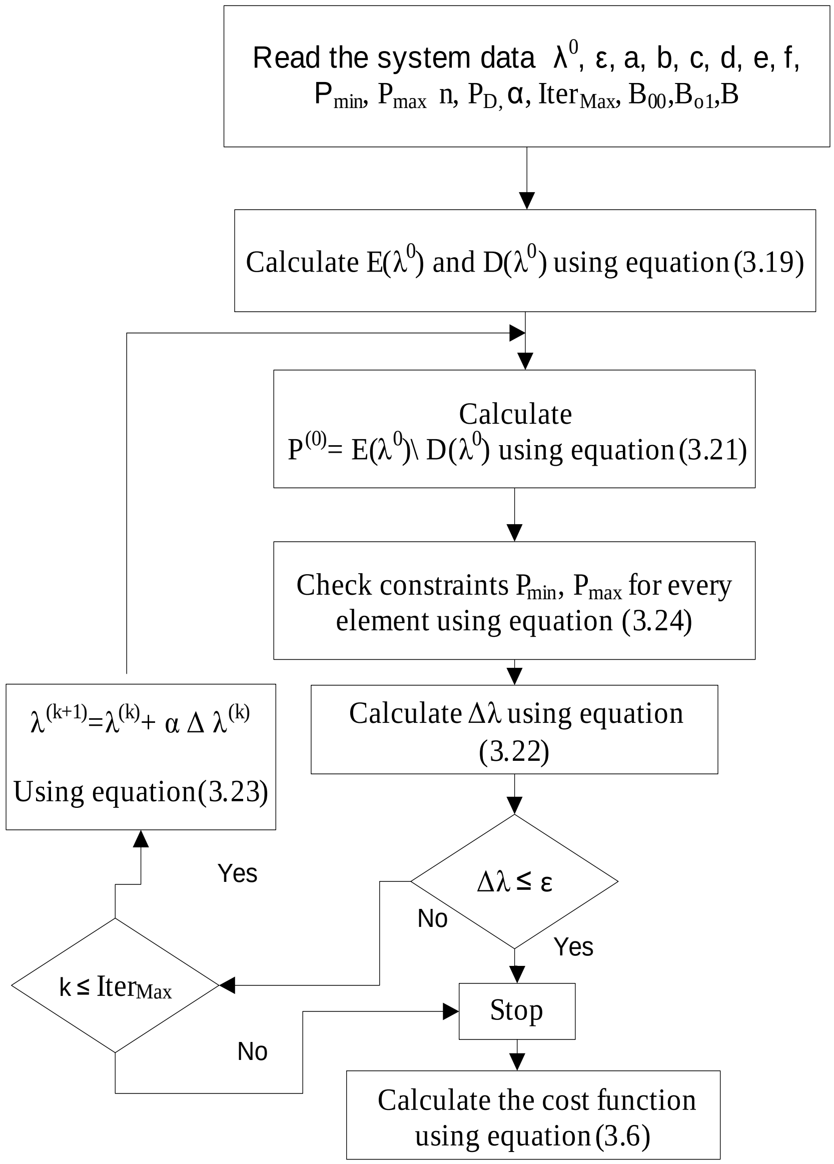


Figure 3.1: **Flow diagram of the bi-criteria dispatch problem algorithm solution based on quadratic optimization functions and using the Lagrange's method**

The software developed for CEED problem is given in Appendix A. It can be seen that the price penalty factor values influence the values of the matrices *E* and *D* in Equation 3.19 and in this way the components of the optimal vector *P* of the active power received by each generator are different for each type of price penalty factors. The dispatch problem can be solved for different expressions of the penalty factors in a process of search of the best solution.

3.2.5 Simulation Results

The proposed dispatch problem is solved for IEEE 30 bus system (Gnanadass,2005), an Indian utility system (Dhillon et al, 1993), IEEE 118 bus system (Guerrero,2004) and 11 generator system (Balamurugan and Subramanian,2007).

3.2.5.1 Test System 1

Table 3.1 represents the IEEE 30 bus system data (Gnanadass, 2005). The system has six generating units. The CEED problem is solved for the cases of various price penalty factors. The different power demand values are introduced. Various levels of disturbance in order to evaluate the solution for the considered power demands are: P_D= [125; 150; 175; 200; 225; 250] MW.

The CEED problem is solved in MATLAB R2011a version. The software is given in Appendix A1 and the MATLAB script file is named as CEED_Casestudy1.m . The four types of price penalty factors introduced in part 3.2.3 are considered. The IEEE 30 bus system data such as fuel cost coefficients, emission coefficients, generator limits, initial transmission loss coefficients B_{00}, transmission loss coefficients B_{01} and B_{ij} are given in Table 3.1.

The Solution of the bi-criteria economic dispatch problem is shown in Table 3.2, it consists of the real power outputs of the generators in MW, fuel cost in [$/hr] and emission values in [kg/hr] for the IEEE 30 bus system. The initial lambda (λ^0) value is assumed as 8. The maximum number of Iterations, $Iter_{Max}$=2000.The optimal solution is computed for the selected load demand. The software developed is given in Appendix A1.

Figure 3.2 shows the one line diagram of IEEE 30 bus system (Gnanadass, 2005) .

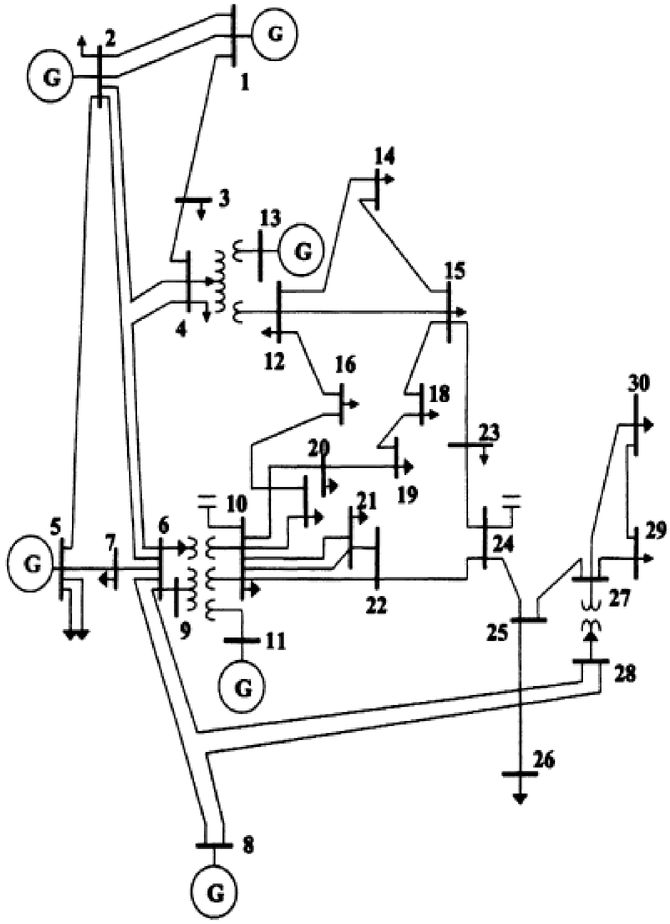


Figure 3.2: **IEEE 30 bus system (Gnanadass, 2005)**

Table 3.1: **IEEE 30 Bus System Data**

Bus Number	Generator limits[MW]		Fuel cost coefficients			Emission coefficients		
	P_{max}	P_{min}	a_i [$/MW^{2}h]	b_i [$/MWh]	c_i [$/h]	d_i [$/kg^{2}h]	e_i [$/kgh]	f_i [$/h]
1	200	50	0.00375	2.00	0.00	0.0126	-0.90	22.983
2	80	20	0.01750	1.70	0.00	0.0200	-0.10	25.313
5	50	15	0.06250	1.00	0.00	0.0270	-0.01	25.505
8	35	10	0.00834	3.25	0.00	0.0291	-0.005	24.900
11	30	10	0.02500	3.00	0.00	0.0290	-0.004	24.700
13	40	12	0.02500	3.00	0.00	0.0271	-0.0055	25.300

<table>
<tr><th colspan="9">Transmission loss coefficients</th></tr>
<tr><th colspan="3">B_{01}</th><th colspan="6">B</th></tr>
<tr><td rowspan="2">0.000003</td><td rowspan="2">0.000021</td><td rowspan="2">-0.00056</td><td>0.000218</td><td>0.000103</td><td>0.000009</td><td>-0.000010</td><td>0.000002</td><td>0.000027</td></tr>
<tr><td>0.000103</td><td>0.000181</td><td>0.000004</td><td>-0.000015</td><td>0.000002</td><td>0.000030</td></tr>
<tr><td rowspan="2">0.000034</td><td rowspan="2">0.000015</td><td rowspan="2">0.000078</td><td>0.000009</td><td>0.000004</td><td>0.000417</td><td>-0.000131</td><td>-0.000153</td><td>-0.000107</td></tr>
<tr><td>0.000010</td><td>-0.000015</td><td>-0.000131</td><td>0.000221</td><td>0.000094</td><td>0.000050</td></tr>
<tr><td colspan="3" rowspan="2">B_{00}=0.000014</td><td>0.000002</td><td>0.000002</td><td>-0.000153</td><td>0.000094</td><td>0.000243</td><td>-0.000001</td></tr>
<tr><td>0.000027</td><td>0.000030</td><td>-0.000107</td><td>0.000050</td><td>-0.000001</td><td>0.0003458</td></tr>
</table>

In Table 3.2 the CEED fuel cost is calculated by the formulae $F_T = \sum_{i=1}^{n}\left(F_{c_i} + h_i E_{T_i}\right)$ in [$/h], where h_i is the price penalty factor calculated as $h_i = \frac{F_{C_i}(P_i \min)}{E_{T_i}(P_i \max)}$, $i = \overline{1,n}$ [$/kg], Equation (3.7) and CT is the Calculating Time for the solution of the problem

Table 3.2: **The Bi-criteria dispatch problem solution using Min-Max price penalty factor**

P_D [MW]	λ	P_1 [MW]	P_2 [MW]	P_3 [MW]	P_4 [MW]	P_5 [MW]	P_6 [MW]	P_L [MW]	F_C [$/h]	E_T [Kg/h]	F_T [$/h]	Number of Iterations	CT [s]
125	2.6643	59.2499	20.0000	15.0000	10.0000	10.0000	12.0000	1.2499	308.1608	144.5304	377.1857	195	2.1216
150	3.0287	78.8314	26.2653	15.0000	10.0000	10.0000	12.0000	2.0967	373.5001	162.2299	448.5151	153	1.5965
175	3.3682	96.7116	32.8891	16.5112	10.0000	10.0000	12.0000	3.1119	443.9670	190.4187	528.5570	139	1.3095
200	3.7044	114.0699	39.3656	18.8704	10.0000	10.0000	12.0000	4.3059	519.5033	228.1548	616.9529	129	1.2071
225	3.9709	127.5965	44.4446	20.7448	14.3840	11.2166	12.0000	5.3865	601.1024	268.0758	713.0526	75	0.6671
250	4.1980	138.9680	48.7367	22.3456	18.8093	13.7143	13.8597	6.4337	687.0737	310.2842	815.1852	85	0.8626

In Table 3.3 the CEED fuel cost is calculated by the formulae $F_T = \sum_{i=1}^{n}\left(F_{c_i} + h_i E_{T_i}\right)$ in [$/h], where h_i is the price penalty factor calculated as $h_i = \frac{F_{C_i}(P_i \max)}{E_{T_i}(P_i \max)}$, $i = \overline{1,n}$ [$/kg], Equation (3.8).

Table 3.3: **The Bi-criteria dispatch problem solution using Max-Max price penalty factor**

P_D [MW]	λ	P_1 [MW]	P_2 [MW]	P_3 [MW]	P_4 [MW]	P_5 [MW]	P_6 [MW]	P_L [MW]	F_C [$/h]	E_T [Kg/h]	F_T [$/h]	Number of Iterations	CT [s]
125	3.2434	59.2499	20.0000	15.0000	10.0000	10.0000	12.0000	1.2499	308.1608	144.5304	619.2590	612	1.2622
150	4.3942	79.5289	25.5782	15.0000	10.0000	10.0000	12.0000	2.1071	373.4835	162.2107	716.1636	413	0.4531
175	5.0946	91.6419	31.7901	17.0781	13.4045	11.9357	12.0000	2.8502	447.4342	186.5649	835.6432	272	0.3182
200	5.6417	100.9851	36.5995	19.2509	17.3134	14.8896	14.5196	3.5581	527.9257	214.4270	969.9120	247	0.2791
225	6.1847	110.1595	41.3375	21.4101	21.1780	17.8129	17.4465	4.3444	612.0186	247.2931	1117.7000	243	0.332
250	6.7333	119.3263	46.0870	23.5931	25.0658	20.7567	20.3863	5.2153	699.5472	285.3648	1279.1000	238	0.343

In Table 3.4 the CEED fuel cost is calculated by the formulae $F_T = \sum_{i=1}^{n}\left(F_{c_i} + h_i E_{T_i}\right) in$ [$/h], where h_i is the price penalty factor calculated as $h_i = \frac{F_{C_i}(P_i \min)}{E_{T_i}(P_i \min)}, i = \overline{1,n}$ [$/kg], Equation (3.9).

Table 3.4: The Bi-criteria dispatch problem solution using Min-Min price penalty factor

P_D [MW]	λ	P_1 [MW]	P_2 [MW]	P_3 [MW]	P_4 [MW]	P_5 [MW]	P_6 [MW]	P_L [MW]	F_C [$/h]	E_T [Kg/h]	F_T [$/h]	Number of Iterations	CT [s]
125	4.1304	50.0000	25.9840	18.0185	10.1153	10.0000	12.0000	1.1177	310.8005	149.3334	606.337	382	0.3509
150	4.7914	50.0000	32.7289	21.7591	17.5298	15.2409	14.0981	1.3569	395.0747	171.2493	718.3321	277	0.4787
175	5.3818	50.0000	38.7247	25.1093	24.1084	20.0523	18.6623	1.6571	486.0034	199.8198	845.5137	275	0.4875
200	5.9764	50.0000	44.7350	28.4911	30.6914	24.8733	23.2261	2.0169	582.7623	235.5157	987.5126	277	0.5305
225	6.6608	50.0000	51.6180	32.3929	35.0000	30.0000	28.4384	2.4492	686.3078	277.5522	1144.8000	449	0.534
250	7.7624	50.0000	62.6194	38.6929	35.0000	30.0000	36.7388	3.0511	800.2705	327.6648	1325.3000	378	0.545

In Table 3.5 the CEED fuel cost is calculated by the formulae $F_T = \sum_{i=1}^{n}\left(F_{c_i} + h_i E_{T_i}\right) in$ [$/h], where h_i is the price penalty factor calculated as $h_i = \frac{F_{C_i}(P_i \max)}{E_{T_i}(P_i \min)}, i = \overline{1,n}$ [$/kg], Equation (3.10).

Table 3.5: The Bi-criteria dispatch problem solution using Max-Min price penalty factor

P_D [MW]	λ	P_1 [MW]	P_2 [MW]	P_3 [MW]	P_4 [MW]	P_5 [MW]	P_6 [MW]	P_L [MW]	F_C [$/h]	E_T [Kg/h]	F_T [$/h]	Number of Iterations	CT [s]
125	7.1899	50.0000	20.0000	15.0000	14.1427	14.8777	12.0000	1.0204	317.8361	148.1861	1749.2000	1140	2.423
150	9.3366	50.0000	21.2648	18.2209	21.7146	22.1223	17.9171	1.2397	408.3563	171.7818	1957.7000	703	1.734
175	11.1257	50.0000	25.7111	21.8670	27.9762	28.1204	22.8706	1.5482	500.9697	202.0717	2213.5000	763	0.834
200	13.3933	50.0000	31.3237	26.5019	35.0000	30.0000	29.0965	1.9221	596.6819	238.1609	2517.6000	1359	2.632
225	16.9178	50.0000	39.9969	33.7344	35.0000	30.0000	38.6594	2.3907	702.0336	278.2113	2897.2000	1540	4.367
250	22.2824	50.0000	53.0827	44.8046	35.0000	30.0000	40.0000	2.8874	818.3178	326.4815	3387.6000	1850	3.532

Table 3.6 presents the comparison of simulation results of the bi-criteria dispatch problem using various price penalty factor approaches. The comparison is done for the maximum power demand P_D = 250 [MW] because the biggest differences in the values of the criteria appeared for the maximum power demand. The values of the criteria of Min-Max penalty factor are taken for a basis and the differences for the other penalty factors are given in percentage.

Table 3.6: Comparison of the simulation results for the various price penalty factors

Criterion	Min-Max price penalty factor approach	Max-Max price penalty factor approach	Min-Min price penalty factor approach	Max-Min price penalty factor approach
CEED fuel cost F_T [$/h]	100 %	156.90 %	162.57 %	415.56 %
Fuel cost F_C [$/h]	100 %	101.81 %	116.47 %	119.10 %
Emission value E_T [kg/h]	100 %	91.96 %	105.60 %	105.22 %
Power loss P_L [MW]	100 %	81.06 %	47.42 %	44.87 %

Figure 3.3 shows the CEED problem fuel cost values for the various price penalty factors using Lagrange's method for optimization. In Figure 3.3, x-axis represents power demand ranges from 125 to 250 [MW]. The y-axis represents the fuel cost values of the CEED problem. Figure 3.4 shows that the CEED fuel cost values are less when using Min-Max price penalty factor in comparison with the other price penalty factors for the solution of bi-criteria dispatch problem. Figure 3.5 shows the calculated emission values of the environmental dispatch problem for the used various price penalty factors. Figure 3.5 shows that the emission values are less when using Max-Max price penalty factor in comparison with the other price penalty factors for the bi-criteria dispatch problem.

In Figure 3.6, the power loss values of the bi-criteria dispatch problem for the various price penalty factors using Lagrange's method are shown. It can be seen that the power loss values are less when using Max-Min price penalty factor for the bi-criteria dispatch problem in comparison with the power loss result obtained by using the other price penalty factors.

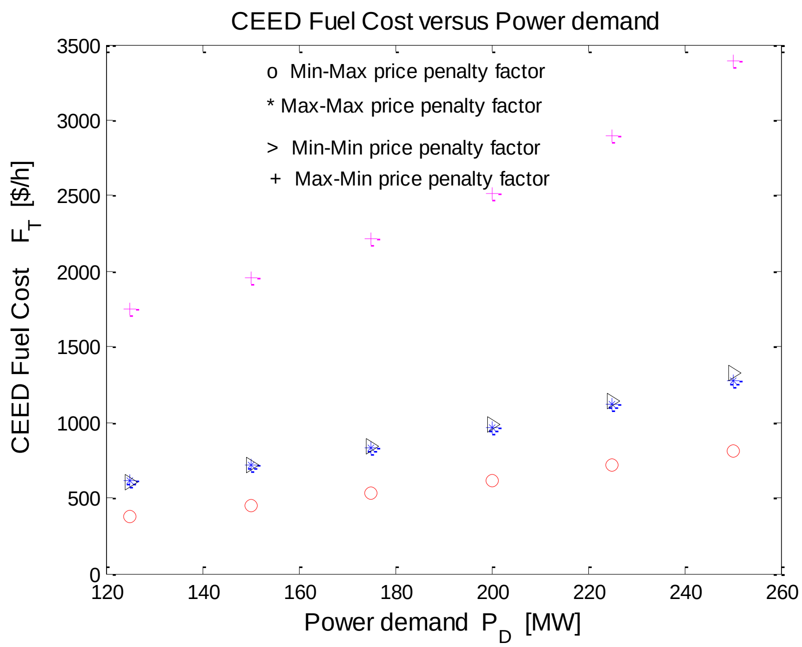


Figure 3.3: CEED fuel cost values of the bi-criteria dispatch problem using various price penalty factors

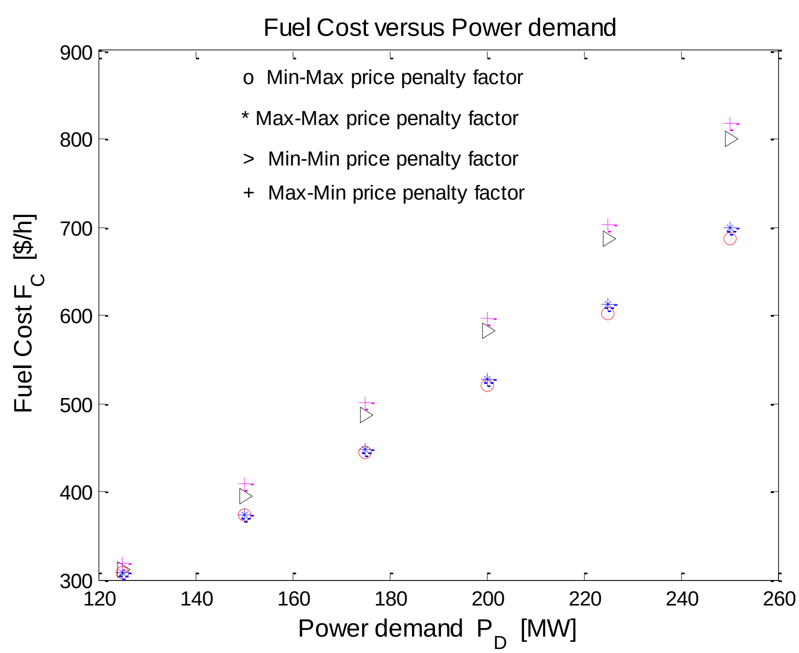


Figure 3.4: Fuel cost values of the economic dispatch problem using various price penalty factors

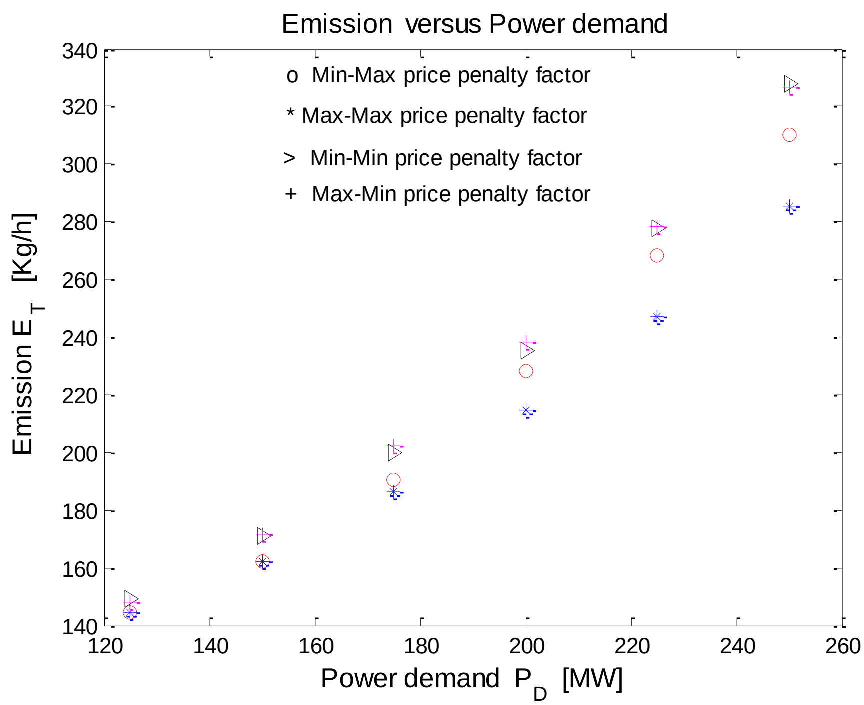


Figure 3.5: Emission values of the environmental dispatch problem using various price penalty factors

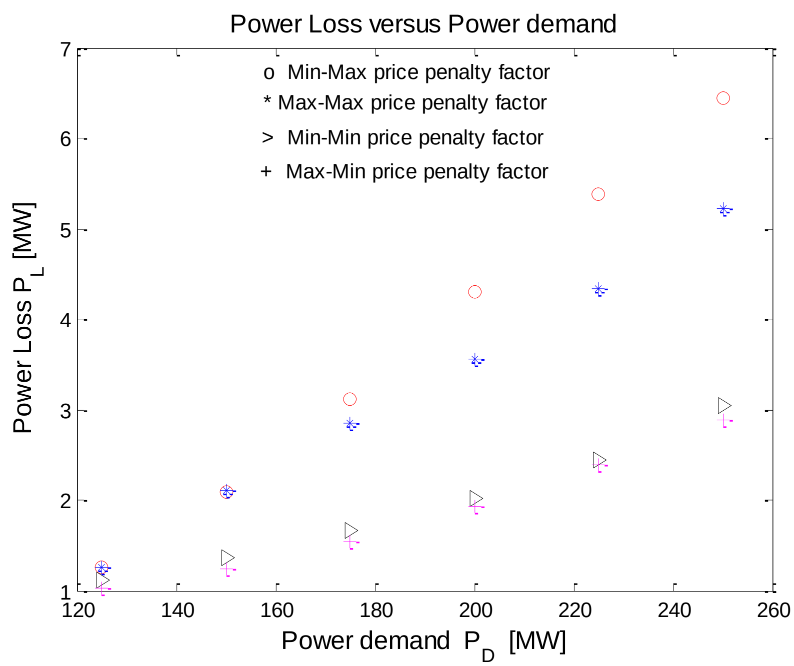


Figure 3.6: Power loss values of the bi-criteria dispatch problem using various price penalty factors

Figure 3.7 shows the Calculating time of the bi-criteria dispatch problem. Figure 3.8 and 3.9 shows the Lagrange's optimization values of lambda and deltalambda for bi-criteria dispatch problem for initial lambda of 4 and 20 respectively.

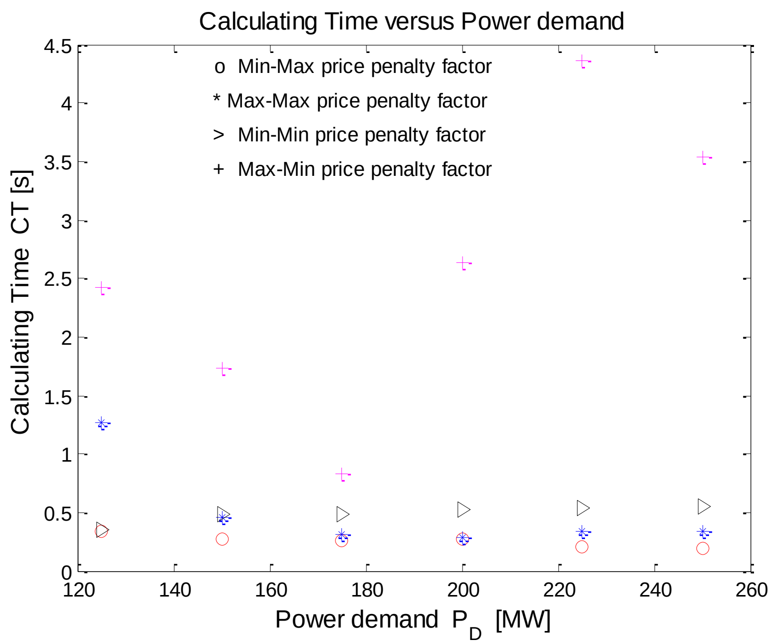


Figure 3.7: **Calculating Time values of the bi-criteria dispatch problem using various price penalty factors**

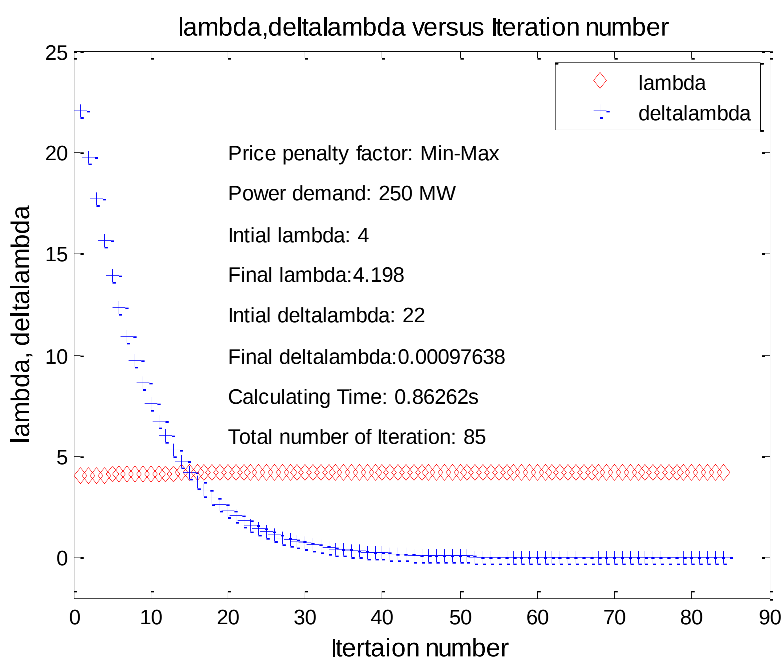


Figure 3.8: **Lagrange's optimization values of lambda and deltalambda for bi-criteria dispatch problem using initial value of lambda equal to 4**

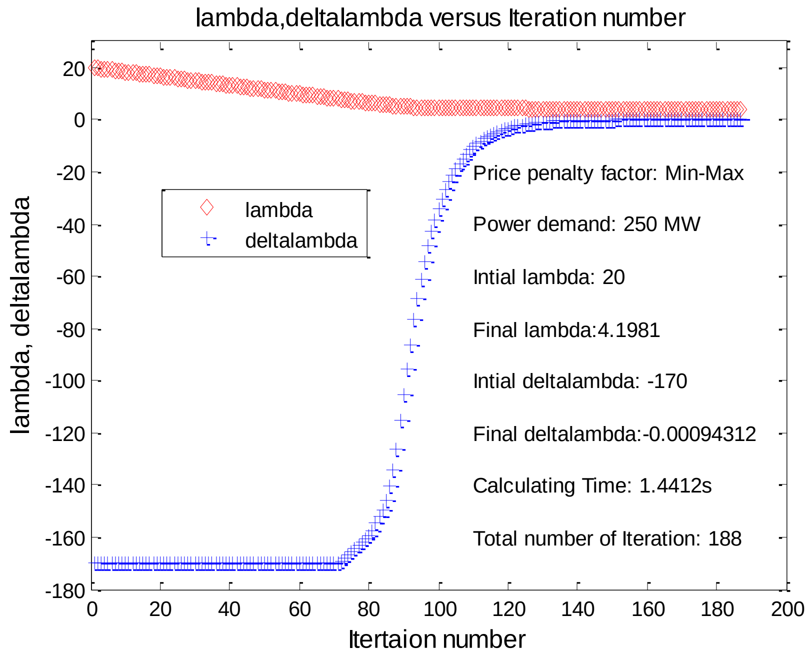


Figure 3.9: **Lagrange's optimization values of lambda and deltalambda for bi-criteria dispatch problem using initial value of lambda equal to 20**

3.2.5.2 Test System 2:

The CEED problem for an Indian utility system with six generators is solved in (Dhillon, Parti and Kothari, 1993) using fuzzy logic theory. The data such as fuel cost coefficients, emission coefficients, real power limits and transmission loss coefficients are given in Table 3.7. Table 3.8 shows the optimum generator scheduling of the Indian utility system calculated on the basis of the developed in section 3.2.4 Lagrange's method. Table 3.9 shows the comparison of the economic emission dispatch solutions using Lagrange's and Fuzzy logic algorithms. The best solutions such for the power loss, fuel cost, emission, combined economic emission dispatch fuel cost and computation time are highlighted in bold for various power demand of 500, 700 and 900 MW respectively. The software developed CEED_Casestudy2.m is given in Appendix A2.

Table 3.7: Economic dispatch data for a six generators Indian utility system

Fuel cost coefficients			Emission coefficients			Generator real power limits	
a_i [$/MW^{2}h]	b_i [$/MWh]	c_i [$/h]	d_i [$/kg^{2}h]	e_i [$/kgh]	f_i [$/h]	*Pmin [MW]*	*Pmax [MW]*
0.15247	38.53973	756.79886	0.00419	0.32767	13.85932	10	125
0.10587	46.15916	451.32513	0.00419	0.32767	13.85932	10	150
0.02803	40.39655	1049.99770	0.00683	-0.54551	40.26690	35	225
0.03546	38.30553	1243.53110	0.00683	-0.54551	40.26690	35	210
0.02111	36.32782	1685.56960	0.00461	-0.51116	42.89553	130	325
0.01799	38.27041	1356.65920	0.00461	-0.51116	42.89553	125	315

Transmission loss coefficients B					
0.002022	-0.000286	-0.000534	-0.000565	-0.000454	-0.000103
-0.000286	0.003243	0.000016	-0.000307	-0.000422	-0.000147
-0.000533	0.000016	0.002085	0.000831	0.000023	-0.000270
-0.000565	-0.000307	0.000831	0.001129	0.000113	-0.000295
-0.000454	-0.000422	0.000023	0.000113	0.000460	-0.000153
0.000108	-0.000147	-0.000270	-0.000295	-0.000153	0.000898

Table 3.8: **Indian utility system power generation scheduling and lambda values using various price penalty factors - Lagrange's method**

h_i	Min-Max Penalty factor method			Max-Max Penalty factor method			Min-Min Penalty factor method			Max-Min Penalty factor method		
P_D [MW]	500	700	900	500	700	900	500	700	900	500	700	900
Lambda	325.1127	487.2371	666.6109	87.2602	114.9876	144.9562	60.4543	73.0457	87.5158	110.5526	169.094	245.8591
P1 [MW]	46.8783	88.0155	125.0000	47.7648	83.4961	120.7504	62.4352	97.2627	125.0000	65.9707	122.2162	125.0000
P2 [MW]	25.4463	53.4651	80.938	32.6458	60.0234	86.5274	41.9516	68.9096	96.7687	51.8101	92.1086	132.3806
P3 [MW]	72.8025	89.0485	104.6206	67.7896	86.1267	102.6653	55.4074	74.8731	92.8588	68.6969	91.7727	115.1214
P4 [MW]	83.5137	108.5951	135.5438	82.1638	109.9663	138.6262	85.0241	118.9153	155.2812	80.7520	116.7905	160.8612
P5 [MW]	154.5583	213.6081	279.1968	155.5861	216.882	282.6253	149.3897	208.4449	275.5615	130.0000	171.9969	245.6828
P6 [MW]	143.3666	187.7282	232.8189	137.7595	183.2229	228.5141	125.0000	167.4083	214.1671	125.0000	149.6032	200.9753

Table 3.9: Comparison of the dispatch solution for the Indian utility system using Lagrange's and fuzzy logic theory

Criterion function	P_D	Fuzzy logic theory Case 1 (Dhillon, Parti, and Kothari, 1993)	Fuzzy logic theory Case 2 (Dhillon, Parti, and Kothari, 1993)	Lagrange's method			
				Penalty factors			
				Min-Max	Max-Max	Min-Min	Max-Min
Power loss P_L [MW]	500	**17.162**	20.799	19.2142	25.0995	22.3205	28.0221
	700	**34.927**	39.669	35.8143	40.6562	44.5003	41.3703
	900	**54.498**	65.032	59.6461	61.1746	80.3786	61.1068
Fuel cost F_C [INR/h]	500	28550.15	28476.63	28440.81	**28355.94**	28774.79	28413.15
	700	39070.74	39010.74	39030.09	**38765.85**	40555.89	38799.72
	900	50807.24	50854.86	50605.8	50252.83	52401.4	**50196.31**
Emission E_T [kg/h]	500	312.513	287.483	280.3481	279.2457	**276.9582**	280.8189
	700	528.447	493.977	474.5622	478.7498	**474.4816**	479.3919
	900	864.060	800.629	770.8086	773.729	**770.1269**	771.1824
CEED fuel cost F_T [INR/h]	500	-------------------	-------------------	**32180.57**	42170.55	54276.45	130151.0
	700	--------------------	--------------------	**45542.65**	62459.32	82743.99	211694.3
	900	---------------------	---------------------	**61525.97**	88426.75	124597.7	325503.9
Time for calculation C_T [s]	500	-------------------	-------------------	**7.69**	8.708	35.276	36.065
	700	--------------------	--------------------	**8.192**	8.671	31.753	50.317
	900	---------------------	---------------------	**10.048**	10.826	33.115	65.702

3.2.5.3 Discussion of the results for the Indian utility system:

The solution of CEED problem using Lagrange's method is compared with (Dhillon, Parti and Kothari,2009), Table 3.9. But the authors (Dhillon, Parti and Kothari,2009) solved the Economic Emission Dispatch (EED) problem for separate fuel cost and emission function. The solution of Lagrange's method gives less fuel cost and emission values in comparison with the fuzzy logic method (Dhillon, Parti and Kothari,2009),Table 3.9. In addition to that, CEED problem using various price penalty factors considered in Lagrange's method, the solution is given in Table 3.9.

3.2.5.4 Test System 3:

The CEED problem for an IEEE 118 bus system with fourteen generators is solved using Min-Max and Max-Max price penalty factors using Lagrange's algorithm. The data such as fuel cost coefficients, emission coefficients, generator real power limits and transmission loss coefficients (Guerrero, 2004) are given in Table 3.10 and 3.11 respectively. The optimum generator scheduling of the IEEE118 bus system is calculated on the basis of the Lagrange's method and given in Table 3.12. The Lagrange's solution of the CEED Problem is compared with Evolutionary Programming based Modified Combined Economic Emission Dispatch (EPMCEED) algorithm (Venkatesh, et al., 2003), Table 3.13. The Lagrange's solution using Max-Max price penalty factor for fuel cost, emission and CEED fuel cost is less in comparison with the EPMCEED solution . The transmission loss value is less in EPMCEED in comparison with the Lagrange's solution. The load level of 3668 [MW] is considered. The software developed *CEED_Casestudy3.m* is given in *Appendix A3*.

Table 3.10: Fuel cost and emission coefficients of IEEE 118 bus system

Fuel cost coefficients			Emission coefficients			Generator real power limits	
a_i [$/MW^{2}h]	b_i [$/MWh]	c_i [$/h]	d_i [$/lbs^{2}h]	e_i [$/lbsh]	f_i [$/h]	*Pmin [MW]*	*Pmax [MW]*
0.0050	1.89	150.000	0.016	-1.500	23.333	50	1000
0.0055	2.00	115.000	0.031	-1.820	21.022	50	1000
0.0060	3.50	40.000	0.013	-1.249	22.050	50	1000
0.0050	3.15	122.000	0.012	-1.355	22.983	50	1000
0.0050	3.05	125.000	0.020	-1.900	21.313	50	1000
0.0070	2.75	120.000	0.007	0.805	21.900	50	1000

0.0070	3.45	70.000	0.015	-1.401	23.001	50	1000
0.0070	3.45	70.000	0.018	-1.800	24.003	50	1000
0.0050	2.45	130.000	0.019	-2.000	25.121	50	1000
0.0050	2.45	130.000	0.012	-1.360	22.990	50	1000
0.0055	2.35	135.000	0.033	-2.100	27.010	50	1000
0.0045	1.60	200.000	0.018	-1.800	25.101	50	1000
0.0070	3.45	70.000	0.018	-1.810	24.313	50	1000
0.0060	3.89	45.000	0.030	-1.921	27.119	50	1000

P_D = 3668 [MW]

Where

The international System of Unit (SI) for mass is the kilogram.

1 kilogram is equal to 2.20462262185 lbs.

Table 3.11: Transmission loss coefficients of IEEE 118 bus system

The initial transmission loss coefficient B_{00} is 0.028378

B Coefficients (Matrix)													
-0.283225	-0.192940	-0.264240	0.017755	0.021917	0.040508	0.012216	0.014007	0.004407	0.032732	0.217820	0.032560	0.155630	-0.283225
0.030108	0.019242	0.021506	-0.002880	-0.004000	-0.004470	-0.002720	-0.003230	-0.006940	-0.007450	-0.019520	-0.012170	-0.017180	0.030108
0.037946	0.020710	0.020912	-0.003630	-0.005250	-0.004480	-0.003660	-0.003590	-0.006950	-0.010180	-0.020040	-0.018440	-0.020570	0.037946
0.020710	0.026780	0.024696	-0.002470	-0.003780	-0.002980	-0.002390	-0.002310	-0.004670	-0.007860	-0.015830	-0.015290	-0.016680	0.020710
0.020912	0.024696	0.024393	-0.002320	-0.003520	-0.003090	-0.002230	-0.002300	-0.004750	-0.007150	-0.016000	-0.134600	-0.015880	0.020912
-0.003630	-0.002470	-0.002320	0.009543	0.003659	0.002951	0.003740	0.003341	0.002486	0.001192	-0.002790	-0.002880	-0.003310	-0.003630
-0.005250	-0.003780	-0.003520	0.003659	0.010678	0.005763	0.003740	0.003341	0.002486	0.001192	-0.002790	-0.002880	-0.003310	-0.005250
-0.004480	-0.002980	-0.003090	0.002951	0.005763	0.008092	0.003370	0.003566	0.003054	0.001252	-0.002520	-0.001920	-0.002720	-0.004480
-0.003660	-0.002390	-0.002230	0.003116	0.003740	0.003370	0.003876	0.003746	0.002934	0.002063	-0.001520	-0.001420	-0.001880	-0.003660
-0.003590	-0.002310	-0.002300	0.004207	0.003341	0.003566	0.003746	0.005404	0.002869	0.001477	-0.002250	-0.001890	-0.002540	-0.003590
-0.006950	-0.004670	-0.004750	0.002066	0.002486	0.003054	0.002934	0.002869	0.006738	0.003054	0.001212	0.001331	0.000955	-0.006950
-0.010180	-0.007860	-0.007150	0.000366	0.001192	0.001293	0.002063	0.001477	0.003054	0.008576	0.006171	0.008179	0.007260	-0.010180
-0.020040	-0.015830	-0.016000	-0.003650	-0.002790	-0.002520	-0.001520	-0.002250	0.001212	0.006171	0.036153	0.018390	0.020017	-0.020040
-0.018440	-0.015290	-0.013460	-0.003810	-0.002880	-0.001920	-0.001420	-0.001890	0.001331	0.008179	0.018390	0.033117	0.029414	-0.018440
B_{01} coefficients (vector)													
-0.538520	0.042741	0.030108	0.019242	0.021506	-0.002880	-0.004000	-0.004470	-0.002720	-0.003230	-0.006940	-0.007450	-0.019520	-0.012170

Table 3.12: **Generators real power optimal values of the IEEE 118 bus system**

Power [MW] / Price Penalty factors	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	P7 [MW]
Max-Max	283.0817	279.1622	258.46	285.1584	294.2349	310.3541	226.725
Min-Max	215.0298	208.4636	310.5390	330.4441	322.1951	219.9083	262.9498
Power [MW] / Price Penalty factors	P8 [MW]	P9 [MW]	P10 [MW]	P11 [MW]	P12 [MW]	P13 [MW]	P14 [MW]
Max-Max	223.3824	285.8279	278.7118	280.551	302.8963	223.1066	279.14
Min-Max	262.7275	273.8372	266.5802	236.9018	204.8774	262.7091	341.3145

Table 3.13: **Comparison of Combined Economic Emission Dispatch problem EPMCEED and Lagrange's solutions for the IEEE 118 bus system**

Reference	Algorithm	Penalty factor	Fuel cost F_C [$/h]	Emission E_T [lbs/h]	CEED F_T [$/h]	Transmission loss P_L [MW]	Computation Time [s]
(Venkatesh, et al., 2003)	EPMCEED	Max-Max	18479.155	15815.079	141566.899	130.743	224.1
Proposed	Lagrange’s	Max-Max	17916.1113	14219.3073	25015.1159	142.7925	56.6
		Min-Max	18046.1791	13577.1113	18251.3046	50.447	46.23

3.2.5.5 Discussion of the results for the IEEE 118 bus system

Lagrange's method using Max-Max penalty factor gives less fuel cost, emission and CEED fuel cost values in comparison with the EPMCEED method (Venkatesh, et al., 2003) for power demand of 3668 [MW], Table 3.13. In addition to that, CEED problem is solved using Min-Max (proposed) penalty factor by Lagrange's method. Min-Max penalty factor using Lagrange's method gives less fuel cost, emission and CEED fuel cost values in comparison with the Max-Max penalty factor.

3.2.5.6 Test System 4:

The CEED problem for eleven generator system is solved using Max-Max penalty factor for the various power demands 1000 to 2500 [MW]. The transmission loss is not considered in this case and is assumed as small value. The data such as fuel cost coefficients, emission coefficients, and generator real power limits are given in Table 3.14 (Balamurugan and Subramanian, 2007). The optimum generator real powers of the eleven generator system is calculated on the bases of the Lagrange's method and is given in Table 3.15. The Lagrange's solution of the CEED Problem for the objective function, calculating time send the used number of iterations are given in Table 3.16. The Lagrange's solution of fuel cost and emission is less in comparison with various algorithms such as Lambda Iteration, Recursive, Simple Dynamic Programming (SDP), Differential Evolution (DE), and PSO algorithms and is given in Table 3.17 and 3.18 respectively. The software developed ***CEED_Casestudy4.m*** is given in ***Appendix A4***.

Table 3.14: Fuel cost and emission coefficients of eleven generator system

Fuel cost coefficients			Emission coefficients			Generator real power limits	
a_i [$/MW^{2}h]	b_i [$/MWh]	c_i [$/h]	d_i [$/kg^{2}h]	e_i [$/kgh]	f_i [$/h]	Pmin [MW]	Pmax [MW]
0.00762	1.92699	387.85	0.00419	-0.67767	33.93	20	250
0.00838	2.11969	441.62	0.00461	-0.69044	24.62	20	210
0.00523	2.19169	422.57	0.00419	-0.67767	33.93	20	250
0.00140	2.01983	552.50	0.00683	-0.54551	27.14	60	300
0.00154	2.22181	557.75	0.00751	-0.40060	24.15	20	210
0.00177	1.91528	562.18	0.00683	-0.54551	27.14	60	300
0.00195	2.10681	568.39	0.00751	-0.40006	24.15	20	215
0.00106	1.99138	682.93	0.00355	-0.51116	30.45	100	455
0.00117	1.99802	741.22	0.00417	-0.56228	25.59	100	455
0.00089	2.12352	617.83	0.00355	-0.41116	30.45	110	460
0.00098	2.10487	674.61	0.00417	-0.56228	25.59	110	465

Table 3.15: **The optimum generator active power of the eleven generator system**

P_D [MW]	1000	1250	1500	1750	2000	2250	2500
P1	85.606	94.617	103.628	112.639	121.650	130.661	139.672
P2	76.673	82.691	88.709	94.727	100.745	106.763	112.781
P3	87.258	97.016	106.773	116.530	126.288	136.045	145.802
P4	78.495	102.333	126.172	150.011	173.850	197.688	221.527
P5	47.916	62.725	77.535	92.345	107.154	121.964	136.774
P6	79.325	102.534	125.742	148.951	172.160	195.369	218.578
P7	49.765	64.848	79.931	95.013	110.096	125.178	140.261
P8	129.604	165.511	201.418	237.325	273.232	309.139	345.046
P9	122.370	156.889	191.408	225.927	260.446	294.965	329.484
P10	119.600	160.274	200.948	241.622	282.296	322.971	363.645
P11	123.389	160.562	197.736	234.909	272.083	309.256	346.430

Table 3.16: **The CEED problem solution using Lagrange's algorithm for the eleven generator system**

P_D [MW]	1000	1250	1500	1750	2000	2250	2500
F_C [$/h]	8502.2936	9108.3741	9733.5344	10377.7710	11041.0833	11723.4716	12424.9352
E_T [Kg/h]	205.2039	339.8696	540.5428	807.2229	1139.9096	1538.6032	2003.3032
F_T [$/h]	9235.2456	10266.5305	11533.1802	13035.1896	14772.5569	16745.2836	18953.3670
Iteration	39	40	43	45	46	47	47
CT [S]	0.0476	0.0308	0.0332	0.0338	0.0357	0.0350	0.0347

Table 3.17: **Comparison of Lagrange's and various algorithms fuel cost values for the eleven generator system**

P_D [MW]	Fuel cost F_C in [$/h]					
	(Balamurugan and Subramanian, 2007)					Proposed
	λ - iteration	Recursive	PSO	DE	SDP	Lagrange's
1000	8502.30	8502.29	8508.24	8505.81	**8502.29**	8502.2936
1250	9108.38	9108.38	9114.42	9117.63	9108.38	**9108.3741**
1500	9733.54	9733.54	9737.33	9736.22	9733.54	**9733.5344**
1750	10377.78	10377.77	10380.82	10377.77	10377.86	**10377.7710**
2000	11041.08	11041.08	11041.09	11041.08	**11041.08**	11041.0833

2250	11723.47	11723.47	11725.68	11723.47	**11723.47**	11723.4716
2500	12424.94	12424.94	12428.63	12425.06	12424.94	**12424.9352**

Table 3.18: Comparison of Lagrange's and various algorithms emission values for the eleven generator system

P_D [MW]	Total Emission E_T in [kg/h]					
	(Balamurugan and Subramanian, 2007)					Proposed
	λ - iteration	Recursive	PSO	DE	SDP	Lagrange's
1000	205.205	205.204	208.012	205.206	205.204	**205.2039**
1250	339.870	339.870	345.669	339.935	339.870	**339.8696**
1500	540.544	540.544	545.307	544.298	540.544	**540.5428**
1750	807.220	807.220	812.863	807.236	**807.220**	807.2229
2000	1139.911	1139.911	1142.182	1139.911	1139.911	**1139.9096**
2250	1538.600	1538.600	1540.465	1538.659	**1538.600**	1538.6032
2500	2003.300	2003.300	2009.720	2003.350	**2003.300**	2003.3032

3.2.5.7 Discussion of the results for the elven generator system

Lagrange's solution for CEED problem using Max-Max price penalty factor method is given in Table 3.16. Lagrange's solution is compared with various algorithms (Balamurugan and subramanian,2007), Table 3.18. The fuel cost, emission values are less in Lagrange's method in comparison with λ-iteration, Recursive, PSO, DE, and SDP algorithms ((Balamurugan and subramanian,2007), Table 3.17 and 3.18 respectively.

3.2.6 Discussion and conclusion

New type Min-Max, and Max-Min price penalty factors are proposed for solution of the bi-criteria power system dispatch optimization problem. Lagrange's algorithm is developed to solve the bi-criteria dispatch problem. The Lagrangian optimal solutions of the dispatch problem for the IEEE 118 bus, IEEE 30 bus and 11 generator system are used to test the developed algorithm. The solutions of the IEEE 118, 30 and 11 bus system are compared with, the solutions obtained (Venkatesh et al., 2003), (Gnanadass, 2005), and (Balamurugan and Subramanian, 2007) by using Evolutionary Programming (EP) and Dynamic Programming (DP), λ-iteration, Recursive, PSO, DE, and SDP algorithms respectively. The obtained results show that the Min-Max (proposed) price penalty factor provides better optimization solution for the CEED problem in comparison to the other types of price penalty factors. The optimization results also show that the order of price penalty factors depends on the Input-Output (IO) unit curves used. It can be concluded that:

i. The optimal overall cost for the bi-criteria dispatch problem and the fuel cost of the economic dispatch problem are less when using the Min-Max Price penalty factor in comparison with other price penalty factors.

ii. The Max-Max price penalty factor is good to yield a minimum emission values in comparison with other price penalty factors.

iii. The Max-Min price penalty factor is good to yield a minimum power loss values in comparison with other price penalty factors.

3.3.1 Bi-criteria dispatch problem with valve point loading effect in the fuel cost and emission criterion functions

3.3.2 Economic dispatch problem formulation with valve point loading effect

Large steam turbine generators usually have a number of steam admission valves that are opened in sequence to do gradually loading of the generators. Loading output levels at which a new steam admission valve is opened is called valve point loading effect. At these levels discontinuities in the cost curves and in the incremental rate curves occur as a result of the sharp increases in throttle losses. As the unit loading increases , the input to the unit increases and the incremental cost heat rate decreases between the opening points for any two valves. This gives rise to the discontinuous type of incremental heat rate characteristic. It increases the non-linearity of the search space as well as number of local minima of the CEED problem solution. The output power of the fossil plant is increased sequentially by opening set of valves at the inlet of the turbine as shown in Figure 3.10. As a result the operating cost is usually approximated by one or more quadratic segments as shown in Figure 3.12. (Gernald and Aaron, 1958)

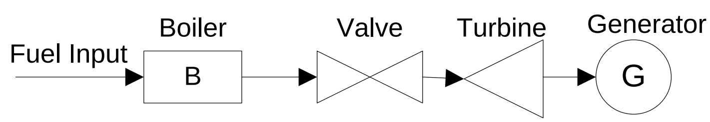


Figure 3.10: **Fossil power plant**

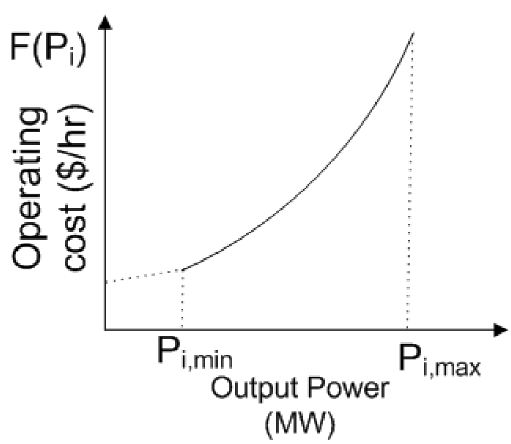


Figure 3.11: **Operating characteristics curve for the fossil power plant**

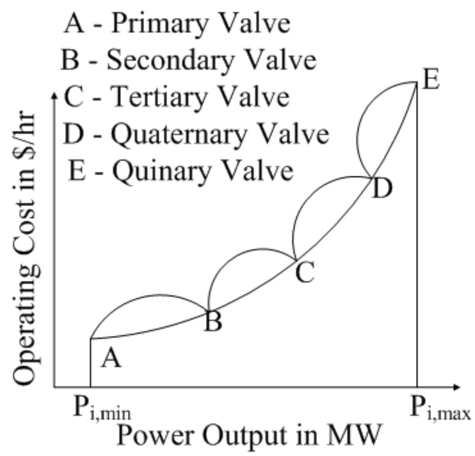


Figure 3.12: **Valve point loading effect for the fossil power plant**

Most of the literature papers concentrate on combined economic emission dispatch problem without considering valve point effect loading. The input-output characteristics or cost functions of a generator are approximated using quadratic or piecewise quadratic functions, under the assumption that the incremental cost curves of the units are monotonically increasing piece-wise linear functions. However, real input-output characteristics of generators have higher order non-linarites and discontinuities in the fuel cost curve due to the valve point effect loading of the fossil fuel burning plants. (Kothari and Dhillon, 2011). The valve point loading effect has been modeled as a recurring rectified sinusoidal function given in Equation (3.28), shown in Figure 3.3. The authors (Gernald and Aaron, 1958) describe that the turbine operating between the curves of the valve points operates less efficiency because of the throttling of steam passing through the throttled valve. The control valve arrangement maintains a higher average efficiency over a wide range of loads by successively admitting steam to groups of nozzles to meet the increasing loads. It is described in the literature that more valves a turbine has, the less the spread in efficiency between the valve points. The valves are operated in sequence by a camshaft or other device under the control of the governor. Actually the power plant has multiple steam valves, so it is necessary to consider the valve point loading effect. Therefore, solution of the dispatch problem is also more accurate when the dispatch problem with valve point loading effect is solved.

A. Fuel cost function with valve point loading effect

Thermal power plants have multiple steam valves. To accurately evaluate the fuel cost function, the valve point effect loading is considered as follows: (Kothari and Dhillon,2011)

$$F_{CVP} = \sum_{i=1}^{n} F_i\left(P_i\right) = \sum_{i=1}^{n}\left[(a_i + b_i P_i + c_i P_i{}^2) + \left|\alpha_i \sin(\beta_i (P_{i,\min} - P_i))\right|\right] \text{ [\$/h]} \qquad (3.26)$$

Where

F_{CVP} Total Fuel Cost with valve point effect

$a_{i,} b_i, c_i$ Fuel cost coefficients of the generating unit i in [$/h], [$/MWh] and [$/MW2h] respectively.

$\alpha_{i,} \beta_i$ Valve point effect cost coefficients of the generating unit *i*

B. Emission function with valve point loading effect

The emission of the thermal power plant with inserting the valve point loading effect is given as follows:

$$E_{TVP} = \sum_{i=1}^{n} \left(d_i + e_i P_i + f_i P_i^2 \right) + \gamma_i \exp(\delta_i P_i) \qquad \text{[kg/h]} \qquad (3.27)$$

Where

E_{TVP}	Total emission with valve point loading effect
d_i, e_i, f_i	Emission coefficients of the generating unit i in [ton/h], [ton/MWh] and [ton/MW^{2}h] respectively.
γ_i, δ_i	Valve point loading effect emission coefficients of the generating unit i

C. Bi-criteria objective function with valve point loading effect using penalty factors

The combined economic emission dispatch problem is formulated as follows:

$$F_{TVP} = \sum_{i=1}^{n} \left[\begin{array}{l} \left[(a_i + b_i P_i + c_i P_i^{\,2}) + \left| \alpha_i \sin(\beta_i (P_{i,\min} - P_i)) \right| \right] + \\ + h_i \left[\left(d_i + e_i P_i + f_i P_i^2 \right) + \gamma_i \exp(\delta_i P_i) \right] \end{array} \right] \qquad \text{[\$/h]} \qquad (3.28)$$

Where

h_i	Price penalty factor
F_{TVP}	CEED fuel cost value with valve point loading effect

D. Formulation of price penalty factors of bi-criteria dispatch problem with valve point loading effect

The economic dispatch and emission dispatch are two different problems. Emission dispatch can be included in economic dispatch problem by the addition of emission cost to the normal dispatch. The bi-objective Combined Economic Emission Dispatch (CEED) problem can be converted into single objective optimization problem by introducing a price penalty factor *h*. The fuel cost and emission functions with valve point loading effect are correlated with any one of the considered below price penalty factors. The thesis uses six types of penalty factors for formulation of the bi-criteria dispatch problem. Two of which are proposed for the first time in the thesis in Chapter 3, point 3.2.3

i. The ratio of the minimum fuel cost and the maximum emission with valve point loading effects for the i^{th} generator is called Min-Max Penalty factor. It is proposed in the thesis and can be formulated as

$$h_i = \frac{(a_i + b_i P_{i,\min} + c_i P_{i,\min}^2) + \left|\alpha_i \sin(\beta_i (P_{i,\min} - P_{i,\min}))\right|}{\left(d_i + e_i P_{i,\max} + f_i P_{i,\max}^2\right) + \gamma_i \exp(\delta_i P_{i,\max})}, \; i = \overline{1,n} \qquad [\$/kg] \qquad (3.29)$$

ii. The ratio of the maximum fuel cost and the maximum emission with valve point loading effects for the i^{th} generator is called Max- Max Penalty factor. It can be formulated as (Balamurugan and Subramanian,2007)

$$h_i = \frac{(a_i + b_i P_{i,\max} + c_i P_{i,\max}^2) + \left|\alpha_i \sin(\beta_i (P_{i,\min} - P_{i,\max}))\right|}{\left(d_i + e_i P_{i,\max} + f_i P_{i,\max}^2\right) + \gamma_i \exp(\delta_i P_{i,\max})}, \; i = \overline{1,n} \qquad [\$/kg] \qquad (3.30)$$

iii. The ratio of the minimum fuel cost and the minimum emission with valve point loading effects for the i^{th} generator is called Min- Min Penalty factor. It is proposed in the thesis and can be formulated as

$$h_i = \frac{(a_i + b_i P_{i,\min} + c_i P_{i,\min}^2) + \left|\alpha_i \sin(\beta_i (P_{i,\min} - P_{i,\min}))\right|}{\left(d_i + e_i P_{i,\min} + f_i P_{i,\min}^2\right) + \gamma_i \exp(\delta_i P_{i,\min})}, \; i = \overline{1,n} \qquad [\$/kg] \qquad (3.31)$$

iv. The ratio of the maximum fuel cost and the minimum emission with valve point loading effects for the i^{th} generator is called Max- Min Penalty factor. It can be formulated as

$$h_i = \frac{(a_i + b_i P_{i,\max} + c_i P_{i,\max}^2) + \left|\alpha_i \sin(\beta_i (P_{i,\min} - P_{i,\max}))\right|}{\left(d_i + e_i P_{i,\min} + f_i P_{i,\min}^2\right) + \gamma_i \exp(\delta_i P_{i,\min})}, \; i = \overline{1,n} \qquad [\$/kg] \qquad (3.32)$$

v. The average of the Min-Max, Max-Max, Min-Min, and Max-Min with valve point loading effects for the i^{th} generator is called the average penalty factor. It can be formulated as

$$h_i = \frac{\left[\begin{array}{l} \dfrac{(a_i + b_i P_{i,\min} + c_i P_{i,\min}^2) + \left|\alpha_i \sin(\beta_i (P_{i,\min} - P_{i,\min}))\right|}{\left(d_i + e_i P_{i,\max} + f_i P_{i,\max}^2\right) + \gamma_i \exp(\delta_i P_{i,\max})} + \\ + \dfrac{(a_i + b_i P_{i,\max} + c_i P_{i,\max}^2) + \left|\alpha_i \sin(\beta_i (P_{i,\min} - P_{i,\max}))\right|}{\left(d_i + e_i P_{i,\max} + f_i P_{i,\max}^2\right) + \gamma_i \exp(\delta_i P_{i,\max})} + \\ + \dfrac{(a_i + b_i P_{i,\min} + c_i P_{i,\min}^2) + \left|\alpha_i \sin(\beta_i (P_{i,\min} - P_{i,\min}))\right|}{\left(d_i + e_i P_{i,\min} + f_i P_{i,\min}^2\right) + \gamma_i \exp(\delta_i P_{i,\min})} + \\ + \dfrac{(a_i + b_i P_{i,\max} + c_i P_{i,\max}^2) + \left|\alpha_i \sin(\beta_i (P_{i,\min} - P_{i,\max}))\right|}{\left(d_i + e_i P_{i,\min} + f_i P_{i,\min}^2\right) + \gamma_i \exp(\delta_i P_{i,\min})} \end{array}\right]}{4}, \; i = \overline{1,n} \qquad [\$/kg] \qquad (3.33)$$

vi. Apart from the penalty factors discussed above, a common penalty factor exist for the bi-criteria dispatch problem. It can be formulated as a modification to the one presented (Balamurugan and Subramanian, 2011)

$$h_i = \sum_{i=1}^{n} \frac{\left[\begin{array}{l} \frac{(a_i + b_i P_{i,\min} + c_i P_{i}^{2}{}_{,\min}) + \left|\alpha_i \sin(\beta_i (P_{i,\min} - P_{i,\min}))\right|}{\left(d_i + e_i P_{i,\max} + f_i P_{i,\max}^2\right) + \gamma_i \exp(\delta_i P_{i,\max})} + \\ + \frac{(a_i + b_i P_{i,\max} + c_i P_{i}^{2}{}_{,\max}) + \left|\alpha_i \sin(\beta_i (P_{i,\min} - P_{i,\max}))\right|}{\left(d_i + e_i P_{i,\max} + f_i P_{i,\max}^2\right) + \gamma_i \exp(\delta_i P_{i,\max})} + \\ + \frac{(a_i + b_i P_{i,\min} + c_i P_{i}^{2}{}_{,\min}) + \left|\alpha_i \sin(\beta_i (P_{i,\min} - P_{i,\min}))\right|}{\left(d_i + e_i P_{i,\min} + f_i P_{i,\min}^2\right) + \gamma_i \exp(\delta_i P_{i,\min})} + \\ + \frac{(a_i + b_i P_{i,\max} + c_i P_{i}^{2}{}_{,\max}) + \left|\alpha_i \sin(\beta_i (P_{i,\min} - P_{i,\max}))\right|}{\left(d_i + e_i P_{i,\min} + f_i P_{i,\min}^2\right) + \gamma_i \exp(\delta_i P_{i,\min})} \end{array} \right]}{4n}, \; i = \overline{1,n} \quad [\$/kg] \qquad (3.34)$$

As was mentioned before the role of all penalty factors is to transfer the physical meaning of emission criterion from weight of the emission to the fuel cost for the emission. The difference between these penalty factors is in the weight of the fuel cost for emission in the final optimal fuel cost for generation and emission, criterion 3.6.

Comparison between the influences of the separate price factors over the optimization problem solution is done through application of the Lagrange's method for calculation of this solution.

E. Constraints of the dispatch problem

The constraints such as (i) Power balance and (ii) Generator operational limits are given in Equation (3.2), (3.3) and (3.4) respectively. The combined economic dispatch problem with criterion functions (3.28) are solved for every type of the price penalty factors (3.29) to (3.34) under the above constraints.

3.3.3 Development of the Lagrange's algorithm for the bi-criteria dispatch problem with valve point loading effect

The objective function in this case, is described by equation (3.28). The optimization problem is to find the optimal active power $P_{i,}\ i = \overline{1,n}$ produced by the generators in such a way that the criterion (3.28) is minimized and the constraints (3.2), (3.3) and (3.4) are satisfied. The problem has to be solved for every one of the price penalty factors (3.29) to (3.34).

The problem is solved using Lagrange's method by introduction of the Lagrange's variable λ and formulation of a Lagrange's function:

$$L = F_{TVP} + \lambda \left(P_D + P_L - \sum_{i=1}^{n} P_i \right) \qquad (3.37)$$

$$L=\sum_{i=1}^{n}\left[\begin{bmatrix}\left[(a_i+b_iP_i+c_iP_i^{\ 2})+\left|\alpha_i\sin(\beta_i(P_{i,\min}-P_i))\right|\right]+\\+h_i\left[\left(d_i+e_i\,P_i+f_iP_i^2\right)+\gamma_i\exp(\delta_iP_i)\right]\end{bmatrix}+\atop+\lambda\left(P_D+\sum_{i=1}^{n}\sum_{j=1}^{n}P_i\,B_{i\,j}\,P_j+\sum_{i=1}^{n}B_{0i}\,P_i+B_{00}\ -\sum_{i=1}^{n}P_i\right)\right] \tag{3.38}$$

The solution for the bi-criteria dispatch problem from equation (3.28), subject to the constraints (3.2), (3.3) and (3.4) with penalty factors (3.29) - (3.34) is transferred to the problem for minimization of L according to $P_{i,}\ i=\overline{1,n},$ under the constraints (3.4).

Necessary conditions for optimality for solution of the problem (3.38), are:

According to $P_i,\ \dfrac{\partial L}{\partial P_i}=0, i=\overline{1,n}$ (3.39)

According to $\lambda,\ \dfrac{\partial L}{\partial \lambda}=0$ (3.40)

Condition (3.39) is:

$$\frac{\partial L}{\partial P_i}=\begin{bmatrix}\left(2c_iP_i+b_i\right)-\left(\alpha_i\cos\left(\beta_i\left(P_{i,\min}-P_i\right)\right)\right)+2h_if_iP_i+h_ie_i+\\+h_i\gamma_i\delta_i\left(\exp\left(\delta_i\right)\right)+\lambda\left(2P_iB_{ij}+B_{0i}\right)-\lambda\end{bmatrix}=0\,,\ i=\overline{1,n} \tag{3.41}$$

Equation (3.41) is a nonlinear one and it cannot be presented in a linear vector-matrix form. It is a sum of first order nonlinear differential equations in which the derivative has to be equal to zero. This fact allows to transfer these equations to algebraic ones. Their solution can be done with some of the methods for solution of a set on nonlinear equations. The Matlab function *fsolve* solves systems of nonlinear equations of several variables using three different algorithms such as trust region dogleg, trust region reflective, and Levenberg-Marquardt. The MATLAB function *fsolve* is used to solve the equation (3.41) and hence the real power $P_{i,}\ i=\overline{1,n}$ of the CEED problem with valve point effect is found. This solution depends on the Lagrange's variable ***λ.***

The value of ***λ*** is unknown and has to be found from the necessary condition for optimality (3.40)

This condition is found from (3.38) to be:

$$\frac{\partial L}{\partial \lambda}=\left(P_D+\sum_{i=1}^{n}\sum_{j=1}^{n}P_i\,B_{i\,j}\,P_j+\sum_{i=1}^{n}B_{0i}\,P_i+B_{00}\ -\sum_{i=1}^{n}P_i\right)=0=\Delta\lambda \tag{3.42}$$

Equation (3.42) is not a function of λ, but represents the gradient of L according to λ. At the optimal solution this gradient has to be equal to zero. As analytical solution is not possible and a gradient procedure for calculation of λ has to be developed as follows

$$\lambda^{(k+1)} = \lambda^{k} + \alpha\,\Delta\lambda^{(k)}, \lambda \neq 0 \qquad (3.43)$$

Where

k is the iteration number and , $k = \overline{0:m}$,

m is the given maximum number of iterations.

$\Delta\lambda^{(k)}$ is determined by the equation (3.42) and α is the step of the gradient procedure. The calculations start from some given initial value of the Lagrange's variable $\lambda^{(0)}$. When during the iterations $\Delta\lambda = 0$, the optimal solution for the Lagrange's variable is obtained. It will determine the optimal solution for the power that has to be produced by the generators.

The obtained solutions for $P_i, i = \overline{1,n}$ have to belong to the constraint domain (3.4). That is why for every step of the gradient procedure, the obtained solution is fit to the constraint domain following the procedure

$$P_i^{(k)} = \begin{pmatrix} P_{i,\min,} & if \;\; P_i^{(k)} \langle P_{i,\min} \\ P_i^{(k)}, & if \;\; P_{i.\min} \leq P_i^{(k)} \langle P_{i,\max} \\ P_{i,\max,} & if \;\; P_i^{(k)} \rangle P_{i,\max} \end{pmatrix} \qquad (3.44)$$

The condition for end of the iterations is

$$\Delta\lambda^{(k)} \leq \varepsilon \qquad (3.45)$$

Where $\varepsilon \rangle 0$ is a small positive number.

The algorithm of the method is:

1) Initial value of the Lagrange's multiplier is guessed: $\lambda^{(0)}$, and the value of the condition for optimality ε is given.
2) Initial value of the active power $P_i^0, i = \overline{1,n}$ is guessed for the case of the dispatch problem with the valve point effect
3) Equation (3.41) is solved by Matlab function ***fsolve*** and $P^{(1)}$ is determined.
4) The obtained vector P_i^1 is fit to the constraints (3.44).
5) $\Delta\lambda^{(0)}$ is calculated using equation (3.42) where $P^{(1)}$ is substituted.

6) The condition (3.45) is checked. If it is fulfilled the calculations stop, if not, improved value of $\lambda \to \lambda^{(1)}$ is calculated using equation (3.43)
7) Calculations of the improved values of $P_i \to P_i^{(2)}$ is done as using equation (3.41). Iterations continue until condition (3.45) is satisfied or until the maximum number of iterations ***m*** is reached.
8) The optimal solution is used to calculate the fuel cost, emission, and CEED fuel cost using equations (3.26), (3.27) and (3.28).

Flowchart of the algorithm is given in Figure 3.13

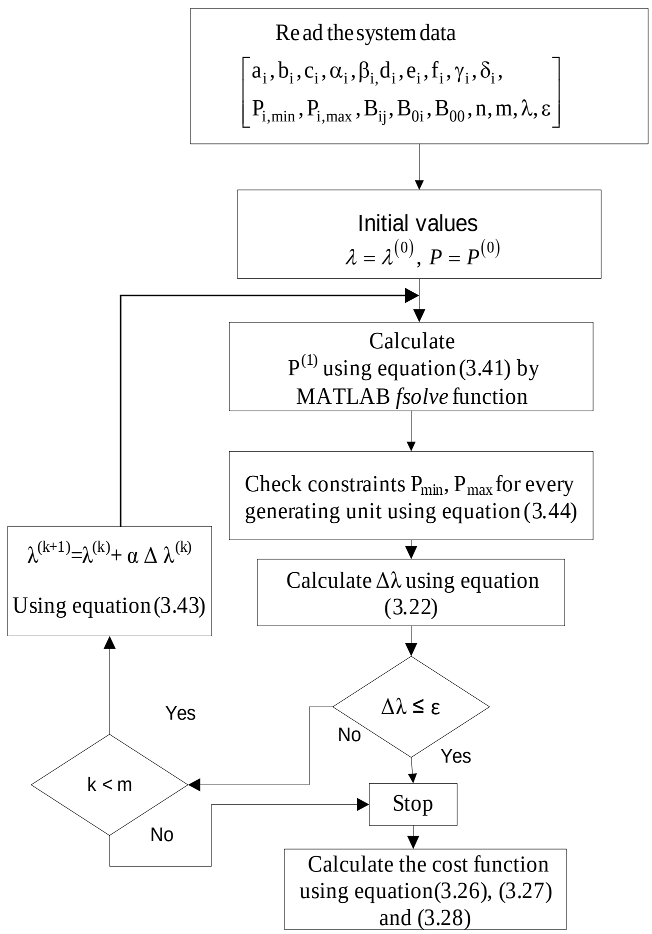


Figure 3.13: **Flow diagram of the solution of the bi-Criteria CEED problem with valve point loading effect using Lagrange's method**

The software developed *CEED_VP_Casestudy1.m* and *CEED_VP_Casestudy1_funct.m* for solution of the CEED problem with valve point loading effect is given in *Appendix B*. The simulation is done for two different case studies and the developed software for the two case studies are given in Appendix B1 to B4.

3.3.4 Lagrange's algorithm for economic dispatch problem with valve point loading effect neglecting emission and transmission line losses

This section formulate and proposes solution of the economic dispatch problem with valve point loading effect by neglecting emission function and transmission losses. The economic dispatch problem is formulated as follows:

Find the values of the active power produced by the generators $P_{i,}\, i=\overline{1,n}$ such that the criterion

$$F_{TVP}=\sum_{i=1}^{n}(a_i+b_iP_i+c_iP_i^{\,2})+\left|\alpha_i\sin(\beta_i(P_{i,\min}-P_i))\right| \qquad [\$/h] \qquad (3.46)$$

Where

F_{TVP} CEED fuel cost value with valve point loading effect

Under the constraints:

- Power balance constraint

$$\sum_{i=1}^{n}P_i=P_G=P_D \qquad [MW] \qquad (3.47)$$

Where

P_G Total power generation of the system

P_D Total demand of the system

- Generator operational constraints described by the equation (3.4)

The optimization problem is to find the optimal active power $P_{i,}\; i=\overline{1,n}$ produced by the generators in such a way that the criterion (3.46) is minimized and the constraints (3.4) and (3.47) are satisfied. The problem is solved using Lagrange's method by introduction of the Lagrange's variable *λ* and formulation of a Lagrange's function:

$$L=F_{TVP}+\lambda\left(P_D-\sum_{i=1}^{n}P_i\right) \qquad (3.48)$$

$$L=\sum_{i=1}^{n}(a_i+b_iP_i+c_iP_i^{\,2})+\left|\alpha_i\sin(\beta_i(P_{i,\min}-P_i))\right|+\lambda\left(P_D-\sum_{i=1}^{n}P_i\right) \qquad (3.49)$$

The solution for the bi-criteria dispatch problem is based on optimisation of the Equation (3.49), subject to the constraints (3.4) and (3.47) is transferred to the problem for minimization of L according to $P_{i,}\; i=\overline{1,n}$ and maximization of L according to λ

The necessary conditions for optimality for solution of the problem (3.39), (3.40) are: Which results in the following equations:

$$\frac{\partial L}{\partial P_i} = \left[\left(2c_i P_i + b_i\right) - \left(\alpha_i \cos\left(\beta_i \left(P_{i,\min} - P_i\right)\right)\right) - \lambda \right] \tag{3.50}$$

$$\frac{\partial L}{\partial \lambda} = \left(P_D - \sum_{i=1}^{n} P_i \right) = 0 = \Delta\lambda \tag{3.51}$$

Equations (3.50), (3.51) are solved under the constraints (3.4) and following algorithm is given in Figure 3.13.

3.3.5.1 Results from the solution of the bi-criteria dispatch problem with valve point loading effect for IEEE 30 bus system

The IEEE 30 bus system with six generators is represented by a fuel cost, emission and transmission loss coefficients with valve point loading effect and real power limits as given in Table 3.19. The problem is formulated as a bi-objective function using the various penalty factors from point 3.3.2. The dispatch problem is solved in Matlab environment. The Lagrange's method is used to obtain the solution of the dispatch problem. The solution of the multi-criteria dispatch problem with valve point loading effect is shown in Table 3.20 for the considered various penalty factors. The software developed *CEED_VP_Casestudy1.m* and *CEED_VP_Casestudy1_funct.m* is given in *Appendix B1* and *B2.*

Table 3.19: **Data of the IEEE 30 bus system for the bi-criteria dispatch Problem with valve point loading effect**

Generator Number		1	2	3	4	5	6
Fuel cost coefficients and valve point loading effect coefficients	a_i [\$/h]	10	10	20	10	20	10
	b_i [\$/MWh]	200	150	180	100	180	150
	c_i [\$/MW^{2}h]	100	120	40	60	40	100
	α_i	15	10	10	5	5	5
	β_i	6.283	8.976	14.784	20.944	25.133	18.48
Emission coefficients and valve point loading effect coefficients	d_i [\$/h]	4.091	2.543	4.258	5.326	4.258	6.131
	e_i [\$/t.h]	-5.554	-6.047	-5.094	-3.55	4.586	5.151
	f_i [\$/t^{2}h]	6.49	5.638	4.586	3.38	4.586	5.151
	γ_i	0.0002	0.0005	0.000001	0.002	0.000001	0.00001
	δ_i	2.857	3.333	8	2	8	6.667
Generator real power operational limits	$P_{i,min}$ (p.u)	0.05	0.05	0.05	0.05	0.05	0.05
	$P_{i,max}$ (p.u)	0.50	0.60	1.00	1.20	1.00	0.60

Transmission loss coefficients B_{00}=0.0014						
B coefficients						B_{01}
0.0218	0.0107	-0.00036	-0.0011	0.00055	0.0033	0.000010731
0.0107	0.01704	-0.0001	-0.00179	0.00026	0.0028	0.0017704
-0.0004	-0.0002	0.02459	-0.01328	-0.0118	-0.0079	-0.0040645

-0.0011	-0.00179	-0.01328	0.0265	0.0098	0.0045	0.0038453
0.00055	0.00026	-0.0118	0.0098	0.0262	-0.0001	0.0013832
0.0033	0.0028	-0.00792	0.0045	-0.0001	0.0297	0.0055503

Table 3.20: **Bi-criteria dispatch Problem with valve point loading effect solution using various price penalty factors**

Power demand (P_D) = 2.86 (p.u) Maximum number of iteration (m) = 2000 Initial lambda $\lambda^{(0)}$ = 20							(Hemam--alini and Simon, 2008)	(Abido, 2003)	(Abido, 2003)
Algorithm	Lagrange's (Proposed)						PSO	Evolution-ary	SPEA
Type of price penalty factor	Min - Max	Max - Max	Min - Min	Max - Min	Average	Common	Max-Max	--------	--------
Optimal (λ)	214.879	197.0550	216.9261	206.7494	208.7931	217.5211	--------	--------	--------
P1 (p.u)	0.2096	0.3164	0.2024	0.3083	0.2801	0.1928	0.14089	0.2699	0.2996
P2 (p.u)	0.3699	0.4892	0.3217	0.4518	0.4502	0.3390	0.34415	0.3885	0.4474
P3 (p.u)	0.5414	0.5269	0.5488	0.5384	0.5356	0.5508	0.67558	0.5645	0.7327
P4 (p.u)	0.9220	0.6211	0.9325	0.6368	0.6997	0.9206	0.83971	0.6570	0.7284
P5 (p.u)	0.4966	0.5174	0.5316	0.5321	0.5237	0.5355	0.49043	0.5441	0.1197
P6 (p.u)	0.3508	0.4138	0.3535	0.4173	0.3961	0.3515	0.39797	0.4398	0.5364
P_L (p.u)	0.0397	0.0348	0.0404	0.0346	0.0353	0.04	--------	--------	--------
F_C [$/h]	649.347	667.2978	648.7502	660.5127	658.2729	647.933	639.65	618.689	629.394
E_T [t/h]	0.1975	0.1876	0.1988	0.1880	0.1895	0.1984	0.21105	0.19940	0.21043
F_T [$/h]	650.537	677.0402	649.7328	668.7905	663.3373	648.8344	--------	--------	--------
Iter.No when $\lambda^{(0)}$ =20	185	538	193	517	402	187	--------	--------	--------
Iter.No when $\lambda^{(0)}$ =195	95	116	108	258	225	203	--------	--------	--------
CT [s] when $\lambda^{(0)}$ =20	4.3994	10.2016	4.4761	9.6737	34.0414	31.4099	--------	--------	--------
CT [s] when $\lambda^{(0)}$ =195	3.2318	3.3570	3.9164	6.0567	16.1770	13.8698	--------	--------	--------

Where

SPEA Strength Pareto Evolutionary Algorithm

The notation "--------" means that data is not available in the considered literature papers.

3.3.5.2 Discussion on results for IEEE 30 bus system

The results calculated according to the Lagrange's method for the CEED problem solution are compared with the results from the reference papers (Hemamalini and Simon, 2008),(Abido, 2003), and (Abido, 2003) using various methods of solution, Table 3.20. The following conclusions are made:

i) The fuel cost values and CEED fuel cost values are less when using common penalty factor in comparison with the other types of penalty factors.

ii) The emission function value is less using Max-Max penalty factor in comparison with the other types of penalty factors.

iii) The computation time and number of iterations required to obtain the global solution depend on the initial selection of the Lagrangian variable λ^0 and is independent on the type of the penalty factors as shown in Table 3.20

iv) The fuel cost values are bigger in Lagrange's method in comparison with (Hemamalini and Simon, 2008), (Abido, 2003), and (Abido, 2003) as shown in Table 3.20. It is because the authors (Hemamalini and Simon, 2008), (Abido, 2003), and (Abido, 2003) neglected the transmission loss constraints, and optimised the fuel cost and emission functions separately.

3.3.5.3 Results of the solution of the economic dispatch problem with valve point loading effect for 40 generator system

The 40 generator system data such as fuel cost and emission coefficients with valve point loading effect and real power limits are given in Table 3.21. The transmission loss is neglected in this case. The software developed *CEED_VP_Casestudy2.m* and *CEED_VP_Casestudy2_funct.m* is given in *Appendix B3 and B4*.

Table 3.21: 40 generator system data and results from the solution of the economic dispatch Problem with valve point loading effect

Generator Number	Fuel cost coefficients and valve point loading effect coefficients					Generator real power operational limits		Optimized real power values using Lagrange's algorithm
	a_i [$/MW^{2}h]	b_i [$/MWh]	c_i [$/h]	α_i	β_i	P_i,min [MW]	P_i,max [MW]	P_i [MW]
1	0.0069	6.73	94.705	100	0.084	36	114	73.4195
2	0.0069	6.73	94.705	100	0.084	36	114	73.4195
3	0.02028	7.07	309.54	100	0.084	60	120	97.4574
4	0.00942	8.18	369.03	150	0.063	80	190	129.8982
5	0.0114	5.35	148.89	120	0.077	47	97	87.8319
6	0.01142	8.05	222.33	100	0.084	68	140	105.4323
7	0.00357	8.03	287.71	200	0.042	110	300	296.4329
8	0.00492	6.99	391.98	200	0.042	135	300	300.0000
9	0.00573	6.6	455.76	200	0.042	135	300	209.8323
10	0.00605	12.9	722.82	200	0.042	130	300	204.8341
11	0.00515	12.9	635.2	200	0.042	94	375	168.8290

12	0.00569	12.8	654.69	200	0.042	94	375	168.8321
13	0.00421	12.5	913.4	300	0.035	125	500	394.3023
14	0.00752	8.84	1760.4	300	0.035	125	500	394.3203
15	0.00708	9.15	1728.3	300	0.035	125	500	214.7983
16	0.00708	9.15	1728.3	300	0.035	125	500	500.0000
17	0.00313	7.97	647.85	300	0.035	220	500	489.2964
18	0.00313	7.95	649.69	300	0.035	220	500	489.2964
19	0.00313	7.97	647.83	300	0.035	242	550	511.2964
20	0.00313	7.97	647.81	300	0.035	242	550	511.2964
21	0.00298	6.63	785.96	300	0.035	254	550	523.2956
22	0.00298	6.63	785.96	300	0.035	254	550	523.2956
23	0.00284	6.66	794.53	300	0.035	254	550	523.2948
24	0.00284	6.66	794.53	300	0.035	254	550	523.2948
25	0.00277	7.1	801.32	300	0.035	254	550	523.2944
26	0.00277	7.1	801.32	300	0.035	254	550	343.7749
27	0.52124	3.33	1055.1	120	0.077	10	150	52.2683
28	0.52124	3.33	1055.1	120	0.077	10	150	52.2683
29	0.52124	3.33	1055.1	120	0.077	10	150	52.2683
30	0.0114	5.35	148.89	120	0.077	47	97	87.8319
31	0.0016	6.43	222.92	150	0.063	60	190	184.4935
32	0.0016	6.43	222.92	150	0.063	60	190	184.4935
33	0.0016	6.43	222.92	150	0.063	60	190	184.4935
34	0.0001	8.95	107.87	200	0.042	90	200	164.8004
35	0.0001	8.62	116.58	200	0.042	90	200	164.8004
36	0.0001	8.62	116.58	200	0.042	90	200	164.8004
37	0.00161	5.88	307.45	80	0.098	25	110	105.0367
38	0.00161	5.88	307.45	80	0.098	25	110	105.0367
39	0.00161	5.88	307.45	80	0.098	25	110	105.0367
40	0.00313	7.97	647.83	300	0.035	242	550	511.2964

Comparison of the results from the solution of the economic dispatch problem with valve point loading effect with the results published in some papers, is given in Table 3.22

Table 3.22: **Comparison of the economic dispatch problem solutions for the 40 generator system with valve point loading effect with some results from application of other methods published**

Reference	Algorithm	F_T [$/h]	CT [s]
Developed	Lagrange's	126586.34	4.27
Park, et-al., 2010	Improved Particle Swarm optimization (IPSO) with both Chaotic sequences and Crossover operation (CCPSO)	121445.32	19.3
Selvakumar and Thanushkodi, 2007	New PSO –Local Random search (NPSO-LRS)	121664.43	3.93
Coelho, and Mariani, 2006	Differential Evolutionary Computation with Sequential Quadratic Programming (DEC-SQP)	121741.97	------------------
Sinha, et-al., 2003	Improved Fast Evolutionary Programming (IFEP)	123382.00	1167.35

3.3.5.4 Discussion on results for 40 generator system

The Lagrange's method is used to obtain the solution of the dispatch problem. The selection of initial lambda is $\lambda^0 = 500,$ the maximum number of iterations is taken as 1000 and the global solution is attained at 28th iteration. The dispatch problem is solved in Matlab environment. The results from the considered economic dispatch problem solution are compared with some results obtained by other methods such as PSO (Park, et-al.,2010) and Evolutionary Programming (Nidul Sinha, et-al,2003) as shown in Table 3.22. The following conclusions are made:

i) The fuel cost value of the economic dispatch problem with valve point loading effect using Lagrange's algorithm is bigger in comparison with the PSO (Park, et-al.,2010) and Evolutionary Programming (Nidul Sinha, et-al,2003) solutions as shown in Table 3.22. It concludes that Lagrange's algorithm provide near global solution for the 40 generator large scale optimization dispatch problem.

ii) The computation time of the Lagrange's algorithm is small in comparison with the random search algorithms PSO and Evolutionary Programming as shown in Table 3.22. It is observed that IFEP takes more computation time since the old computers have lower characteristics referring to the operation speed.

3.4 Impact of price penalty factors on the solution of the combined economic emission dispatch problem using cubic criterion functions

3.4.1 Introduction

The CEED problem using cubic fuel cost and emission objective functions is formulated and solved. Representation of generator fuel cost curves by polynomials in real-time economic dispatch is standard practice in the industry because it has a great influence on the accuracy of the economic dispatch problem solution. The second order polynomial function of the economic dispatch problem is formulated in (Rani et al., 2006), (Balamurugan and Subramanian, 2007), (Hemamalini and Simon, 2009), (Chen and Wang, 2009), and (Ming et al., 2010). The rough approximation of the generator cost function makes the Economic Dispatch (ED) solution to deviate from the optimal one. ED solution can be improved by introducing higher order generator fuel cost functions. Cubic cost functions are more accurately models of the actual thermal generators fuel cost. These cost functions are considered in (Mishra et al., 2006), (Al-Sumait et al., 2008),(Chayakulkeree and Ongsakul, 2007), (Venkatesh and Lee, 2008), (Adhinarayanan and Sydulu, 2010), (Adhinarayanan and Sydulu, 2010), and (Chen and Wang, 2009). In addition to that , various thermal pollutants such as SO_2, NO_x and CO_2 are considered in the cubic function model of the CEED problem.

3.4.2 Cubic objective functions

The objective of CEED problem is to minimize four conflicting objective functions: fuel cost and NOx, SO_2 and CO_2 emissions, while satisfying equality and inequality constraints. The objective functions are:

The fuel cost function is described as

$$F_C = \sum_{i=1}^{n} F_i(P_i) = \sum_{i=1}^{n} (a_i P_i^3 + b_i P_i^2 + c_i P_i + d_i) \qquad [\$/h] \qquad (3.52)$$

Where

F_C	Total Fuel Cost
$F_i(P_i)$	Fuel cost of the i^{th} generator
P_i	Real power generation of the unit i
a_i , b_i, c_i , d_i	Cost coefficients of the generation of the unit i
n	Number of generating units

The problem for minimization of the quantity of the emissions consider the following objective functions:

The criterion for minimization of the SO_2 emission is formulated as

$$E_{T\,so2} = \sum_{i=1}^{n} \left(a_{so_2 i} P_i^3 + b_{so_2 i} P_i^2 + c_{so_2 i} P_i + d_{so_2 i} \right) \qquad \text{[kg/h]} \qquad (3.53)$$

Where

E_{Tso2} Total sulphur dioxide emission

$a_{so_2 i}, b_{so_2 i}, c_{so_2 i}, d_{so_2 i}$ Sulphur-dioxide Emission coefficients of the generating unit *i*

The criterion for minimization of the NOx emission is formulated as

$$E_{T\,NOx} = \sum_{i=1}^{n} \left(a_{NOxi} P_i^3 + b_{NOxi} P_i^2 + c_{NOxi} P_i + d_{NOxi} \right) \qquad \text{[kg/h]} \qquad (3.54)$$

Where

E_{TNOx} Total nitrogen oxide emission

$a_{NOxi}, b_{NOxi}, c_{NOxi}, d_{NOxi}$ Nitrogen oxide Emission coefficients of the generating unit *i*

The criterion for minimization of the CO_2 emission is formulated as

$$E_{T\,Co2} = \sum_{i=1}^{n} \left(a_{Co_2 i} P_i^3 + b_{cO_{2i}} P_i^2 + c_{co_2 i} P_i + d_{co_2 i} \right) \qquad \text{[kg/h]} \qquad (3.55)$$

Where

E_{TCO2} Total carbon dioxide emission

$a_{Co_2 i}, b_{CO_{2i}}, c_{Co_2 i}, d_{Co_2 i}$ Carbon-dioxide Emission coefficients of the generating unit *i*

The following problems are formulated and solved:

1) The criteria (3.53), (3.54), and (3.55) are used in combination with the fuel cost criterion separately in order to formulate and solve bi-criteria economic emission dispatch problem.

2) All three criteria (3.53), (3.54) and (3.55) are used together with the criterion for the fuel cost in one multi-criteria dispatch problem.

3.4.3 Formulation of the bi-objective and the multi-objective economic dispatch problems

A bi-objective optimization is converted into a single objective optimization problem by introducing price penalty factor h_i to the various pollutants objective functions. Single objective functions are formulated for every pollutant separately as follows:

$$F_{TSO_2} = \sum_{i=1}^{n} \left[\begin{array}{l} (a_i P_i^{\,3} + b_i P_i^2 + c_i P_i + d_i) + \\ + h_{SO_2 i} \left(a_{so_2 i} P_i^3 + b_{so_2 i} P_i^2 + c_{so_2 i} P_i + d_{so_2 i} \right) \end{array} \right] \qquad \text{[\$/h]} \qquad (3.56)$$

$$F_{TNOx} = \sum_{i=1}^{n} \begin{bmatrix} (a_i P_i^{\,3} + b_i P_i^2 + c_i P_i + d_i) + \\ + h_{NOxi} \left(a_{NOxi} P_i^3 + b_{NOxi} P_i^2 + c_{NOxi} P_i + d_{NOxi} \right) \end{bmatrix} \text{[\$/h]} \qquad (3.57)$$

$$F_{TCO_2} = \sum_{i=1}^{n} \begin{bmatrix} (a_i P_i^{\,3} + b_i P_i^2 + c_i P_i + d_i) + \\ + h_{CO_2 i} \left(a_{Co_2 i} P_i^3 + b_{cO_{2i}} P_i^2 + c_{co_2 i} P_i + d_{co_2 i} \right) \end{bmatrix} \text{[\$/h]} \qquad (3.58)$$

Where

F_{TSO2} CEED's fuel cost of SO_2 emission

F_{TNOx} CEED's fuel cost of NO_x emission

F_{TCO2} CEED's fuel cost of CO_2 emission

In the cases where the impact of all 3 pollutants is important, the multi-objective dispatch problem is solved for all pollutants at the same moment. Then the CEED fuel cost for all SO_2, NOx, and CO_2 emissions is given by the equation (3.59)

$$F_{T_3} = \sum_{i=1}^{n} \begin{bmatrix} (a_i P_i^{\,3} + b_i P_i^2 + c_i P_i + d_i) + \\ + h_{SO_2 i} \left(a_{so_2 i} P_i^3 + b_{so_2 i} P_i^2 + c_{so_2 i} P_i + d_{so_2 i} \right) + \\ + h_{NOxi} \left(a_{NOxi} P_i^3 + b_{NOxi} P_i^2 + c_{NOxi} P_i + d_{NOxi} \right) + \\ + h_{CO_2 i} \left(a_{Co_2 i} P_i^3 + b_{cO_{2i}} P_i^2 + c_{co_2 i} P_i + d_{co_2 i} \right) \end{bmatrix} \text{[\$/h]} \qquad (3.59)$$

The bi-objective dispatch problem is solved for all pollutants individually using the average price penalty factor, Equation (3.60) to (3.62). Then the CEED fuel cost for the SO_2, NOx, and CO_2 emissions using average penalty factor is given in the equations (3.60), (3.61) and (3.62).

$$F_{TAvgSO_2} = \sum_{i=1}^{n} \begin{bmatrix} (a_i P_i^{\,3} + b_i P_i^2 + c_i P_i + d_i) + \\ + h_{AvgSO_2 i} \left(a_{so_2 i} P_i^3 + b_{so_2 i} P_i^2 + c_{so_2 i} P_i + d_{so_2 i} \right) \end{bmatrix} \text{[\$/h]} \qquad (3.60)$$

$$F_{TAvgNOx} = \sum_{i=1}^{n} \begin{bmatrix} (a_i P_i^{\,3} + b_i P_i^2 + c_i P_i + d_i) + \\ + h_{AvgNOxi} \left(a_{NOxi} P_i^3 + b_{NOxi} P_i^2 + c_{NOxi} P_i + d_{NOxi} \right) \end{bmatrix} \text{[\$/h]} \qquad (3.61)$$

$$F_{TAvgCO_2} = \sum_{i=1}^{n} \begin{bmatrix} (a_i P_i^{\,3} + b_i P_i^2 + c_i P_i + d_i) + \\ + h_{AvgCO_2 i} \left(a_{Co_2 i} P_i^3 + b_{cO_{2i}} P_i^2 + c_{co_2 i} P_i + d_{co_2 i} \right) \end{bmatrix} \text{[\$/h]} \qquad (3.62)$$

The bi-objective dispatch problem is solved for all pollutants individually using the common price penalty factor, Equation (3.63) to (3.65). Then the CEED fuel cost for the SO_2, NOx, and CO_2 emissions using common penalty factor is given in the equation (3.63), (3.64) and (3.65)

$$F_{TcomSO_2} = \sum_{i=1}^{n} \begin{bmatrix} (a_i P_i{}^3 + b_i P_i^2 + c_i P_i + d_i) + \\ + h_{comSO_2 i} \left(a_{so_2 i} P_i^3 + b_{so_2 i} P_i^2 + c_{so_2 i} P_i + d_{so_2 i} \right) \end{bmatrix} \quad [\$/h] \qquad (3.63)$$

$$F_{TcomNOx} = \sum_{i=1}^{n} \begin{bmatrix} (a_i P_i{}^3 + b_i P_i^2 + c_i P_i + d_i) + \\ + h_{comNOxi} \left(a_{NOxi} P_i^3 + b_{NOxi} P_i^2 + c_{NOxi} P_i + d_{NOxi} \right) \end{bmatrix} \quad [\$/h] \qquad (3.64)$$

$$F_{TcomCO_2} = \sum_{i=1}^{n} \begin{bmatrix} (a_i P_i{}^3 + b_i P_i^2 + c_i P_i + d_i) + \\ + h_{comCO_2 i} \left(a_{Co_2 i} P_i^3 + b_{cO_{2i}} P_i^2 + c_{co_2 i} P_i + d_{co_2 i} \right) \end{bmatrix} \quad [\$/h] \qquad (3.65)$$

The objective function (3.56) to (3.65) are solved under the power balance constraint, given by Equation (3.2) and generator operational limit constraint given by Equation (3.4). It is assumed that the line losses are minimal and they are not considered. Equation (3.2) can be written as Equation (3.66) by neglecting the transmission losses as:

$$\sum_{i=1}^{n} P_i = P_D \qquad [MW] \qquad (3.66)$$

The problem for bi-criteria optimization of the real power are formulated for every pollutant separately in the following way: Find the optimal value of the real power produced by the generators in such a way that the criteria (3.56) to (3.58) and (3.60) to (3.65) are minimized under the constraints Equations (3.2) and (3.66)

The problem for multi-criteria optimization of the real power is formulated for all three pollutants together in the following way: Find the optimal value of the real power produced by the generators in such a way that the criterion (3.59) is minimized under the constraints (3.2) and (3.66).

3.4.4 Formulation of the price penalty factors

As described in most of the existing literature the price penalty factor h_i is the ratio between maximum fuel cost and maximum emission of corresponding generator (Krishnamurthy and Tzoneva, 2011), (Balamurugan and Subramanian, 2007), and (Ming et al., 2010). The Max-Max price penalty factor is formulated for all the three pollutants individually as follows:

$$h_{SO_2 i} = \frac{(a_i P_{i\max}{}^3 + b_i P_{i\max}{}^2 + c_i P_{i\max} + d_i)}{(a_{so_2 i} P_{i\max}^3 + b_{so_2 i} P_{i\max}^2 + c_{so_2 i} P_{i\max} + d_{so_2 i})}, \; i = \overline{1,n} \quad [\$/kg] \qquad (3.67)$$

$$h_{NOxi} = \frac{(a_i P_{i\max}{}^3 + b_i P_{i\max}{}^2 + c_i P_{i\max} + d_i)}{(a_{NOxi} P_{i\max}^3 + b_{NOxi} P_{i\max}^2 + c_{NOxi} P_{i\max} + d_{NOxi})}, \; i = \overline{1,n} \quad [\$/kg] \qquad (3.68)$$

$$h_{CO_2 i} = \frac{(a_i P_{i\max}{}^3 + b_i P_{i\max}{}^2 + c_i P_{i\max} + d_i)}{(a_{Co_2 i} P_{i\max}^3 + b_{CO_2 i} P_{i\max}^2 + c_{Co_2 i} P_{i\max} + d_{Co_2 i})},\ i = \overline{1,n} \quad [\$/kg] \quad (3.69)$$

Where

$h_{SO_2 i}$ - Price penalty factor of SO_2 emission

h_{NOxi} - Price penalty factor of NO_x emission

$h_{CO_2 i}$ - Price penalty factor of CO_2 emission

This research work proposes a new price penalty factor “*Min-Max”* and "*Max-Min*" in addition to the described ones in literature to solve the bi-objective economic dispatch problem. The Min-Max price penalty factor is formulated as follows:

$$h_{SO_2 i} = \frac{(a_i P_{i\min}{}^3 + b_i P_{i\min}{}^2 + c_i P_{i\min} + d_i\ )}{(a_{so_2 i} P_{i\max}^3 + b_{so_2 i} P_{i\max}^2 + c_{so_2 i} P_{i\max} + d_{so_2 i})},\ i = \overline{1,n} \quad [\$/kg] \quad (3.70)$$

$$h_{NOxi} = \frac{(a_i P_{i\min}{}^3 + b_i P_{i\min}{}^2 + c_i P_{i\min} + d_i\ )}{(a_{NOxi} P_{i\max}^3 + b_{NOxi} P_{i\max}^2 + c_{NOxi} P_{i\max} + d_{NOxi})},\ i = \overline{1,n} \quad [\$/kg] \quad (3.71)$$

$$h_{CO_2 i} = \frac{(a_i P_{i\min}{}^3 + b_i P_{i\min}{}^2 + c_i P_{i\min} + d_i\ )}{(a_{Co_2 i} P_{i\max}^3 + b_{CO_2 i} P_{i\max}^2 + c_{Co_2 i} P_{i\max} + d_{Co_2 i})},\ i = \overline{1,n} \quad [\$/kg] \quad (3.72)$$

The Min-Min price penalty factor is formulated as follows:

$$h_{SO_2 i} = \frac{(a_i P_{i\min}{}^3 + b_i P_{i\min}{}^2 + c_i P_{i\min} + d_i\ )}{(a_{so_2 i} P_{i\min}^3 + b_{so_2 i} P_{i\min}^2 + c_{so_2 i} P_{i\min} + d_{so_2 i})},\ i = \overline{1,n} \quad [\$/kg] \quad (3.73)$$

$$h_{NOxi} = \frac{(a_i P_{i\min}{}^3 + b_i P_{i\min}{}^2 + c_i P_{i\min} + d_i\ )}{(a_{NOxi} P_{i\min}^3 + b_{NOxi} P_{i\min}^2 + c_{NOxi} P_{i\min} + d_{NOxi})},\ i = \overline{1,n} \quad [\$/kg] \quad (3.74)$$

$$h_{CO_2 i} = \frac{(a_i P_{i\min}{}^3 + b_i P_{i\min}{}^2 + c_i P_{i\min} + d_i\ )}{(a_{Co_2 i} P_{i\min}^3 + b_{CO_2 i} P_{i\min}^2 + c_{Co_2 i} P_{i\min} + d_{Co_2 i})},\ i = \overline{1,n} \quad [\$/kg] \quad (3.75)$$

The proposed Max-Min price penalty factor is formulated as follows:

$$h_{SO_2 i} = \frac{(a_i P_{i\max}{}^3 + b_i P_{i\max}{}^2 + c_i P_{i\max} + d_i\ )}{(a_{so_2 i} P_{i\min}^3 + b_{so_2 i} P_{i\min}^2 + c_{so_2 i} P_{i\min} + d_{so_2 i})},\ i = \overline{1,n} \quad [\$/kg] \quad (3.76)$$

$$h_{NOxi} = \frac{(a_i P_{i\max}{}^3 + b_i P_{i\max}{}^2 + c_i P_{i\max} + d_i\ )}{(a_{NOxi} P_{i\min}^3 + b_{NOxi} P_{i\min}^2 + c_{NOxi} P_{i\min} + d_{NOxi})},\ i = \overline{1,n} \quad [\$/kg] \quad (3.77)$$

$$h_{CO_2 i} = \frac{(a_i P_{i\max}{}^3 + b_i P_{i\max}{}^2 + c_i P_{i\max} + d_i\ )}{(a_{Co_2 i} P_{i\min}^3 + b_{CO_2 i} P_{i\min}^2 + c_{Co_2 i} P_{i\min} + d_{Co_2 i})},\ i = \overline{1,n} \quad [\$/kg] \quad (3.78)$$

The average and common price penalty factors are proposed by (Balamurugan and Subramanian, 2007) to solve the bi-objective economic emission dispatch problem are also considered. The thesis proposes variants of these penalty factors when four different specific penalty factors Min-Max, Max-Max, Min-Min, Max-Min are considered.

$$h_{AvgSO2i} = \frac{\begin{bmatrix} F_C(P_{i\min})/E_{SO_2}(P_{i\max}) + F_C(P_{i\max})/E_{SO_2}(P_{i\max}) + \\ +F_C(P_{i\min})/E_{SO_2}(P_{i\min}) + F_C(P_{i\max})/E_{SO_2}(P_{i\min}) \end{bmatrix}}{4}, \; i = \overline{1,n} \quad [\$/kg] \quad (3.79)$$

$$h_{AvgNOxi} = \frac{\begin{bmatrix} F_C(P_{i\min})/E_{NO_x}(P_{i\max}) + F_C(P_{i\max})/E_{NO_x}(P_{i\max}) + \\ +F_C(P_{i\min})/E_{NO_x}(P_{i\min i}) + F_C(P_{i\max})/E_{NO_x}(P_{i\min}) \end{bmatrix}}{4}, \; i = \overline{1,n} \quad [\$/kg] \quad (3.80)$$

$$h_{AvgCO2i} = \frac{\begin{bmatrix} F_C(P_{i\min})/E_{CO_2}(P_{i\max}) + F_C(P_{i\max})/E_{CO_2}(P_{i\max}) + \\ +F_C(P_{i\min})/E_{CO_2}(P_{i\min}) + F_C(P_{i\max})/E_{CO_2}(P_{i\min}) \end{bmatrix}}{4}, \; i = \overline{1,n} \quad [\$/kg] \quad (3.81)$$

Where

h_{AvgSO_2i} - Average price penalty factor of SO_2 emission

$h_{AVgNOxi}$ - Average price penalty factor of NO_x emission

h_{AvgCO_2i} - Average price penalty factor of CO_2 emission

The common price penalty factor for various pollutants is expressed as follows:

$$h_{ComSo2i} = \sum_{i=1}^{n} \frac{h_{AvgSO2i}}{n}, \; i = \overline{1,n} \quad [\$/kg] \quad (3.82)$$

$$h_{ComNoxi} = \sum_{i=1}^{n} \frac{h_{AvgNOxi}}{n}, \; i = \overline{1,n} \quad [\$/kg] \quad (3.83)$$

$$h_{ComCo2i} = \sum_{i=1}^{n} \frac{h_{AvgCO2i}}{n}, \; i = \overline{1,n} \quad [\$/kg] \quad (3.84)$$

Where

h_{ComSO_2i} - Common price penalty factor of SO_2 emission

$h_{ComNOxi}$ - Common price penalty factor of NO_x emission

h_{ComCO_2i} - Common price penalty factor of CO_2 emission

3.4.5 Formulation of Lagrange’s algorithm for multi-objective economic emission dispatch (MEED) problem

Lagrange’s variables are introduced to the objective functions (3.56), or (3.57), or (3.58). When consider the equation (3.56) alone, the dispatch problem is formulated using the Lagrange’s variable as follows: (for the case of SO_2):

$$L_{SO_2} = F_{TSO_2} + \lambda(P_D - \sum_{i=1}^{n} P_i) \tag{3.85}$$

$$L_{SO_2} = \sum_{i=1}^{n} \left[\begin{array}{l} (a_i P_i^{\,3} + b_i P_i^2 + c_i P_i + d_i\ ) + \\ h_{SO_2 i} \left(a_{so_2 i}\ P_i^3 + b_{so_2 i}\ P_i^2 + c_{so_2 i} P_i + d_{so_2 i} \right) \end{array} \right] + \lambda(P_D - \sum_{i=1}^{n} P_i) \tag{3.86}$$

Similar Lagrange's functions can be built for other pollutants NOx and CO_2

The optimization problem (3.56), subject to the constraints (3.2), and (3.66) is transferred to a problem for minimization of L_{SO2} according to $P_{i,}\ i = \overline{1, n},$ and maximization of L_{SO2} according to λ.

The function of Lagrange for the case of impact of all 3 pollutants is given by equation (3.87)

$$F_{T_3} = \sum_{i=1}^{n} \left[\begin{array}{l} (a_i P_i^{\,3} + b_i P_i^2 + c_i P_i + d_i) + \\ + h_{SO_2 i} \left(a_{so_2 i}\ P_i^3 + b_{so_2 i}\ P_i^2 + c_{so_2 i} P_i + d_{so_2 i} \right) + \\ + h_{NOxi} \left(a_{NOxi}\ P_i^3 + b_{NOxi}\ P_i^2 + c_{NOxi} P_i + d_{NOxi} \right) + \\ + h_{CO_2 i} \left(a_{Co_2 i}\ P_i^3 + b_{cO_{2i}} P_i^2 + c_{co_2 i} P_i + d_{co_2 i} \right) \end{array} \right] + \lambda \left(P_D - \sum_{i=1}^{n} P_i \right) \tag{3.87}$$

Necessary conditions for optimality for solution of the problem (3.86) and (3.87) are:

According to P_i, $\quad \dfrac{\partial L}{\partial P_i} = 0, i = \overline{1, n}$ (3.88)

According to λ, $\quad \dfrac{\partial L}{\partial \lambda} = 0$ (3.89)

Derivation of the condition (3.88) by considered the single pollutant SO_2 alone is given in Equation (3.90):

$$\frac{\partial L}{\partial P_i} = 3a_i P_i^2 + 2b_i P_i + c_i + h_{SO_2 i}(3a_{SO_2 i} P_i^2 + 2b_{SO_2 i} P_i + c_{SO_2 i}) - \lambda = 0 \tag{3.90}$$

$$3P_i^2 \left(a_i + h_{SO_2 i} a_{SO_2 i} \right) + 2P_i \left(b_i + h_{SO_2 i} b_{SO_2 i} \right) + \left(c_i + h_{SO_2 i} c_{SO_2 i} \right) - \lambda = 0 \tag{3.91}$$

Equation (3.91) has the structure of a quadratic equation as:

$$x_i P_i^2 + y_i P_i + z_i = 0 \tag{3.92}$$

where

$$\begin{array}{l} x_i = 3\left(a_i + h_{SO_2 i} a_{SO_2 i} \right)\ \ , i = \overline{1, n} \\ y_i = 2\left(b_i + h_{SO_2 i} b_{SO_2 i} \right)\ , i = \overline{1, n} \\ z_i = \left(c_i + h_{SO_2 i} c_{SO_2 i} \right) - \lambda\ , i = \overline{1, n} \end{array}$$

The necessary conditions for optimality of the Lagrange's function (3.87) by considered all the three pollutants according to the real power are derived as follows:

$$\frac{\partial L}{\partial P_i} = \begin{bmatrix} (3a_i P_i^2 + 2b_i P_i + c_i + d_i) + h_{SO_2 i}\left(3a_{so_2 i} P_i^2 + b_{so_2 i} P_i + c_{so_2 i}\right) + \\ + h_{NOxi}\left(3a_{NOxi} P_i^2 + 2b_{NOxi} P_i + c_{NOxi}\right) + \\ + h_{CO_2 i}\left(3a_{Co_2 i} P_i^2 + 2b_{cO_{2i}} P_i + c_{co_2 i}\right) - \lambda \end{bmatrix} = 0 \ , \ i = \overline{1,n} \quad (3.93)$$

From here the equation (3.93) can be written as

$$\begin{bmatrix} 3P_i^2 (a_i + h_{SO_2 i} a_{so_2 i} + h_{NOxi} a_{NOxi} + h_{CO_2 i} a_{Co_2 i}) + \\ + 2P_i \left(b_i + h_{SO_2 i} b_{so_2 i} + h_{NOxi} b_{NOxi} + h_{CO_2 i} b_{cO_{2i}}\right) + \\ + \left(c_i + h_{SO_2 i} c_{so_2 i} + h_{NOxi} c_{NOxi} + h_{CO_2 i} c_{co_2 i}\right) - \lambda \end{bmatrix} = 0 \ , \ i = \overline{1,n} \quad (3.94)$$

The Equation (3.94) can be presented as a quadratic equation with the coefficients given as follows:

$$\begin{aligned} x_i &= 3(a_i + h_{SO_2 i} a_{so_2 i} + h_{NOxi} a_{NOxi} + h_{CO_2 i} a_{Co_2 i}) \ , \ i = \overline{1,n} \\ y_i &= 2\left(b_i + h_{SO_2 i} b_{so_2 i} + h_{NOxi} b_{NOxi} + h_{CO_2 i} b_{cO_{2i}}\right) \ , \ i = \overline{1,n} \\ z_i &= \left(c_i + h_{SO_2 i} c_{so_2 i} + h_{NOxi} c_{NOxi} + h_{CO_2 i} c_{co_2 i}\right) - \lambda, \ i = \overline{1,n} \end{aligned} \quad (3.95)$$

The optimal real power of each generator is found by using Equations (3.91) or (3.92) for single pollutant SO_2 and for all three pollutants given in Equation (3.94). The solution is based on the formula for solution of a quadratic equation:

For a single pollutant:

$$P_i = \frac{\left(-y_i \pm sqrt\left(y_i^2 - 4x_i z_i\right)\right)}{2x_i} \ , \ i = \overline{1,n} \quad (3.96)$$

Substitute the values of x_i, y_i, z_i in Equation (3.96) and the real power of the generator is given in Equation (3.97)

$$P_i = real \frac{\left(-\left[2\left(b_i + h_{SO_2 i} b_{SO_2 i}\right)\right] + sqrt\left(\begin{matrix}\left[2\left(b_i + h_{SO_2 i} b_{SO_2 i}\right)\right]^2 - \\ -4\left[3\left(a_i + h_{SO_2 i} a_{SO_2 i}\right)\right]\left[\left(c_i + h_{SO_2 i} c_{SO_2 i}\right) - \lambda\right]\end{matrix}\right)\right)}{2\left[3\left(a_i + h_{SO_2 i} a_{SO_2 i}\right)\right]} \quad (3.97)$$

For all 3 pollutants, derivation of the condition (3.88) is given in Equation (3.98) as:

$$P_i = \frac{\begin{bmatrix} \lambda - 3P_i^2(a_i + h_{SO_2i}a_{so_2i} + h_{NOxi}a_{NOxi} + h_{CO_2i}a_{Co_2i}) - \\ -\left(c_i + h_{SO_2i}c_{so_2i} + h_{NOxi}c_{NOxi} + h_{CO_2i}c_{co_2i}\right) \end{bmatrix}}{2\left(b_i + h_{SO_2i}b_{so_2i} + h_{NOxi}b_{NOxi} + h_{CO_2i}b_{cO_{2i}}\right)}, \; i = \overline{1,n} \tag{3.98}$$

Equation (3.98) has the structure of a quadratic equation as given in (3.96), the solution of the generator real power P_i is found by solving the Equation (3.98) as:

$$P_i = real \frac{\begin{pmatrix} -\left[2\left(b_i + h_{SO_2i}b_{so_2i} + h_{NOxi}b_{NOxi} + h_{CO_2i}b_{cO_{2i}}\right)\right] + \\ sqrt\begin{pmatrix} \left[2\left(b_i + h_{SO_2i}b_{so_2i} + h_{NOxi}b_{NOxi} + h_{CO_2i}b_{cO_{2i}}\right)\right]^2 - \\ -4\begin{bmatrix} \left[3(a_i + h_{SO_2i}a_{so_2i} + h_{NOxi}a_{NOxi} + h_{CO_2i}a_{Co_2i})\right]x \\ x\left[\left(c_i + h_{SO_2i}c_{so_2i} + h_{NOxi}c_{NOxi} + h_{CO_2i}c_{co_2i}\right) - \lambda\right] \end{bmatrix} \end{pmatrix} \end{pmatrix}}{2\left[3(a_i + h_{SO_2i}a_{so_2i} + h_{NOxi}a_{NOxi} + h_{CO_2i}a_{Co_2i})\right]}, \; i = \overline{1,n} \tag{3.99}$$

If the value of the Lagrange's multiplier λ is known the equations (3.97) can be solved according to the unknown vector P_i.

The value of λ is unknown and has to be found from the necessary condition for optimality (3.89). This condition is:

$$\frac{\partial L}{\partial \lambda} = P_D - \sum_{i=1}^{n} P_i = 0 = \Delta\lambda \tag{3.100}$$

The gradient procedure for calculation of λ has to be developed as follows:

$$\lambda^{(k+1)} = \lambda^k + \alpha\,\Delta\lambda^{(k)}, \lambda \neq 0 \tag{3.101}$$

The obtained solutions for $P_i, \; i = \overline{1,n}$ have to belong to the constraint domain (3.66) and the obtained solution is fit to the constraint domain given in (3.101).

$$P_i = \begin{Bmatrix} P_{i,\min}, & if \;\; P_i \langle P_{i,\min} \\ P_i, \; if & P_{i.\min} \le P_i \langle P_{i,\max} \\ P_{i,\max}, & if \;\; P_i \rangle P_{i,\max} \end{Bmatrix} \tag{3.102}$$

The condition for end of the iteration is

$$\Delta\lambda^{(k)} \le \varepsilon \text{ or } k = m \tag{3.103}$$

The algorithm of the Lagrange's method is:

1) Initial value of the Lagrange's multiplier is guessed: $\lambda^{(0)}$,
2) Equation (3.97) for single pollutant or Equation (3.99) for all three pollutants is solved and the unknown vector P_i is determined.
3) The obtained vector P_i is fit to the constraints (3.102).

4) The optimal solution is used to calculate the CEED problem fuel cost using Equation (3.56) for single pollutants and Equation (3.59) for all three pollutants.

The flowchart of the algorithm is given in Figure 3.14.

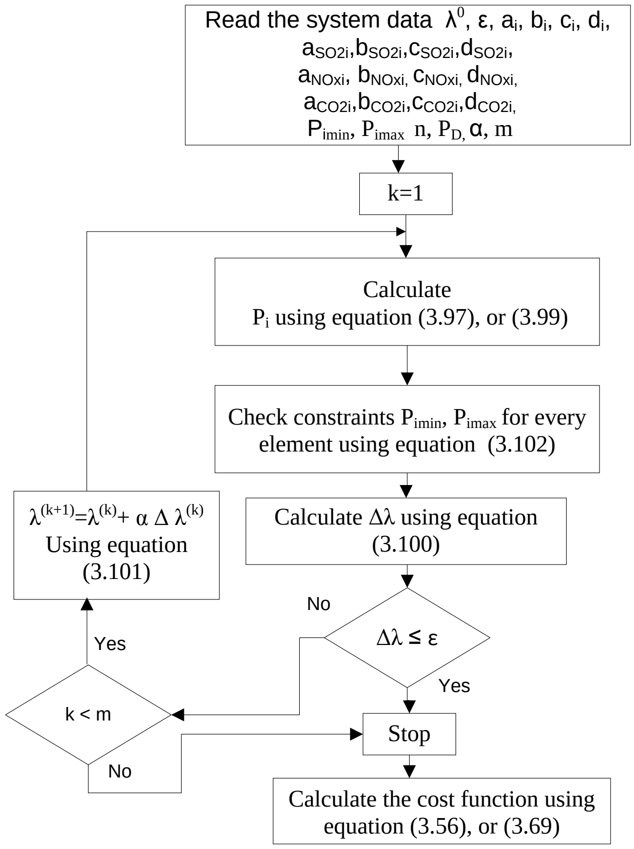


Figure 3.14: **Flow diagram of the cubic function bi-criteria and multi-criteria dispatch problem using Lagrange's method**

The software developed CEEDCubic.m for the CEED problem based on cubic objective function is given in Appendix C.

3.4.6 Results of the multi-objective dispatch problem based on cubic objective functions for various pollutants and penalty factors

This section applies the Lagrangian algorithm to solve the environmental economic dispatch problem for various penalty factors and pollutants. The fuel cost coefficients, SO_2, NOx, and CO_2 emission coefficients, and the real power limits of the generators are provided in (Chayakulkheere and Ongsakul, 2007) and in Table 3.23. The constant power demand of 286 [MW] is used to solve the CEED problem. The simulation is carried out in the software Matlab 2011 and the maximum number of iterations is selected to be m= 50000 iterations. The initial lambda value is set as 40. The simulation results of the CEED-SO_2 emission solution for the considered six penalty factors such as i) Min-Max, ii) Max-Max, iii) Min-Min iv)Max-Min v) Average, and vi) Common is given in Table 3.24, where the optimum real power of the generators, fuel cost, SO_2 emission values, combined economic SO_2 emission dispatch cost values and Computation Time (CT) are provided for the various penalty factors.

The above solutions are repeated for the dispatch problems considering NOx and CO_2 pollutants. The obtained solutions are given in Table 3.25 and Table 3.26 respectively. The results of the solution of the dispatch problem by considering all three pollutants is given in Table 3.27.

The obtained optimal values of the fuel cost, SO_2, NO_x, and CO_2 pollutant quantities, CEED-SO_2, CEED-NO_x, CEED-CO_2, CEED-SO_2-NOx-CO_2 and Computation Time are given in Table 3.24 to Table 3.27 for each type of penalty factors. Table 3.28 compares the solution of CEED problem using the Lagrange's developed algorithm with the Fuzzy Multi-objective Optimal real Power Flow (FMOPF) algorithm given by (Chayakulkeree and Ongsakul, 2007).

Table 3.23: **IEEE 30 bus system coefficients for the cubic objective functions of the fuel cost and the emissions**

Bus Number		1	2	5	8	11	13
Gen. limits	P_{min} [MW]	50.0000	20.0000	15.0000	10.0000	10.0000	12.0000
	P_{max} [MW]	200.0000	80.0000	50.0000	50.0000	50.0000	40.0000
Fuel cost coefficients	a_i [$/MW^{3}h]	0.0010	0.0004	0.0006	0.0002	0.0013	0.0000
	b_i [$/MW^{2}h]	0.0920	0.0250	0.0750	0.1000	0.1200	0.0840
	c_i [$/MWh]	14.5000	22.0000	23.0000	13.5000	11.5000	12.5000
	d_i [$/h]	-136.0000	-3.5000	-81.0000	-14.5000	-9.7500	75.6000
SO_2 emission coefficients	a_{SO2i} [$/t^{3}h]	0.0005	0.0014	0.0010	0.0020	0.0013	0.0021
	b_{SO2i} [$/t^{2}h]	0.1500	0.0550	0.0350	0.0700	0.1200	0.0800
	c_{SO2i} [$/t.h]	17.0000	12.0000	10.0000	23.5000	21.5000	22.5000
	d_{SO2i} [$/h]	-90.0000	-30.5000	-80.0000	-34.5000	-19.7500	25.6000
NO_x emission coefficients	a_{NOxi} [$/t^{3}h]	0.0012	0.0004	0.0016	0.0012	0.0003	0.0014
	b_{NOxi} [$/t^{2}h]	0.0520	0.0450	0.0500	0.0700	0.0400	0.0240
	c_{NOxi} [$/t.h]	18.5000	12.0000	13.0000	17.5000	8.5000	15.5000
	d_{NOxi} [$/h]	-26.0000	-35.0000	-15.0000	-74.0000	-89.0000	-75.0000
CO_2 emission coefficients	a_{CO2i} [$/t^{3}h]	0.0015	0.0014	0.0016	0.0012	0.0023	0.0014
	b_{CO2i} [$/t^{2}h]	0.0920	0.0250	0.0550	0.0100	0.0400	0.0800
	c_{CO2i} [$/t.h]	14.0000	12.5000	13.5000	13.5000	21.0000	22.0000
	d_{CO2i} [$/h]	-16.0000	-93.5000	-85.0000	-24.5000	-59.0000	-70.0000

Table 3.24: **CEED-SO_2 emission dispatch problem solution**

h_i [$/t]	Real power / Lambda	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	F_C [$/hr]	E_{TSO2} [t/h]	F_{TSO2} [$/h]	Iter	CT [s]
Min-Max	36.52	55.24	58.84	32.09	50.00	49.85	40.00	5997.95	7116.75	6980.77	93	0.02
Max-Max	65.19	50.00	60.09	35.90	50.00	50.00	40.00	5984.48	7032.01	12005.88	344	0.13
Min-Min	77.80	69.97	48.61	27.40	50.00	50.00	40.00	6136.46	7361.12	12867.67	322	0.09
Max-Min	311.36	50.00	57.68	38.31	50.00	50.00	40.00	5989.14	6992.77	54088.52	2263	0.39
Average	126.01	50.00	58.27	37.72	50.00	50.00	40.00	5987.76	7001.82	21566.60	832	0.31
Common	127.69	50.52	64.33	50.00	40.90	40.24	40.00	6076.71	6683.44	23951.66	474	0.40

Table 3.25: **CEED-NO_x emission dispatch problem solution**

h_i [$/t]	Real power / Lambda	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	F_C [$/hr]	E_{TNOx} [t/h]	F_{TNOx} [$/h]	Iter	CT [s]
Min-Max	36.06	55.38	59.45	32.51	50.00	48.68	40.00	5994.58	5051.73	7004.59	85	0.03
Max-Max	62.67	50.76	61.55	36.32	50.00	47.36	40.00	5978.69	5005.30	11983.10	189	0.04
Min-Min	77.08	70.22	66.70	49.07	50.00	10.00	40.00	6441.56	5760.84	14331.19	276	0.06
Max-Min	470.45	55.99	80.00	50.00	50.00	10.00	40.00	6359.20	5560.56	57881.32	692	1.50
Average	159.32	55.99	80.00	50.00	50.00	10.00	40.00	6359.20	5560.56	23172.72	2055	0.38
Common	4109.24	50.00	73.93	41.09	30.97	50.00	40.00	6143.36	4870.60	793718.43	17847	3.73

Table 3.26: **CEED-CO_2 emission dispatch problem solution**

h_i [$/t]	Real power / Lambda	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	F_C [$/hr]	E_{TCO2} [t/h]	F_{TCO2} [$/h]	Iter	CT [s]
Min-Max	36.41	56.42	58.08	32.19	50.00	49.32	40.00	6001.91	5990.13	6945.01	92	0.03

Max-Max	63.45	55.22	57.25	34.61	50.00	48.91	40.00	5996.28	5957.41	11535.91	217	0.05
Min-Min	81.21	67.82	41.40	36.77	50.00	50.00	40.00	6132.82	6187.72	13482.75	361	0.07
Max-Min	324.17	50.00	48.65	47.34	50.00	50.00	40.00	6031.42	5915.22	56967.74	2341	0.47
Average	130.35	50.00	50.69	45.30	50.00	50.00	40.00	6018.40	5907.95	22347.09	877	0.22
Common	119.96	50.00	61.62	48.07	50.00	36.34	39.96	6025.49	5788.55	22608.28	436	0.08

Table 3.27: CEED-SO_2-NO_x-CO_2 emission multi-criteria dispatch problem solution

h_i [$/t]	Real power / Lambda	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	F_C [$/hr]	E_{TSO2} [t/h]	E_{TNOx} [t/h]	E_{TCO2} [t/h]	$F_{TSO2-NOx-CO2}$ [$/h]	Iter	CT [s]
Min-Max	47.32	65.53	51.88	28.58	50.00	50.00	40.00	6079.05	7273.37	5219.00	6151.03	8883.22	152	0.09
Max-Max	130.93	51.68	58.31	36.01	50.00	50.00	40.00	5988.74	7035.15	4996.54	5936.70	23573.46	600	0.15
Min-Min	195.64	87.28	54.53	44.18	50.00	10.00	40.00	6730.47	7361.65	6194.84	6560.65	31080.22	903	0.16
Max-Min	1438.48	55.99	80.00	50.00	50.00	10.00	40.00	6359.20	7012.67	5560.56	6040.75	172702.82	18640	3.52
Average	47.32	65.53	51.88	28.58	50.00	50.00	40.00	6079.05	7273.37	5219.00	6151.03	8883.22	152	0.09
Common	130.93	51.68	58.31	36.01	50.00	50.00	40.00	5988.74	7035.15	4996.54	5936.70	23573.46	600	0.15

Table 3.28: Comparison of the CEED problem with a cubic fuel cost and emission objective functions solutions using the developed Lagrange's method with FMOPF solutions (Chayakulkeree and Ongsakul, 2007)

Solution	CEED-SO_2		CEED-NO_x		CEED-CO_2		CEED-SO_2- NOx- CO_2	
Algorithm	Lagrange's	FMOPF	Lagrange's	FMOPF	Lagrange's	FMOPF	Lagrange's	FMOPF
F_C [$/h]	5997.95	6151.39	7116.75	6709.76	5041.89	5009.04	5987.39	5976.29
E_{TSO2} [t/h]	5994.58	6161.39	7098.74	6874.94	5051.73	4897.48	5971.57	6104.2
E_{TNOx} [t/h]	6001.91	6142.73	7116.1	6766.27	5062.1	5187.67	5990.13	5805.99
E_{TCO2} [t/h]	6079.05	6068.14	7273.37	6822.92	5219	5018.07	6151.03	5889.96

FMOPF : Fuzzy Multi-objective Optimal real Power Flow (Chayakulkeree and Ongsakul, 2007)

3.4.7 Discussion on results of the CEED problem solution with a cubic fuel cost and emission functions

The following results are found for the cubic cost model of the CEED problem:

i. The Min-Max penalty factor is good to yield the minimum CEED fuel cost for the individual pollutants (F_T-SO_2, F_T-NOx, F_T-CO_2) and for all three pollutants together (F_{T3}-SO_2-NOx-CO_2). The solution is calculated for the minimum computation time in comparison with other price penalty factors.

ii. The Max-Max penalty factor is good to yield the minimum fuel cost for the individual pollutants (F_C-SO_2, F_C-NOx , F_C-CO_2) in comparison with the other price penalty factors.

iii. The Common penalty factor is good to yield minimum emission for the individual pollutants (E_T-SO_2, E_T-NOx, and E_T-CO_2).

3.5 Conclusion

This chapter solves the combined economic emission dispatch problem more accurately, by considered both fuel cost and emission criterion for the three types of the criterion functions, i.e., i) quadratic criterion functions without valve point loading effect, ii) quadratic criterion functions with valve point loading effect, and iii) cubic criterion functions. Various pollutants of thermal power plants such as SO_2, NO_x and CO_2 emissions are also considered in the cubic model of the CEED problem. . In addition to that, various types of price penalty factors are used in the bi-criteria dispatch problem. The role of all penalty factors is to transfer the physical meaning of the emission criterion from weight of the emission to the fuel cost for the emission. A conclusion can be made that the developed Min-Max price penalty factor provides minimum fuel cost and emission values in comparison with other types of penalty factors for all the three types of the criterion functions considered.

Chapter four developed PSO algorithm to solve the CEED problem using various price penalty factors.

CHAPTER FOUR

COMBINED ECONOMIC EMISSION DISPATCH PROBLEM SOLUTION BASED ON THE PARTICLE SWARM OPTIMIZATION ALGORITHM

4.I Introduction

The Combined Economic Emission Dispatch (CEED) problem is solved using Particle Swarm Optimization (PSO) algorithm in this chapter. The literatures (Palanichamy and Babu, 2008), (Srikrishna and Palanichamy, 1991), (Kulkari et al., 2000), (Balamurugan and Subramanian, 2008) solve the CEED problem by minimizing both fuel cost and emission by using price penalty factors, where the combined solution is given by the criterion $F_T = F_C + h E_T$ and h is the penalty factor. Different types of h are reported in literature as Max-Max, Min-Min, Average and Common (described in Chapter 3) and they are used so far for the combined economic emission dispatch (CEED) problem solution. The literatures proposing PSO and hybrid PSO algorithms for economic emission dispatch problem are only based on Max-Max price penalty factor so far. This chapter proposes Min-Max penalty factor to be used in addition to Max-Max one for formulation and solution of the CEED problem by the PSO algorithm. Comparison of the impact of the Min-Max and Max-Max penalty factors over the solution of the CEED problem is done on the basis of case study of IEEE 30 bus system and 11 generators system.

This Chapter describes the PSO algorithm in part 4.2, development of the PSO method for solution of the CEED problem in part 4.3, Application of the PSO method for the IEEE power systems models in part 4.4, comparison of Lagrange's and PSO solution of the CEED problem in part 4.5, and a discussion and conclusions are given in part 4.6 and 4.7 respectively.

4.2 Basic PSO algorithm

Particle Swarm Optimization (PSO) is a population based stochastic optimization technique developed by (Eberhart and Kennedy, 1995), inspired by the social behavior of bird flocking or fish schooling. PSO is initialized with a group of random particles and then it searches for an optima by updating the generation. In every iteration, each particle is updated by following two best values. The first one is the best solution it has achieved so far between all particles. This value is called P^{best}. Another best value that is tracked by the PSO algorithm is the best value, obtained so far by any particle in the population. This best value is a global best one and is called G^{best}. After finding the two best values, the particle updates its velocity and positions with following equations (4.1) and (4.2).

$$V[] = \begin{bmatrix} \omega V[\,] + c1 rand1(\,) * \left(P^{best}[\,] - Ppresent[\,]\right) + \\ + c2 rand2(\,)\left(G^{best}[\,] - Ppresent[\,]\right) \end{bmatrix} \tag{4.1}$$

$$Ppresent[] = Ppresent[] + V[] \tag{4.2}$$

Where

$V[\,]$	Particles velocity
ω	Inertia weight and is calculated as,

$$\omega = \omega^{\max} - \left(\frac{\omega^{\max} - \omega^{\min}}{Iter^{\max}}\right) Iter$$

$\omega^{\min} = 0.4$ and $\omega^{\max} = 0.9$

$Ppresent[]$	Position of the current particle
$P^{best}[]$	Best solution of the particle at each iteration
$G^{best}[]$	Best solution out of all the best particle solutions
rand1( ) and *rand2( )*	Random number between (0,1)
c1 and *c2*	Learning factors (usually *c1*=*c2*=2)
Iter	Total number of iterations
$Iter^{max}$	Maximum number of iterations

The thermal and hydro power plants dispatch optimization using various computational algorithms with case studies is described in (Kothari and Dhillon, 2006). The earlier work on economic emission load dispatch is proposed in (Dhillon et al., 1993) using stochastic algorithm. The PSO algorithm was introduced by Eberhart and Kennedy in 1995. Review of PSO algorithm for economic dispatch problem is presented in (Amita Mahor et al., 2009). In reference (Chen and Wnag,2010) the environmental/economic dispatch and multi-area environmental/economic dispatch problem are solved to minimize the fuel cost and emission simultaneously. The security constrained economic dispatch is solved in (Gaing and Chang, 2006) to minimize the total generation cost with valve point effect loading under the constraints such as transmission loss, bus voltage profile under pre-contingent and postcontingent states, and generator ramp rate limits. Emission constrained economic dispatch is solved using various PSO algorithms such as Self Adaptive PSO, Dispersed PSO, Chaotic PSO and New PSO in (Rani and Yazdani, 2011). The non-smooth economic dispatch problem is solved using PSO by proposing the position adjustment strategy to satisfy the inequality constraints and a dynamic search space reduction strategy is applied to accelerate the convergence speed. Economic dispatch problem is solved using Improved PSO in (Park et al., 2005 and 2010) overcoming the drawbacks in existing PSO such as local trapping to premature convergence (exploration problem), insufficient capability to find nearby extreme points (exploitation problem) and lack of efficient mechanism to treat the constraints (constraint handling problem). The combined economic emission dispatch problem (Venkatesh et al., 2003) is solved using novel modified price penalty factor by various evolutionary computation algorithms such as GA, Micro GA and EP. The Evolutionary programming technique is carried out in two parts in (Sinha et al., 2003), part 1 proposed the modification in existing EP by changing the adaptation based on the scaled cost, and part 2 developed the adaptation based on an empirical learning rate. Multi-objective evolutionary algorithm such as Non dominated Sorting GA(NSGA), Pareto GA (PGA) and Strength Pareto EA (SPEA) are applied to economic emission dispatch problem in (Abido, 2006), additionally the Pareto-optimal solution set is developed using fuzzy hierarchical clustering algorithm. The Fuzzy Clustering based PSO (FCPSO) is used in (Shubham Agrawal et al., 2008) to solve the dispatch problem. The fuzzy clustering manages the size of the repository within limits without destroying the characteristics of the Pareto. Front and niching mechanism is used to direct the particles towards the lesser explored regions of the Pareto front. Self-adaptive mutation operator is used to avoid the local optima and enhance the exploratory capability of the particles. In (Khamsawang et al., 2010) the mutation operator of the Differential Evolution (DE) is used for improving diversity exploration of the PSO called hybrid PSO-DE. Four scenario of mutation operators are proposed to be activated if the velocity values of PSO are nearly to zero or violated from the boundaries. The review paper (Amita et al., 2009), concludes that Particle Swarm Optimization (PSO) has a lot of applications for solution of the combined economic emission dispatch problems, as this solution will not suffer from stuck into local optimal solution, dependability on initial variables, premature, slow convergence and curse of dimensionality. In comparison to conventional optimization techniques, PSO has given an improved result within less computational time.

4.3 Development of a PSO algorithm for solution of the CEED problem

The CEED problem is formulated and solved in Chapter 3. The fuel cost equation is given in (3.1) and is solved subject to the constraints (3.2), (3.3.), and (3.4). The emission function is given in (3.5) and the price penalty factors (3.7) and (3.8) are used to formulate the single objective function (3.6) of the CEED problem. The above problem is solved by the basic PSO algorithm, which is adopted to incorporate the specifics of the CEED problem. It is necessary to map the structure of the CEED problem to the structure of the velocity and position Equations (4.1) and (4.2). This is done in the following way:

- It is accepted that the number of generators is equal to the number of the members in a separate particle in the swarm. For the dispatch problem the positions of the members of the particles represent the active power produced by the generators.
- The velocities are variables that have the meaning of the active power but are used to do search in the constraints domain.
- It is assumed that the number of the particles in the swarm is N_p. The developed PSO algorithm for solution of the combined economic dispatch problem is given in Equations (3.1) to (3.6) as follows:

First step of the iteration procedure start at *l=1*

Step1: Initialize the PSO parameters such as inertia weight ω^{min} and ω^{max}, acceleration constants *c1* and *c2*, uniform random values *rand1*, *rand2*, and maximum number of iterations $Iter^{max}$

Step2: Calculate the minimum and maximum initial velocities using the generator limit constraints in Equation (3.4) and are given in Equation (4.3) as follows:

$$-0.5P_{pi}^{\min} \le V_{pi} \le +0.5P_{pi}^{\max}, p=\overline{1,N_p}, i=\overline{1,n-1} \qquad (4.3)$$

Where

N_p Number of particles in the swarm

n Number of the members in one particle and it is equal to the total number of generators.

As one of the generators is accepted to be a slack one, the calculations for the velocity and position in the particles is done for *(n-1)* generators.

Step 3: Calculate the initial velocity of all members of the particles except slack bus generator using Equation (4.4)

$$V_{pi} = V_{pi}^{\min} + rand()\left(V_{pi}^{\max} - V_{pi}^{\min}\right), p = \overline{1, N_p}, i = \overline{1, n-1} \quad (4.4)$$

Where

$V_{pi}^{min}, V_{pi}^{max}$ Previously calculated minimum and maximum velocities respectively.

Step 4: Calculate the initial Position of the particle members using Equation (4.5)

$$P_{pi} = P_{pi}^{\min} + rand()\left(P_{pi}^{\max} - P_{pi}^{\min}\right), p = \overline{1, N_p}, i = \overline{1, n-1} \quad (4.5)$$

Check the calculated positions if they are within the limits given by Equation (3.4), as described

$$P_{pi} = \begin{cases} P_{pi}^{\min}, P_{pi} \le P_{pi}^{\min} \\ P_{pi}^{\max}, P_{pi} \ge P_{pi}^{\max} \\ P_{pi}, \quad P_{pi}^{\min} \le P_{pi} \le P_{pi}^{\max} \end{cases}, p = \overline{1, N_p},\ i = \overline{1, n-1} \quad (4.6)$$

The buses in the power system is classified as (i) Slack bus, (ii) Generator (PV) bus and (iii) Load (PQ) bus. In power system, any one of the buses which is connected with generator having highest power generation capacity is considered as the slack bus. Slack bus provides reference for the voltage and angle in the system. The slack bus real and reactive powers are uncontrolled: the bus supplies whatever real or reactive power is necessary to make the power flows in the system balance. The generation scheduling is carried to solve the economic dispatch problem without considering the generator real power of the slack bus. The purpose of the slack bus in PSO algorithm is to satisfy the power balance constraint given in Equation (3.2). The procedure for the calculation of slack bus generator real power in PSO algorithm is given in Step 5.

Step 5: The slack bus generator is considered as an dependent generator having highest power generating capacity in which voltage magnitude and phase angle of that bus is specified as reference in which the initial active power P_{pd} can be calculated on the basis of the power balance constraint as follows (Kothari and Dhillon, 2011).

$$P_{pd}+\sum_{\substack{i=1\\i\neq d}}^{n}P_{pi}=\left[\begin{array}{l}\sum_{\substack{i=1\\i\neq d}}^{n}\sum_{\substack{j=1\\j\neq d}}^{n}P_{pi}B_{ij}P_{pj}+\sum_{\substack{j=1\\j\neq d}}^{n}P_{pj}\left(B_{jd}+B_{dj}\right)P_{pd}+\\+B_{dd}P_{pd}{}^{2}+\sum_{\substack{i=1\\i\neq d}}^{n}B_{io}P_{pi}+B_{do}P_{pd}+B_{oo}+P_{D}\end{array}\right],\ p=\overline{1,N_p} \tag{4.7}$$

Where

P_{pd} is the power produced by the slack bus generator

P_D is the total power demand of the system

$\sum_{i=1,i\neq d}^{n}P_{pi}$ is the total active power of the power system excluding the slack bus power.

The Equation (4.7) can be transferred to the following quadratic form one, where the P_{pd} is the unknown variable:

$$XP_{pd}^2+YP_{pd}+Z=0 \tag{4.8}$$

Where

$$X=B_{dd} \tag{4.9}$$

$$Y=\sum_{\substack{j=1\\j\neq d}}^{n}\left(B_{jd}+B_{dj}\right)P_{pj}+B_{do}-1 \tag{4.10}$$

$$Z=P_D+B_{oo}+\sum_{\substack{i=1\\i\neq d}}^{n}\sum_{\substack{j=1\\j\neq d}}^{n}P_{pi}B_{ij}P_{pj}+\sum_{\substack{i=1\\i\neq d}}^{n}B_{io}P_{pi}-\sum_{\substack{i=1\\i\neq d}}^{n}P_{pi} \tag{4.11}$$

The positive root of the Equation (4.8) is obtained as:

$$P_{pd}=\frac{-Y+\sqrt{Y^2-4XZ}}{2X},whereY^2-4XZ\geq 0 \tag{4.12}$$

Now the real power vector is formed as $P_p=[P_{pd},P_{pi},\ i=1,n,\ i\neq d]$ where $p=\overline{1,N_p}$

Step 6: Calculate the following objective functions for the initial positions of the particles

i) Fuel cost function,

$$F_{Cp}=\sum_{i=1}^{n}(a_iP_{pi}{}^2+b_iP_{pi}+c_i),p=\overline{1,N_p} \tag{4.13}$$

ii) Emission function,

$$E_{Tp} = \sum_{i=1}^{n} \left(d_i P_{pi}^2 + e_i P_{pi} + f_i \right), p = \overline{1, N_p} \quad (4.14)$$

iii) Combined economic emission function,

$$F_{Tp} = \sum_{i=1}^{n} \left[(a_i P_{pi}{}^2 + b_i P_{pi} + c_i) + h_{pi} \left(d_i P_{pi}^2 + e_i P_{pi} + f_i \right) \right], p = \overline{1, N_p} \quad (4.15)$$

Where the Min-Max price penalty factor is calculated as

$$h_{pi} = \frac{(a_i P_{pi,\min}^2 + b_i P_{pi,\min} + c_i)}{\left(d_i P_{pi,\max}^2 + e_i P_{pi,\max} + f_i \right)}, \; i = \overline{1, n} \; , \; p = \overline{1, N_p} \quad (4.16)$$

The fuel cost values F_{TP} of all the particles are sorted in ascending order. The first position value in the ascending order is selected as the F_T^{best}

Step 7: Select the best initial position and the global best initial position as follows:

i) Initial positions of particles in the swarm are considered as best positions of particles $P_p^{best} = min\, P_{pi}^{best}, \, i = \overline{1,n} \; ; p = \overline{1, N_p}$

ii) The best position out of all the best particles $min\, P_p^{best}, p = \overline{1, N_p}$ is taken as $G^{best} = min\, P_p^{best}, p = \overline{1, N_p}$

l^{th} step of the iteration procedure begins, where $l=l+1$

Step 8: Calculate new velocity using Equation (4.17a) and (4.17b)

$$V_{pi}^{new^l} = \omega . V_{pi}^{l-1} + c1.rand1\left(P_p^{best^{l-1}} - P_{pi}^{l-1} \right) + c2.rand2\left(G^{best^{l-1}} - P_{pi}^{l-1} \right), \; p = \overline{1, N_p}, \; i = \overline{1, n-1} \quad (4.17)$$

Check the constraints for the minimum and maximum values of the velocities

If $V_{pi}^{new^l} > V_{pi}^{\max^{l-1}}, V_{pi}^{new^l} = V_{pi}^{\max^{l-1}}$ and (4.18)

If $V_{pi}^{new^l} < V_{pi}^{\min^{l-1}}, V_{pi}^{new^l} = V_{pi}^{\min^{l-1}}$, $p = \overline{1, N_p}, \; i = \overline{1, n-1}$

Step 9: Calculate the new position of the generators in the particles using Equation (4.18)

$$P_{pi}^{new^l} = P_{pi}^{l-1} + V_{pi}^{new^l}, \; p = \overline{1, N_p}, \; i = \overline{1, n} \quad (4.19)$$

Step 10: Check the new position of the generators in the particles using the constraint Equation (4.6) as follows:

$$P_{pi}^{new^l} = \begin{cases} P_{pi}^{\min}, P_{pi}^{new^l} \le P_{pi}^{\min} \\ P_{pi}^{\max}, P_{pi}^{new^l} \ge P_{pi}^{\max} \\ P_{pi}^{new}, \quad P_{pi}^{\min} \le P_{pi}^{new^l} \le P_{pi}^{\max} \end{cases}, p = \overline{1, N_p},\ i = \overline{1, n-1} \qquad (4.20)$$

Step 11: Calculate the slack bus generator new real power. Form the new real power vector P_{pd}^{l} following the equation from step 5. Check the new position of the slack bus generator in the particles using the constraint Equation (4.22) as follows:

$$P_{pd}^{new^l} = \begin{cases} P_{pd}^{\min}, P_{pd}^{new^l} \le P_{pd}^{\min} \\ P_{pd}^{\max}, P_{pd}^{new^l} \ge P_{pd}^{\max} \\ P_{pd}^{new}, \quad P_{pd}^{\min} \le P_{pd}^{new^l} \le P_{pd}^{\max} \end{cases} \qquad (4.21)$$

Step 12: The full active power vector for the l^{th} iteration is:

$$P_{p}^{new^l} = \left[P_{pd}^{new^l},\ P_{pi}^{new^l}, i = \overline{1, n},\ i \ne d \right],\ p = \overline{1, N_p} \qquad (4.22)$$

Step 13: Calculate the new objective functions F_T^{new} using Step 6

Step 14: Check the new objective function F_T^{new} as described below

$$If\ \ F_T^{new^l} < F_T^{best^{l-1}}\ then\ F_T^{best^l} = F_T^{new^l}\ and\ P_{pi}^{best^l} = P_{pi}^{new^l}$$

$$else\ \ F_T^{best^l} = F_T^{best^{l-1}}\ and\ \ P_{pi}^{best^l} = P_{pi}^{best^{l-1}} \qquad (4.23)$$

$$G^{best^l} = P_p^{best^l},\ p = \overline{1, N_p}$$

where l is the number of iterations

The best solution G^{best} is only one for the whole system.

The best solution per particle is only one $P_p^{best} = min\, P_{pi},\ i = \overline{1, n}$

The best solution for the whole system is $P_p^{best} = \left[P_1^{best},\ P_2^{best},\ ....,P_{N_p}^{best} \right]$. Then

$$G^{best} = min\ P_p^{best}\ for\ p = \overline{1, N_p}$$

Step 15: Repeat the steps 5, 8 to 13 until the maximum number of iterations is reached.

Flowchart of the PSO algorithm is shown in Figure 4.1.

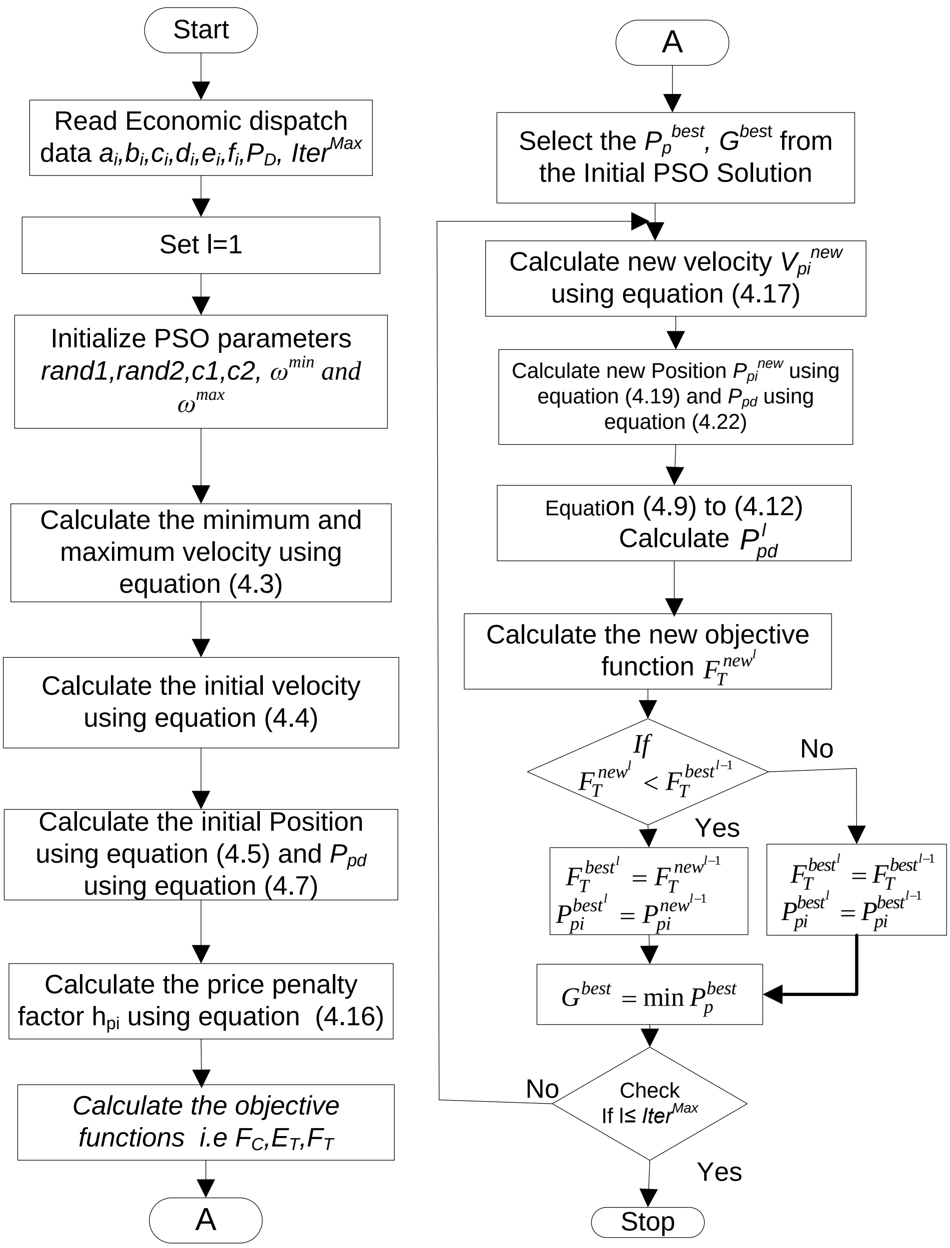


Figure 4.1: **Flowchart of the application of PSO algorithm for CEED problem solution**

The developed PSO software *PSO_Casestudy1.m*, and *PSO_Casestudy2.m*, for the CEED problem is given in the *Appendix D*.

4.4 Application of the developed PSO algorithm for solution of the CEED problem

PSO algorithm is used to solve the CEED problem for the IEEE 30 bus system and the 11 generator systems, given in sections 4.4.1 and 4.4.2 respectively.

4.4.1 Test system 1: IEEE 30 bus system with 6 generators

PSO algorithm described in section 4.3 is applied for the solution of the CEED problem for IEEE 30 bus system and the data of fuel cost, emission coefficients and generator limits are given in Table 3.1, Chapter 3 (Gnanadass, 2005). The various power demands (P_D) of 125 to 250 [MW] are considered. The calculations are carried out in the Matlab software environment. Most of the reference papers used Max-Max penalty factor to solve the CEED problem, but in addition to that Min-Max penalty factor as proposed in Chapter 3 of this thesis is considered. The problem solution results in Chapter 3 conclude that the CEED problem based on quadratic criterion functions are less using Min-Max penalty factor in comparison to the Max-Max one. The optimal solution of the CEED problem using PSO algorithm is given in Tables 4.1 and 4.2 using Min-Max and Max-Max Penalty factors respectively. The initial parameters of the PSO algorithm are given below:

$\omega_{\min} = 0.4$	Minimum weight
$\omega_{\max} = 0.9$	Maximum weight
$\omega = 0.65$	Weight
c1 = c2=2.0	Acceleration factors
m=25	Total number of iterations
$N_p = 10$	Total number of particles in the swarm
$N_p(n-1)$	Total number of velocities
CT	Overall computation time
CT_B	Best solution for the computation time
n=*G*	Total number of generators in every particles

Table 4.1: **Application of the developed PSO algorithm for the CEED problem solution using Min-Max Penalty factor results**

P_D [MW]	P_1 [MW]	P_2 [MW]	P_3 [MW]	P_4 [MW]	P_5 [MW]	P_6 [MW]	PL [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]	Number of Iterations	CT_B [s]	*CT* *[s]*
125	20.108	39.096	18.742	14.928	20.129	12.755	0.758	340.820	184.642	424.808	7	0.089	0.099
150	58.119	26.012	25.828	11.972	10.569	19.031	1.530	394.535	167.327	472.744	17	0.086	0.116
175	81.214	31.346	18.909	16.841	10.122	19.061	2.492	456.759	184.227	541.573	13	0.062	0.211
200	98.774	44.495	15.464	10.040	16.298	18.679	3.750	530.823	219.323	627.296	3	0.071	0.124
225	115.291	42.865	20.665	18.945	15.058	16.835	4.659	607.947	252.196	716.926	24	0.088	0.118
250	141.069	52.575	19.686	16.559	12.650	14.197	6.736	686.735	317.206	816.306	17	0.115	0.123

Table 4.2: **Application of the developed PSO algorithm for the CEED problem solution using Max-Max Penalty factor results**

P_D [MW]	P_1 [MW]	P_2 [MW]	P_3 [MW]	P_4 [MW]	P_5 [MW]	P_6 [MW]	P_L [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]	Number of Iterations	CT_B [s]	*CT* *[s]*
125	34.548	28.323	17.597	19.562	10.888	14.930	0.848	326.883	162.548	674.768	18	0.092	0.094
150	53.370	27.863	16.857	16.262	22.094	14.932	1.377	398.294	168.457	760.436	25	0.106	0.141
175	79.420	25.624	20.761	16.571	14.631	20.298	2.305	463.109	182.529	851.694	12	0.088	0.091
200	98.044	34.517	21.340	16.868	14.053	18.586	3.408	531.877	212.130	972.500	12	0.081	0.109
225	107.548	44.062	18.907	26.812	19.044	12.895	4.268	612.981	250.298	1123.217	18	0.078	0.127
250	124.637	47.712	24.081	25.065	16.516	17.529	5.539	694.519	291.976	1282.514	24	0.108	0.113

The CEED problem solution using the developed Lagrange's method (Table 3.2 and 3.3) and the developed PSO algorithm (Table 4.1 and 4.2) are compared with the solution of the same problem described in (Gnanadass, 2005). Gnanadass used Evolutionary Programming Combined Economic Emission Dispatch (EPCEED) method, on the bases of 283.4 [MW] power demand. The comparison of the CEED problem solution of the developed Lagrange's and PSO methods, based on Min-Max and Max-Max penalty factors are given in Table 4.3 on the bases of 250 [MW] power demand.

Table 4.3: Comparison of Lagrange's and PSO algorithm solutions with the Evolutionary Programming algorithm (Gnanadoss, 2005) one

P_D [MW]	250				283.4		
Penalty factor h_i	Min-Max		Max-Max		Max-Max		
Algorithm	Lagrange's	PSO	Lagrange's	PSO	EPCEED (Gnanadass, 2005)	Lagrange's	PSO
P_L [MW]	6.4337	6.736	5.2153	5.539	5.183	6.511	6.431
F_C [$/h]	687.0737	686.735	699.5472	694.519	866.653	821.928	823.425
E_T [kg/h]	310.2842	317.206	285.3648	291.976	353.331	344.427	344.632
F_T [$/h]	815.1852	816.306	1279.1	1282.514	3288.689	1516.348	1518.549
Iteration Number	117	17	238	24	----------	222	11
CT_B [s]	-----------	0.115	----------	0.108	---------	------	0.085
CT [s]	0.1971	0.123	0.332	0.113	52.01	0.328	0.093

4.4.2 Discussion on the results for Test system 1 using PSO method

Lagrange's algorithm using Min-Max price penalty factor produces minimum fuel cost (F_C, F_T) and computation time (CT) values in comparison with PSO, Evolutionary Programming (EP) and when using the Max-Max penalty factor, as given in Table 4.3. The transmission power loss (P_L) value is less in Evolutionary Programming method (Gnanadoss, 2005) in comparison with the developed Lagrange's and PSO algorithms. But the obtained solutions are not same in Lagrange's and PSO methods, since meta-heuristic (PSO) generate random solution for each execution.

4.4.3 Test system 2: Eleven generator system using PSO method

PSO algorithm described in section 4.3 is applied for the solution of the CEED problem for the eleven generator system. The data of fuel cost, emission coefficients and generator limits are given in Table 3.14 (Balamurugan and Subramanian, 2007). The various power demands (P_D) of 1000 to 2500 [MW] are considered. Max-Max penalty factor is used. The calculations are carried out in the Matlab software environment and is given in the m-file PSO_Casestudy2.m (Appendix D2). The optimal solution of the eleven generator CEED problem using PSO algorithm is given in Table 4.4 and 4.5 respectively. The PSO and Lagrange's fuel cost and emission values of eleven generator system are compared with Dynamic Programming algorithm (Balamurugan and Subramanian, 2007) as given in Table 4.6

Table 4.4: The PSO solution of the generator scheduling for the eleven generators system for various power demands

P_D [MW]	1000	1250	1500	1750	2000	2250	2500
P1	34.234	68.231	46.493	104.763	134.801	115.274	167.992
P2	64.568	43.707	99.706	66.394	98.348	140.762	113.784
P3	62.855	128.528	160.565	66.310	73.924	113.257	102.949
P4	61.676	124.121	103.196	123.713	140.405	179.777	174.001
P5	30.842	82.206	66.902	140.161	135.267	117.849	179.351
P6	70.533	116.715	272.948	132.167	180.032	191.523	214.970
P7	87.797	98.651	46.449	94.274	179.669	135.837	203.323
P8	111.163	108.732	200.998	313.085	184.445	167.015	321.945
P9	105.738	126.921	183.050	180.492	228.336	354.663	332.749
P10	178.633	138.743	203.171	303.631	271.409	406.501	416.014
P11	192.002	213.450	116.521	225.010	373.365	327.543	272.921

Table 4.5: **Application of the developed PSO algorithm for solution of the CEED problem for the eleven generator system under various power demands**

P_D [MW]	1000	1250	1500	1750	2000	2250	2500
F_C [$/h]	8554.249	9245.657	9807.317	10330.121	11056.361	11780.894	12453.618
E_T [kg/h]	235.475	398.440	773.290	944.120	1313.194	1687.337	2054.878
F_T [$/h]	9345.304	10850.956	12425.797	13464.088	15362.523	17282.223	19417.842
Number of Iterations	4	6	20	5	2	7	11
CT_B [s]	0.114	0.109	0.228	0.181	0.115	0.106	0.131
CT [s]	2.905	0.177	0.265	0.222	0.136	0.199	0.192

Table 4.6: **Comparison of the results from the solution of the CEED problem using PSO and Lagrange's algorithms for the eleven generator system with the results using the Dynamic Programming algorithm (Balamurugan and Subramanian, 2007)**

P_D [MW]	Fuel cost F_C in [$/h]			Total Emission E_T in [kg/h]		
	(Balamurugan and Subramanian, 2007)	Developed		(Balamurugan and Subramanian, 2007)	Developed	
	Dynamic Programming	Lagrange's	PSO	Dynamic Programming	Lagrange's	PSO
1000	8502.29	8502.2936	8554.249	205.204	205.2039	235.475
1250	9108.38	9108.3741	9245.657	339.870	339.8696	398.440
1500	9733.54	9733.5344	9807.317	540.544	540.5428	773.290
1750	10377.86	10377.7710	10330.121	807.220	807.2229	944.120
2000	11041.08	11041.0833	11056.361	1139.911	1139.9096	1313.194
2250	11723.47	11723.4716	11780.894	1538.600	1538.6032	1687.337
2500	12424.94	12424.9352	12453.618	2003.300	2003.3032	2054.878

4.4.4 Discussion on the results for eleven generator system using PSO method

The developed PSO algorithm is tested for eleven generator system. The solution is compared with the Lagrange's one calculated in Chapter 3, Dynamic programming one (Balamurugan and Subramanian, 2007), and the developed PSO methods for various power demand values. The fuel cost and emission values of the Lagrange's, PSO and Evolutionary Programming solutions are given in Table 4.6. It concludes that fuel cost and emission values are less in Lagrange's method in comparison with the other methods. Nevertheless, the obtained solutions are not same as Lagrange’s, since the meta-heuristic (PSO) generate random solution for each execution.

4.5 Comparison of Lagrange’s and PSO solution to CEED problem

This section of the thesis compares the two used optimization approaches: the classical one-the (Lagrange’s) and heuristic one-the random variable selection approach (PSO) on the basis of solution of the CEED problem with respect to the obtained solution and the computation time. The application of the Lagrange’s and PSO methods to the CEED problem is described and the algorithms for calculations are given in 3.2.4 and 4.3 respectively. The computational time of the Lagrange’s and PSO algorithms depends on the selection of the initial values of the Lagrange’s variable (λ), and on the initial particle positions and velocity selections in the PSO algorithm. The results are given in Tables 4.7 to 4.9. The IEEE 30 bus system is considered to validate the solution results in MATLAB software environment. It concludes that Lagrange’s algorithm provides better results for the CEED problem in comparison to the PSO algorithm. Graphical presentation of the results is given in Figures 4.2 to 4.6. The best solution of the CEED problem using PSO algorithm is given in Tables 4.8 and Table 4.9 respectively.

4.6 Discussion on the solution of the PSO methods

The PSO algorithm results show that the computational time depends on the selection of number of swarms of the particles. The selection of the smaller number of swarms will narrow the solution search space and will require smaller computational time to find the near optimal solution. In vice-versa when the solution search space is enhanced and computational time required to find near to the optimal solution is bigger. In section 4.5 Lagrange’s algorithm is used with various initial selections of the Lagrange’s variable. If the initial value of the Lagrangian variable is close to its optimal value, then the computational time required to find the optimal solution is small. In vice-versa when the solution space is enhanced and computational time required to find the optimal solution is bigger. The obtained optimal solution and computational time depend on the stopping criteria of Lagrange’s and PSO algorithms, which are tolerance error value and maximum number of iterations respectively.

Table 4.7: **Different selections of the initial Lagrangian variables (lambda) for the CEED problem solution using the Lagrange's method**

P_D [MW]	Optimal λ	P_1 [MW]	P_2 [MW]	P_3 [MW]	P_4 [MW]	P_5 [MW]	P_6 [MW]	P_L [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]	$\lambda^0=4$		$\lambda^0=10$	
												CT [s]	*Iter*	*CT [s]*	*Iter*
150	3.0287	78.8314	26.2653	15.0000	10.0000	10.0000	12.0000	2.0967	373.5001	162.2299	448.5151	0.2174	153	0.2741	180
200	3.7044	114.0699	39.3656	18.8704	10.0000	10.0000	12.0000	4.3059	519.5033	228.1548	616.9529	0.1171	129	0.1929	166
250	4.1980	138.9676	48.7366	22.3455	18.8091	13.7142	13.8596	6.4336	687.0699	310.2823	815.1807	0.0751	85	0.1301	129

The Maximum number of iteration is set to 25, initial selection of the velocity and particles in the swarm are 5 and 10 respectively for the solution of the PSO problem, Table 4.8

Table 4.8: **Application of PSO algorithm for the CEED problem solution using the Min-Max penalty factor for 10 particles in the swarm**

P_D [MW]	P_1 [MW]	P_2 [MW]	P_3 [MW]	P_4 [MW]	P_5 [MW]	P_6 [MW]	PL [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]	Number of Iterations	CT_B [s]	*CT [s]*
150	53.3617	33.3541	26.2260	15.6713	10.4352	12.4333	1.4815	392.6255	171.6735	471.1946	20	2.0074	2.5356
200	116.1479	29.2151	17.9788	11.2252	11.6081	17.9721	4.1471	524.8455	224.7032	623.3757	22	2.2682	2.8656
250	143.1200	47.6339	19.0569	18.7279	15.9568	12.1417	6.6372	686.0111	315.6620	816.5148	20	2.8225	2.9588

In Table 4.1 and 4.8 reports different results for the application of the same algorithm to the same system, since the Meta-heuristic PSO algorithm generates random solution for every simulation run.

The Maximum number of iteration is 25, initial velocity is 5, and 30 particles are considered in the swarm for the solution of the PSO problem, Table 4.9

Table 4.9: Application of PSO algorithm for the CEED problem solution using the Min-Max penalty factor for 30 particles in the swarm

P_D [MW]	P_1 [MW]	P_2 [MW]	P_3 [MW]	P_4 [MW]	P_5 [MW]	P_6 [MW]	PL [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]	Number of Iterations	CT_B [s]	*CT [s]*
150	63.9408	23.6347	27.7912	11.5311	10.3260	14.4160	1.6399	391.0849	165.8066	467.7885	4	2.0069	12.4357
200	102.4183	29.8400	24.4926	18.6308	14.0595	14.0026	3.4438	531.4352	214.1952	627.0118	7	2.8157	11.0558
250	143.0733	48.4094	20.2477	16.1364	11.1358	17.7469	6.7494	686.7437	316.3996	816.8095	2	0.8674	11.7269

Figures 4.2 – 4.5, show the comparison of Lagrange's and PSO solutions: Computation Time (CT), CEED fuel cost (F_T), Transmission power loss (P_L) and Emission (E_T) values for the power demands (P_D) of [150, 200, 250] MW's respectively are compared.

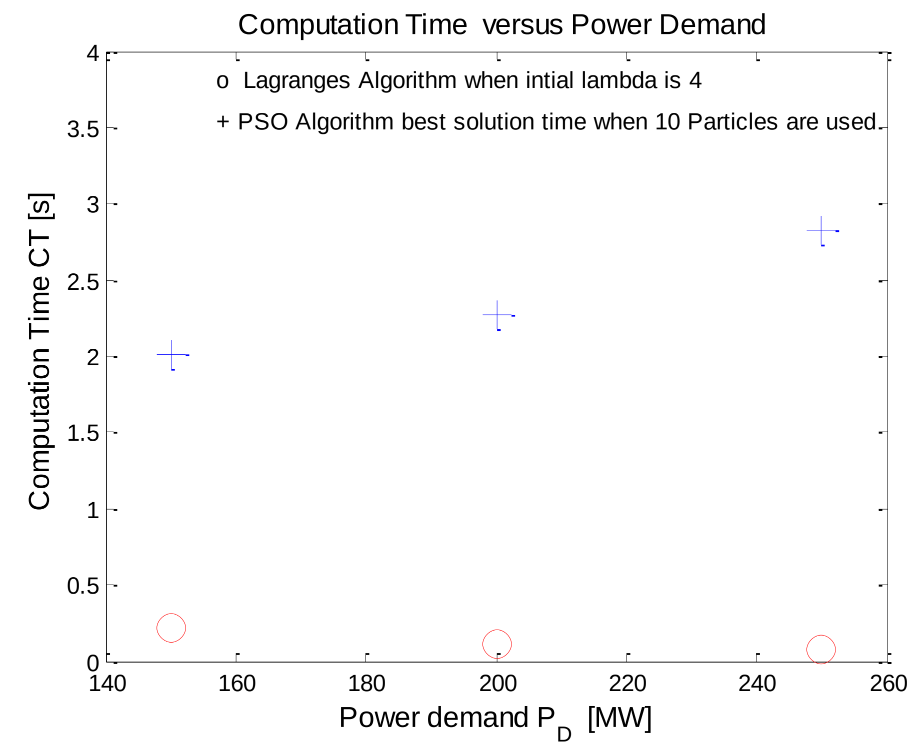


Figure 4.2: **Computation time of Lagrange's and PSO algorithm implementation for different cases of power demand**

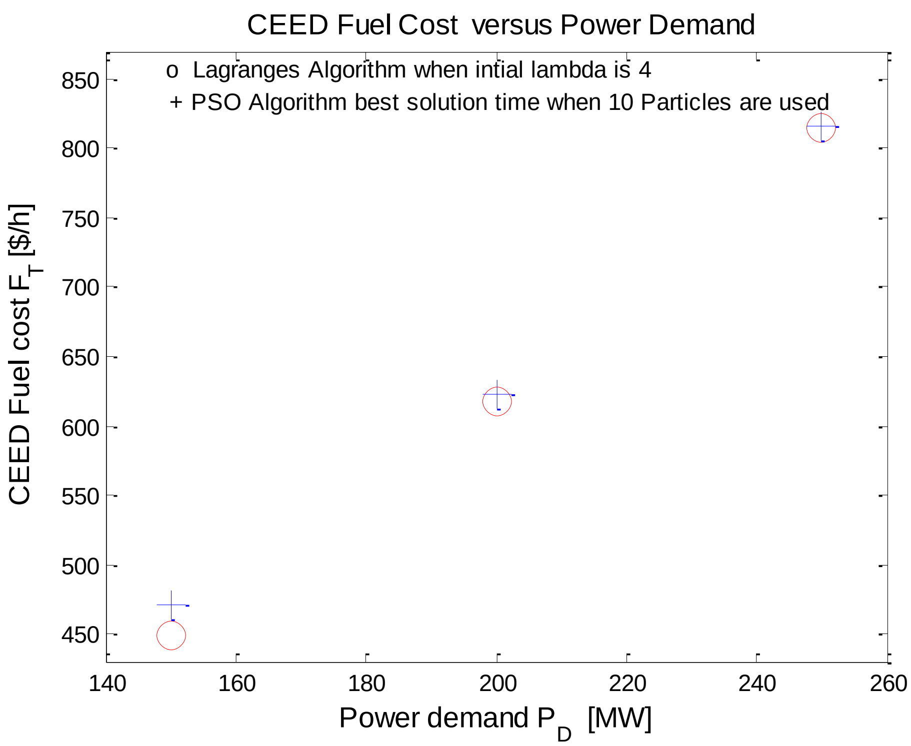


Figure 4.3: **CEED fuel cost of Lagrange's and PSO algorithm implementation for different cases of power demand**

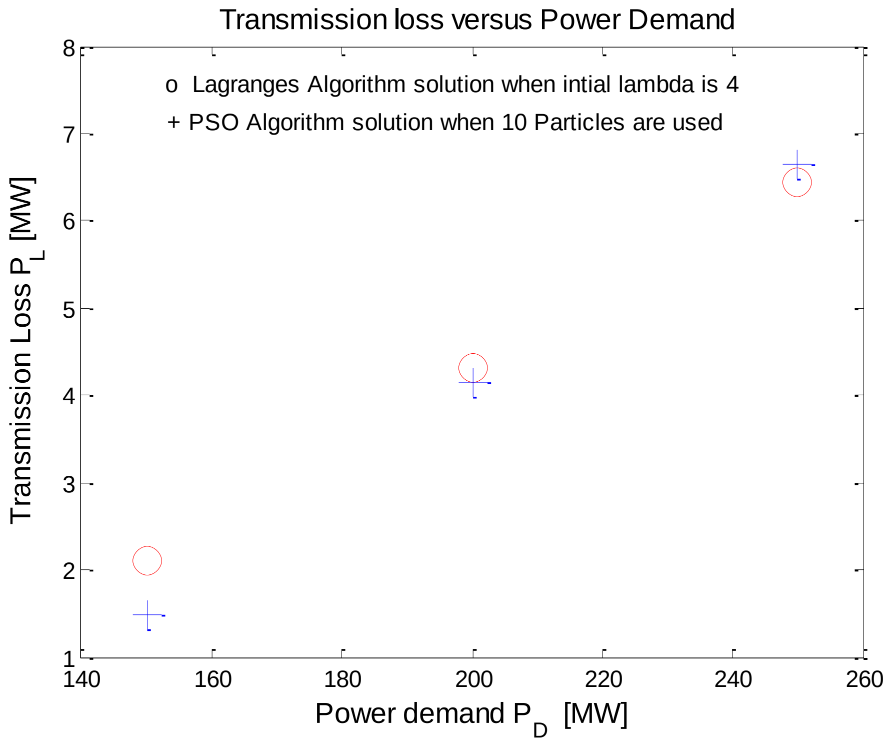


Figure 4.4: **Transmission loss obtained by using the Lagrange's and PSO algorithms for different cases of the power demand**

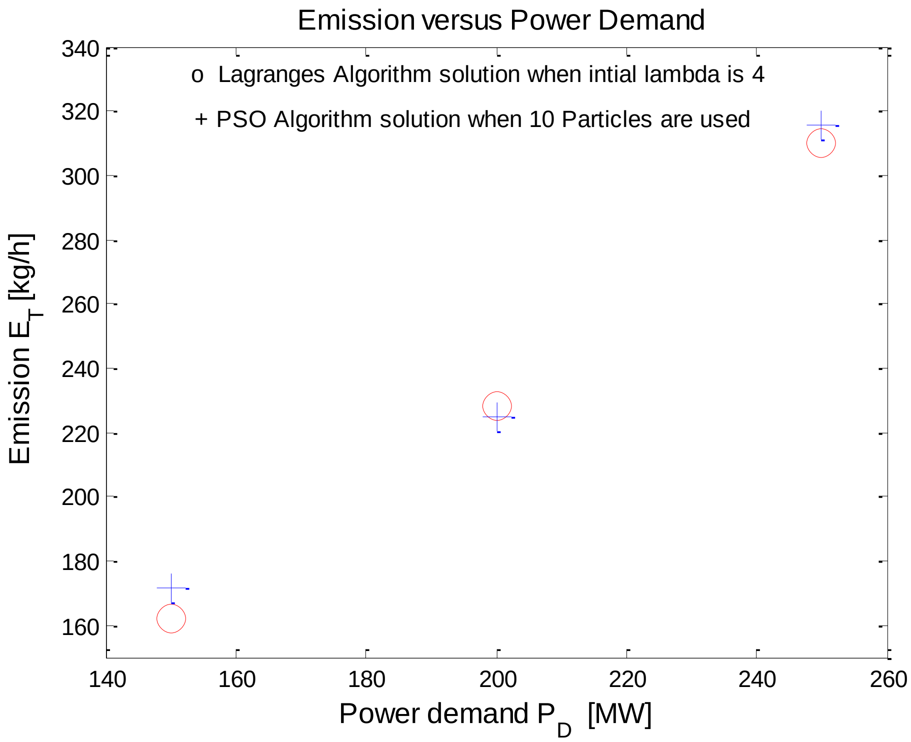


Figure 4.5: **Emission values obtained by using the Lagrange's and PSO algorithms for different cases of the power demand**

The results of IEEE 30 bus system show that Lagrange's algorithm obtained optimal solution is with less computational time and less CEED fuel cost value in comparison with the PSO algorithm one. The transmission power loss and emission values are less in PSO algorithm. Finally, it can be concluded that optimization approach (Lagrange's algorithm) provides better solution of the CEED problem in comparison to the random approach (PSO algorithm) as shown in Figures 4.2 - 4.4.

4.7 Conclusion

This chapter developed the PSO algorithm to solve the CEED problem. IEEE 30 bus and eleven bus systems are used to test the developed PSO algorithm in Matlab software. Min-Max price penalty factor is proposed to the CEED problem in addition to the Max-Max one. The developed PSO solution is compared with the Lagrange's method developed in Chapter 3, Evolutionary Programming (Gnanadass, 2005) and Dynamic Programming (Balamurugan and Subramanian, 2007). Lagrange's algorithm provides less CEED fuel cost and emission values in comparison with the other methods and the results are given in Table 4.3 and 4.6 respectively. The Lagrange's and PSO solutions depend on the initial selection of the Lagrangian variable (lambda) and on the number of particles in the swarm as given in Table 4.7 - 4.8. It can be concluded that Lagrange's algorithm provides global solution irrespective of the selection of initial lambda, and the PSO algorithm provides near global solution irrespective of the selection of the number of particles in the swarm.

On the basis of the results described in Chapters 3 and 4, it can be seen that a lot of calculations and simulations are necessary to be done in order fully investigate the proposed algorithms and their implementation. One of the ways of reduction of this calculation burden is to introduce parallel computation.

Short analysis of the capabilities of the parallel computing for multi-core processors and Cluster of Computers is described in Chapter 5.

CHAPTER FIVE
PARALLEL COMPUTING AND ITS IMPLEMENTATION

5.1 Introduction

The simultaneous use of more than one processor or computer to solve a problem is called parallel computing. The principle of the parallel computing is to divide the large problems into a smaller ones and the calculations are carried out simultaneously, in parallel. Bit-Level, instruction level, data, and task parallelism are the different forms of parallel computing described by (Gottlieb and Almasi, 1989). Parallel computers are classified according to the hardware support for parallelism. The multi-core and multi-processor computers have multiple processing elements within a single machine, while Clusters, Massively Parallel Processing(MPPs),and grids use multiple computers to work on the same task. The authors (Patterson and Hennessy, 2012) described that serial computing is too slow and need for large amounts of memory not accessible by a single processor. Computations on very large data sets involved: load balancing, routing, data partitioning, locality, and prefetching which drastically improve the system's response time. (Roman and Zinterhof, 2009) used parallel computing technique in various applications such as Monte Carlo simulations, lattice Boltzmann simulations, time dependent Schrodinger equations, molecular dynamic simulations, heat transfer bological tissue problems, quantum Fourier transform, parallel multi-objective evolutionary algorithms and optimization problems. This chapter describes the parallel computing principles and their implementation is carried out in chapter 6 to solve the CEED problem in data-parallel way and Multi Area Economic Dispatch (MAED) problem. MAED in Chapter 7. To schedule the generating units within each area and for the whole interconnected power system by satisfying required constraints including tie-lines. Decomposition is the process by which a complex problem or system is broken down into parts that are easier to understand and program. To decompose a large interconnected power system dispatch optimisation problem into smaller sub- problems is required that the set of the optimal solutions of the sub-problems is the same as the one obtained from the solution of the overall problem. Such decomposition methods are necessary for power system problems where it is often not possible to formulate and solve a single optimisation problem for the entire system. Lagrange's algorithm is developed for the multi-area economic emission dispatch problem in Chapter 7 using Parallel Computing Toolbox (PCT) and MATLAB Distributed Computing Server (MDCS) software to solve computationally and data-intensive multi-area dispatch problem in multi-core and multiprocessor computers (Matlab distributed computing server installation guide, 2011).

This chapter describes the Amdahl's and Gustafson's law in part 5.2, types of parallelism in part 5.3, hardware and memory architecture of a parallel computer in part 5.4 and 5.5 respectively, Matlab parallel computing in part 5.6, load balancing algorithm in part 5.7, and program development in part 5.8. The conclusion is given in part 5.9.

5.2 Parallel computation using Amdahl’s and Gustafson’s law

A program solving a large mathematical or engineering problem will typically consist of several parallelizable parts and several non-parallelizable (sequential) parts. The potential speed-up of an algorithm on a parallel computing platform is given by Amdahl’s law, formulated in (Amdahl,1960). It states that a small portion of the program which cannot be parallelized will limit the overall speed-up available from parallelization.

If α is the fraction of running time a program spends on non-parallelizable parts, then, the maximum speed-up (**s**) with parallelization of the program is given as:

$$s = \frac{1}{\alpha} = \lim_{P \to \infty} \frac{1}{\frac{1-\alpha}{p} + \alpha} \tag{5.1}$$

where

p Number of processors

Gustafson's law is another law in computing, closely related to Amdahl's law. It states that the speedup with ***p*** processors given in (Gustafson, 1988) as:

$$s(p) = p - \alpha\left(p-1\right) = \alpha + p\left(1-\alpha\right) \tag{5.2}$$

Both Amdahl's law and Gustafson's law assume that the running time of the sequential portion of the program is independent of the number of processors. Amdahl's law assumes that the entire problem is of fixed size so that the total amount of work to be done in parallel is also independent of the number of processors, whereas Gustafson's law assumes that the total amount of work to be done in parallel varies linearly with the number of the processors.

5.2.1 History of Parallel MATLAB

Cleve Moler, the original Matlab author and the co-founder of Mathworks worked on a version of Matlab for both an Intel hypercube and Ardent Titan in the late 1980s(Moler, 1995). Moler's 1995 article "Why there isn't a parallel Matlab"(Moler, 1995) described the three major obstacles in developing a parallel Matlab language: memory model, granularity of computations, and business situation. The conflict between Matlab's global memory model and the distributed model of most parallel systems meant that the large data matrices had to be sent back and forth between the host and the parallel computer. Also at the time Matlab spent only a fraction of its time in parallelizable routines compared to parser and graphics routines which made a parallelization effort not very attractive. The last obstacle was simply a dose of reality for an organization with finite resources—there were simply not enough Matlab users that wanted to use Matlab on parallel computers, and the focus instead was on improving the uniprocessor Matlab.

Several factors have made the parallel Matlab project a very important one inside Mathworks: Matlab has matured into a preeminent technical computing environment supporting large scale projects, easy access to multiprocessor machines, and demand for a full-fledged solution from the user community. There have been three approaches to creating a system for parallel computing with Matlab. The first approach aims at translating Matlab or similar programs into a lower-level language such as C or Fortran, and uses annotations and other mechanisms to generate parallel code from a compiler. Examples of such projects include Conlab(Jacobson et al, 1992), and Falcon(DeRose et al, 1995). Translating regular Matlab code to C or Fortran is a difficult problem. In fact, the Matlab Compiler software from the Mathworks switched from producing C code to producing wrappers around Matlab code and libraries to be able to support all the language features. The second approach is to use Matlab as a "browser" for parallel computations on a parallel computer while Matlab itself remains unmodified and the Matlab environment does not run natively on the parallel computer. This approach does not really classify as a "parallel Matlab" solution any more than a Web browser used to access a portal for launching parallel applications is itself a parallel application.

The third approach is to extend Matlab through libraries or by modifying the language itself. Recently, the MatlabMPI and pMatlab projects at MIT Lincoln Laboratory and the Multi Matlab project at Cornell University (with which The Math-works was also involved) are among the more successful and widely used libraries for parallel computing with Matlab. Other projects include ParaM and GAMMA projects(Panuganti et al, 2006), Parallel Toolbox for Matlab(Hollingsworth et al, 1996) (which uses PVM for message passing), and various MPI toolbox implementations for Matlab, including the most recent bcMPI (Blue Collar MPI) from Ohio Supercomputer center. The MathWorks introduced Parallel Computing Toolbox software and Matlab Distributed Computing Server in November 2004, (originally named Distributed Computing Toolbox™ and Matlab Distributed Computing Engine™, respectively). These fall into the last category of solutions. When started to expand the capabilities of Matlab into parallel computing, they decided to target embarrassingly parallel problems, given that their initial survey showed a large number of the users wanted to simplify the process of running Monte Carlo or parameter sweep simulations on their groups computers.

Implicit and explicit multithreading of computations are other methods to parallelize Matlab computations on a single multicore or multiprocessor machine and will be discussed later in the chapter.

5.3 Types of parallelism

There are different types of parallelism and they are as follows:

5.3.1 Bit-level parallelism

The amount of information the processor can manipulate per cycle is called Bit-level parallelism. An 8-bit processor requires two instructions to complete a single operation, where a 16-bit processor would be able to complete the operation with a single instruction. Historically, 4-bit microprocessors were replaced with 8-bit, then 16-bit, then 32-bit, and then recently with 64-bit microprocessors.

5.3.2 Instruction-level parallelism

A stream of instructions executed by a processor (Patterson and Hennessy, 2012). These instructions can be re-ordered and combined into groups which are then executed in parallel without changing the result of the program. This is known as instruction-level parallelism. A processor with an N-stage pipeline can have up to N different instructions at different stages of completion.

5.3.4 Data parallelism

Data parallelism is parallelism inherent in program loops, which focuses on distributing the data across different computing nodes to be processed in parallel.

Many scientific and engineering applications exhibit data parallelism and is given in (Patterson and Hennessy, 2012).

5.3.5 Task parallelism

Task parallelism is the characteristic of a parallel program that entirely different calculations can be performed on either the same or different sets of data (Culler and Singh, 1997). This contrasts with data parallelism, where the same calculation is performed on the same or different sets of data. Task parallelism does not usually scale with the size of a problem.

5.4 Hardware Architecture of a Parallel Computer

The core element of parallel processing are the CPUs. The essential computing process is the execution of sequence of instruction on an asset of data. The term stream is used here to denote a sequence of items as executed by a single processor or a multiprocessor. According to (John, 2000), (Behrooz, 2002), and (Changhun, 2002) the number of instruction and data streams can be processed simultaneously and hence the parallel computer system can be classified as:

5.4.1 Single Instruction Single Data (SISD)

The single processing element executes instructions sequentially on a single data stream. The operations are thus ordered in time and may be easily traced from start to finish. Figure 5.1 shows the SISD processor, where IS is the Instruction Stream, DS is the Data Stream, CU is the Control Unit, PU is the Processing Unit and MM is the Memory Module

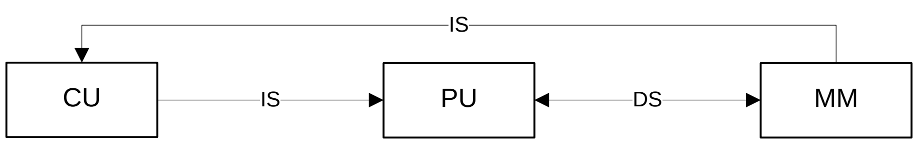


Figure 5.1: **SISD processor (John, 2000)**

5.4.2 Single Instruction Multiple Data (SIMD)

Machines apply a single instruction to a group of data items simultaneously. A master instruction is thus acting over a vector of related operands. Figure 5.2 shows the SIMD processor

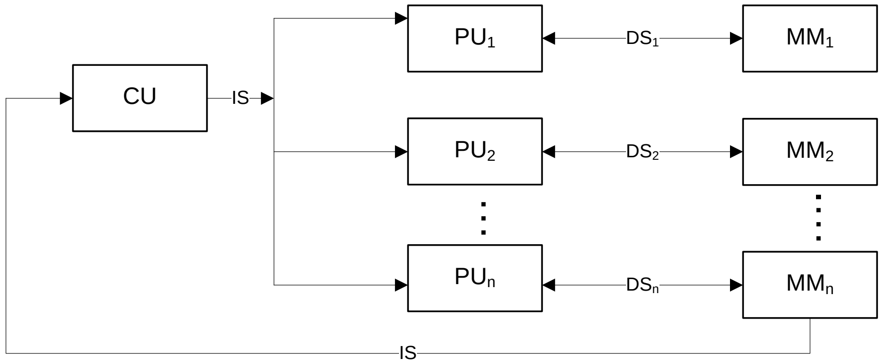


Figure 5.2: **SIMD processor (Behrooz, 2002)**

5.4.3 Multiple Instruction Single Data (MISD)

In this system, there are ***n*** processor units, each receiving distinct instructions for operating over the same data stream. The result of one processor becomes the input of the next processor and is shown in Figure 5.3

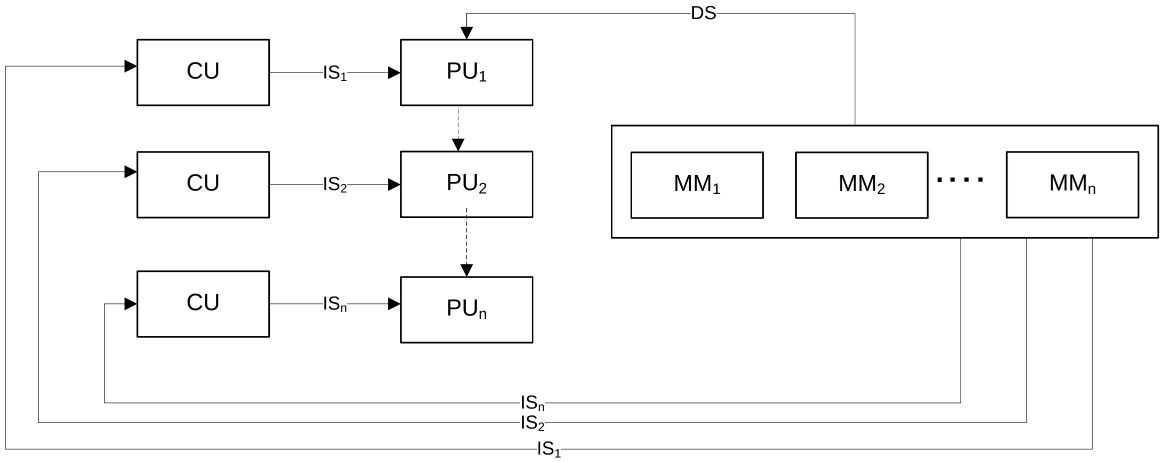


Figure 5.3: **MISD processor (Changhun, 2002)**

5.4.4 Multiple Instruction Multiple Data (MIMD)

MIMD systems provide a separate set of instructions for each processor and is shown in Figure 5.4. This allows the processors to work on different parts of a problem asynchronously and independently.

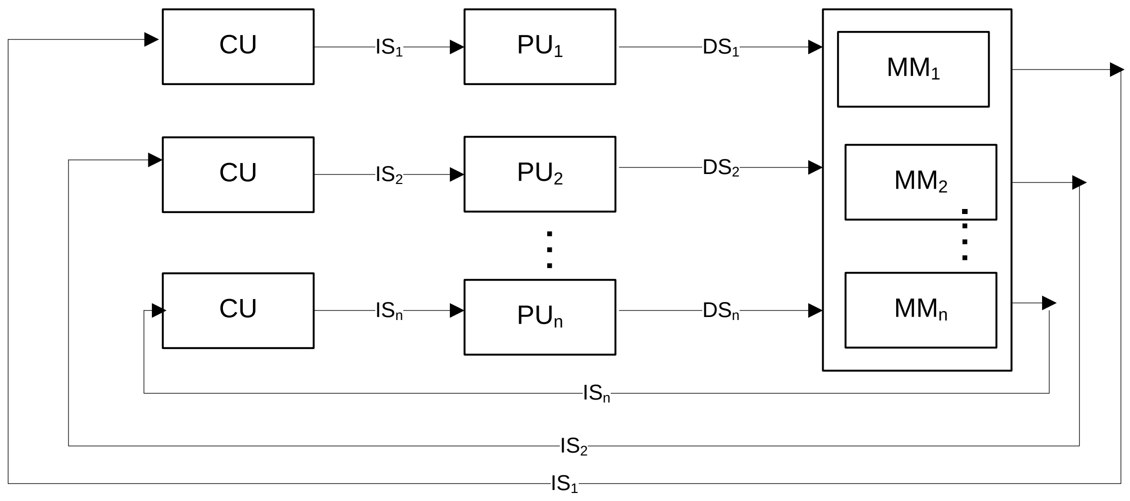


Figure 5.4: **MIMD processor (Changhun, 2002)**

5.5 Memory Architecture of a Parallel Computer

The primary memory architectures are: (Culler et al.,1999)

- Shared memory
- Distributed memory
- Hybrid Distributed-Shared memory

5.5.1 Shared memory

In shared memory architecture, multiple processors operate independently but share the same memory resources shown in Figure 5.5, where PU is the Processing Unit. Only one processor can access the shared memory location at a time. Shared memory machines can be divided into two main classes based upon memory access times: Uniform Memory Access (UMA) and Non-Uniform Memory Access (NUMA).

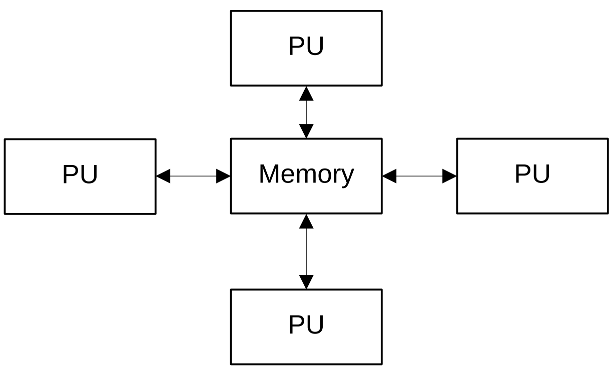


Figure 5.5: **Shared memory architecture (Culler et al.,1999)**

5.5.2 Distributed memory

Processors have their own local memory. Memory addresses in one processor do not map to another processor, so there is no concept of global address space across all processors. Because each processor has its own local memory, it operates independently as shown in Figure 5.6

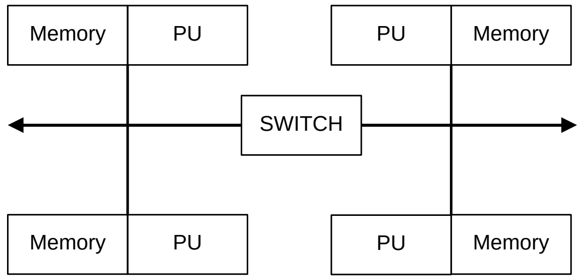


Figure 5.6: **Distributed memory architecture (Culler et al.,1999)**

5.5.3 Hybrid Distributed-shared memory

The largest and fastest computers in the world today employ both shared and distributed memory architectures. The shared memory component is usually a cache coherent Simple Management Protocol (SMP) machine. Processors on a given SMP can address that machine's memory as global. The distributed memory component is the networking of multiple SMPs. SMPs know only about their own memory - not the memory on another SMP. Therefore, network communications are required to move data from one SMP to another. Figure 5.7 shows the Hybrid distributed-shared memory (Culler et al., 1999).

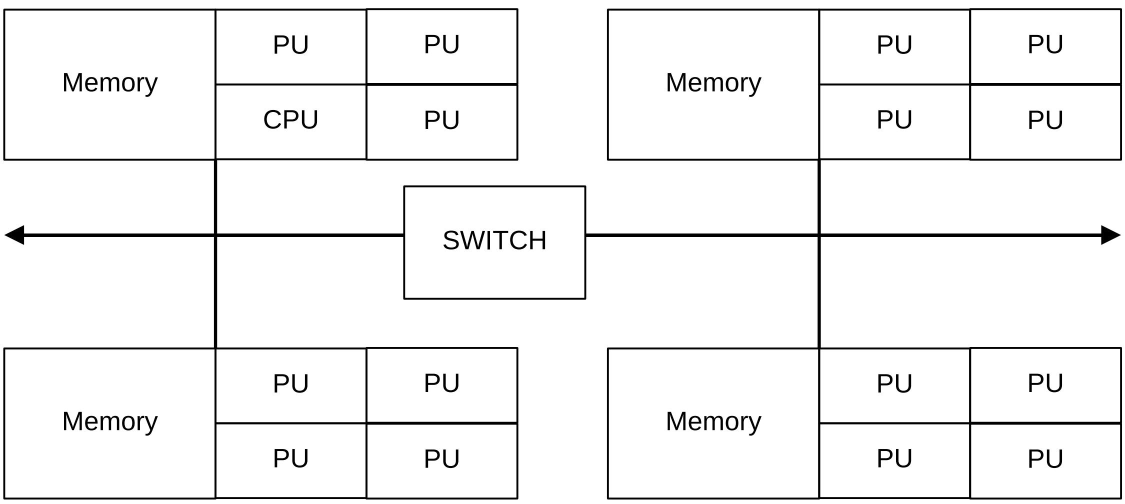


Figure 5.7: **Hybrid distributed-shared memory architecture (Culler et al.,1999)**

5.6 Parallel Computing with Matlab

Parallel Computing Toolbox (PCT) and Matlab Distributed Computing Server (MDCS) solve computationally and data intensive problems using Matlab and Simulink on multicore and multiprocessor computers. Parallel processing constructs such as parallel for-loops, distributed arrays, parallel numerical algorithms and message passing functions. Matlab PCT allows task-parallel and data-parallel algorithms at a high level without programming for specific hardware and network architectures. The basic parallel computing setup is shown in Figure 5.8 (Matlab distributed computing server installation guide, 2011).

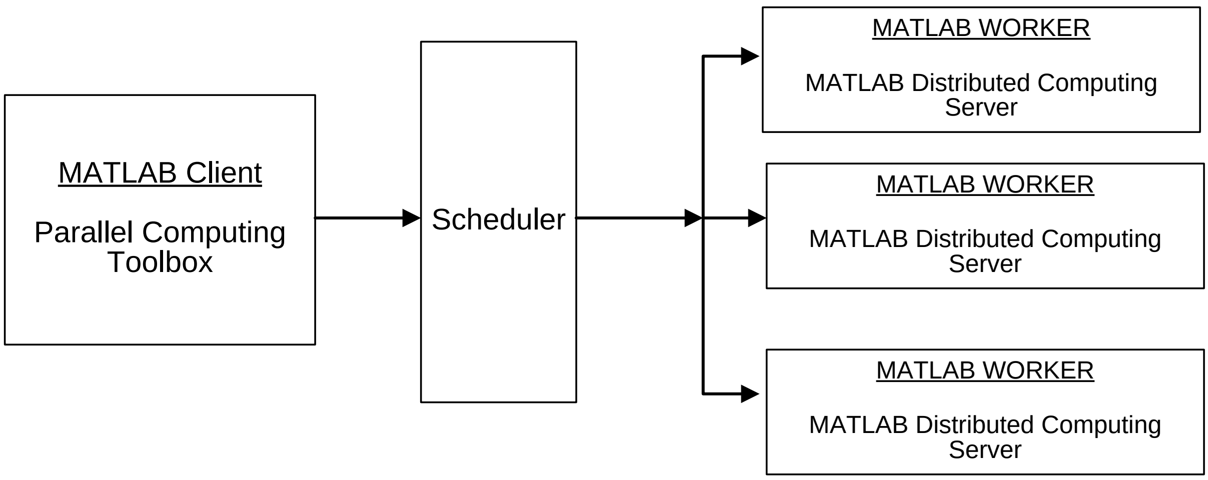


Figure 5.8: **Parallel Computing setup**

5.6.2 Terms used in Matlab parallel computing toolbox

The following definitions are used in Matlab parallel computing toolbox (Matlab distributed computing server installation guide, 2011).

Job: It is a large operation that needs to perform in the user MATLAB session.

Task: A job is broken down into segments called tasks. The user can decide how best to divide his job into tasks. The user can divide the job into identical tasks, but tasks do not have to be identical.

Client session: The MATLAB session in which the job and its tasks are defined is called the client session. Often, this is on the machine where the program in MATLAB is done.

Parallel computing Toolbox: The client uses Parallel Computing Toolbox software to perform the definition of jobs and tasks and to run them on a cluster local to the users machine.

MATLAB Distributed Computing server: It is the Matlab product software that performs the execution of a job on a cluster of machines. A MATLAB Distributed Computing Server software setup usually includes many workers that can all execute tasks simultaneously, speeding up execution of large MATLAB jobs. It is generally not important which worker executes a specific task. In an independent job, the workers evaluate tasks one at a time as available, perhaps simultaneously, perhaps not, returning the results to the Matlab Job Scheduler (MJS). In a communicating job, the workers evaluate tasks simultaneously. The MJS then returns the results of all the tasks in the job to the client session.

MATLAB Job Scheduler (MJS): It is the process that coordinates the execution of jobs and the evaluation of their tasks. The MJS can be run on any machine on the network. The MJS runs jobs in the order in which they are submitted, unless any jobs in its queue are promoted, demoted, canceled, or deleted.

Workers: The MJS distributes the tasks for evaluation to the server's individual MATLAB sessions called workers. Use of the MJS to access a cluster is optional. The distribution of tasks to cluster workers can also be performed by a third-party scheduler, such as Microsoft Windows High Performance Computing (HPC) Server or Platform Load Sharing Facility (LSF). Each worker is given a task from the running job by the MJS, executes the task, returns the result to the MJS, and then is given another task. When all tasks for a running job have been assigned to workers, the MJS starts running the next job on the next available worker.

The interaction of parallel computing sessions with single and multiple MJS is shown in Figures 5.9 and 5.10 respectively.

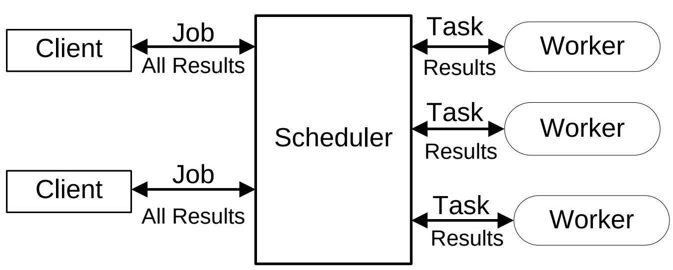


Figure 5.9: **Configuration of MJS and client (Matlab distributed computing server installation guide, 2011)**

A large network might include several MJSs as well as several client sessions. Any client session can create, run, and access jobs on any MJS, but a worker session is registered with and dedicated to only one MJS at a time as shown in Figure 5.10.

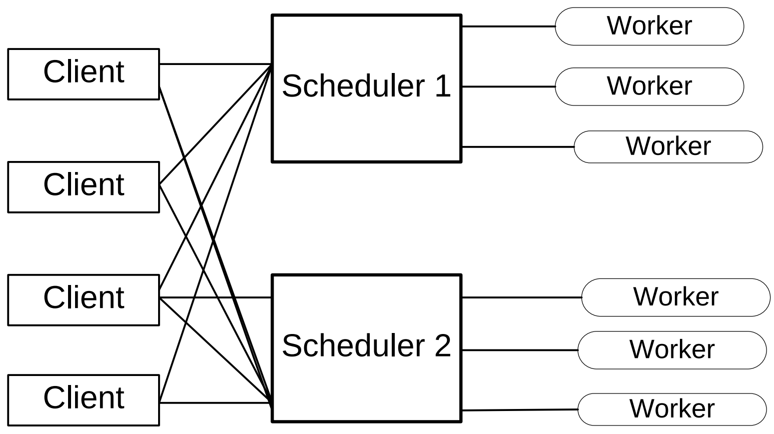


Figure 5.10: Configuration of multiple MJS and client (Matlab distributed computing server installation guide, 2011)

Local Cluster: A feature of Parallel Computing Toolbox software is the ability to run a local scheduler and a cluster of up to twelve workers on the client machine, so that one can run jobs without requiring a remote cluster or MATLAB Distributed Computing Server software. In this case, all the processing required for the client, scheduling, and task evaluation is performed on the same computer. This gives the opportunity to develop, test, and debug the parallel applications before running them on the cluster.

Third-Party Schedulers: As an alternative to using the MJS, one can use a third-party scheduler. This could be a Microsoft Windows HPC Server , Platform LSF scheduler, PBS Pro scheduler, TORQUE scheduler, or a generic scheduler.

MDCE (Matlab Distributed Computing Engine) Service: In MJS every machine that hosts a worker or MJS session must also run the MDCE service. The MDCE service controls the worker and MJS sessions and recovers them when their host machines crash. If a worker or MJS machine crashes, when the MDCE service starts up again, it automatically restarts the MJS and worker sessions to resume their sessions from before the system crash.

Life Cycle of a Job: Creation and running a job, progresses through a number of stages. Each stage of a job is reflected in the value of the job object's state property, which can be pending, queued, running, or finished and is shown in Figure 5.11. Each of these states is briefly described in Table 5.1

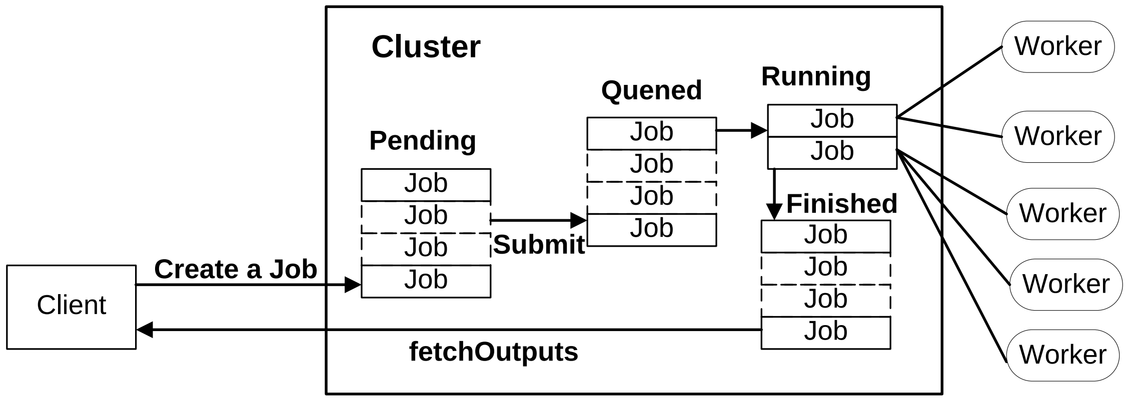


Figure 5.11: States of a Job (Matlab distributed computing server installation guide, 2011)

Table 5.1: States in the life cycle of a job

Job State	Description of each stage of the Job
Pending	A job is created on the scheduler with the *createJob* function in the client session of the Parallel Computing Toolbox (PCT) software. The job's first state is pending. This is when the user defines the job by adding tasks to it.
Queued	To execute the submit function on a job, the MJS or scheduler places the job in the queue, and the job's state is queued. The scheduler executes the jobs in the queue in the sequence in which they are submitted, all jobs moving up the queue as the jobs before them are finished.
Running	When a job reaches the top of the queue, the scheduler distributes the job's tasks to worker sessions for evaluation. The job's state is now running. If more workers are available than are required for a job's tasks, the scheduler begins executing the next job. In this way, there can be more than one job running at a time.
Finished	When all of a job's tasks have been evaluated, the job is moved to the finished state. At this time, one can retrieve the results from all the tasks in the job with the function *fetchOutputs*.
Failed	When using a third-party scheduler, a job might fail if the scheduler encounters an error when attempting to execute its commands or access necessary files.
Deleted	When a job's data has been removed from its data location or from the MJS with the *delete* function, the state of the job in the client is deleted. This state is available only as long as the job object remains in the client.

5.7 Load-balancing algorithm

Load balancing is a computer networking method to distribute workload across multiple computers or a computer cluster, network links, central processing units, disk drives, or other resources, to achieve optimal resource utilization, maximize throughput, minimize response time, and avoid overload. Using multiple components with load balancing, instead of a single component, may increase reliability through redundancy.

Load balancing is very important on designing parallel programs for performance reasons (Shah et al., 2007). In general any load balancing algorithm consists of two basic policies namely a transfer policy and a location policy. Location policy determines the under loaded processor. In other words it locates complementary nodes to/from which a node can send/receive workload to improve the cluster performance. Further, while balancing the load, types of information such number of jobs waiting in queue, job arrival rate, CPU processing rate, at each processor may be exchanged among the processors to improve system performance. Load can be classified as either static, dynamic and adaptive but the thesis uses static scheduling. The principle of load balancing can be presented in the following flowchart, Figure 5.12 (Shah et al, 2007)

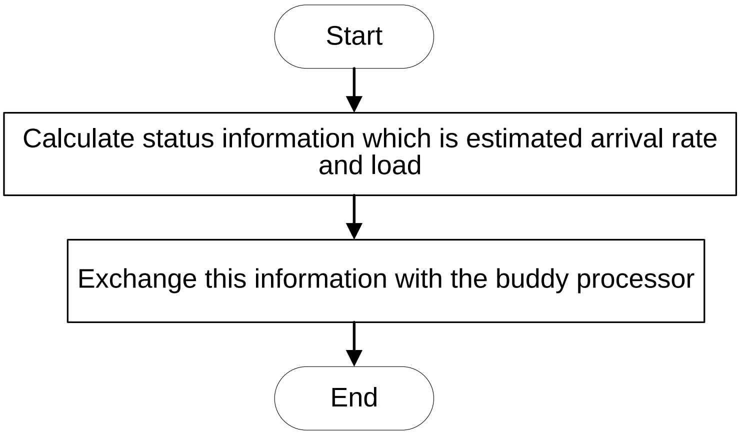


Figure 5.12: Flowchart of the load-balancing algorithm (Shah et al, 2007)

5.8 Program Development

The algorithms described in chapter 3 and chapters 4 are used to develop software for calculation of the optimal dispatch problem in two ways: sequential and parallel.

5.8.1 Sequential in one computer

This algorithm is used for calculation of the optimal generation of the economic dispatch problem. Execution of a program leads to a sequence of calls to functions defined in different m-files in a sequential way, called sequential composition. In sequential programming the program is run on a single computer meaning having single Central processing unit (CPU). The program is discretized into series of instruction. Block diagram for sequential implementation of the program is given in Figure 5.13 (Patterson and Hennessy, 2012)

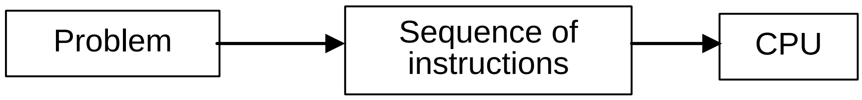


Figure 5.13: Sequential execution of a program (Patterson and Hennessy, 2012)

In a parallel program constructed using only sequential composition, each processor inevitably executes the same program, which in turn performs a series of calls to different program components. These program components may themselves communicate and synchronize, but they cannot create new tasks. Hence, the entire computation moves sequentially from one parallel operation to the next. The software written to solve the optimization problems based on the Lagrange's method is based on the sequential way of programming. The load balancing of the processor is given in Figure 5.14 (Shah et al, 2007).

5.8.2 Parallel computing using the Cluster of Computers

A parallel composition specifies which program components are to execute in which part of the computer and how these components are to exchange data. In principle any program expressed as a parallel composition can be converted to a sequential composition that interleaves the execution of the various program components appropriately. And hence the use of parallel composition can enhance scalability and locality. Parallel programming can be performed on one computer with multi-cores or in a cluster of computers connected on a single network. An example of parallel configuration of a cluster of computers is given on Figure 5.15 (Shah et al., 2007).

The algorithm for optimal generation of economic dispatch problem is programmed in both sequential and parallel way. The goal of parallelization is to obtain high performance and increased speed over the sequential program that solves the same problem. This is to ensure efficient load balancing among processors, reduction in communication overhead and synchronization. The parallel implementation and the results from the calculations are described in chapters 6 and 7.

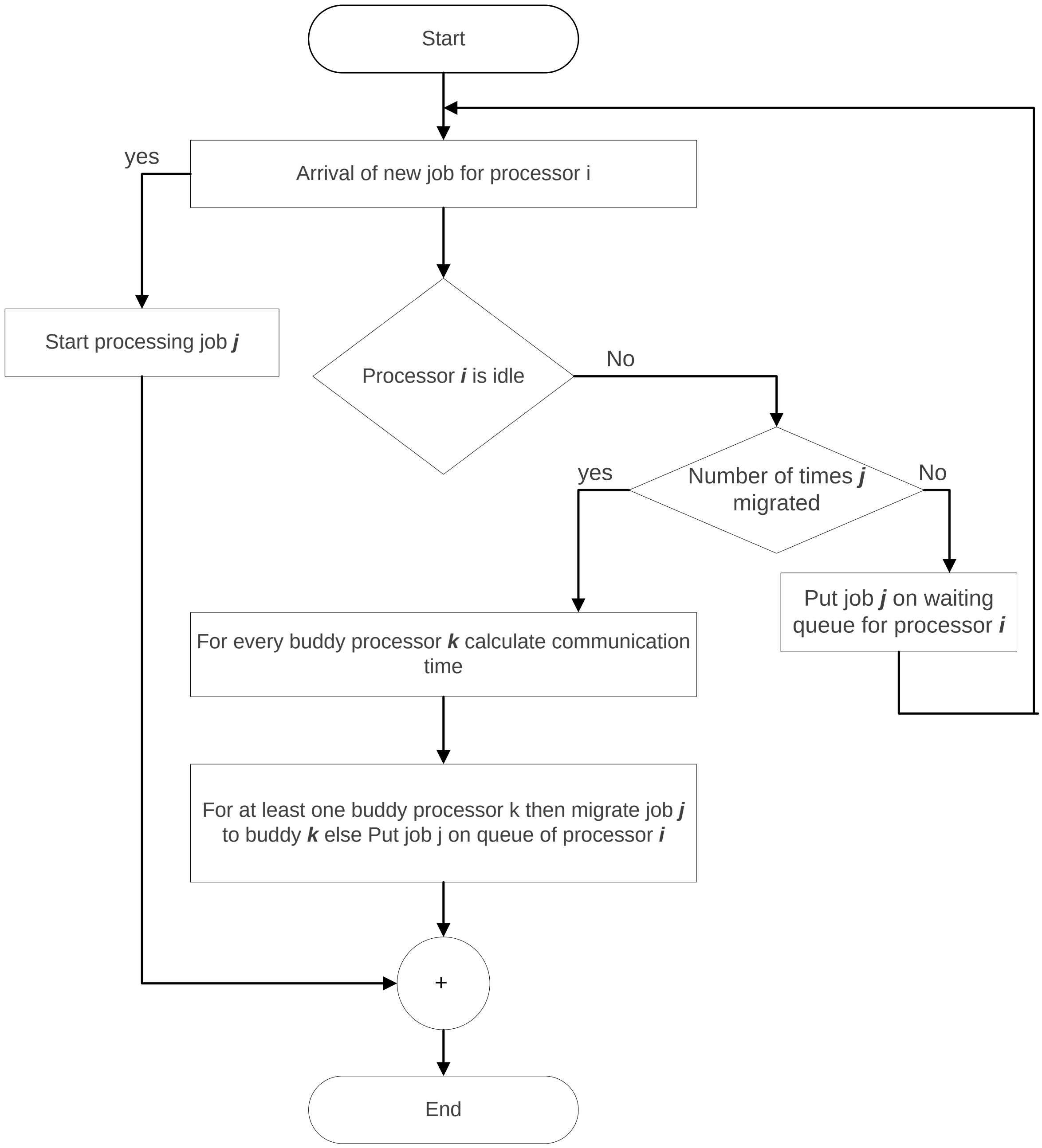


Figure 5.14: Flowchart for load-balancing processing by the processor pi on arrival of job j (adapted from Shah et al, 2007)

Chapter 6 solves the same sequential problem in parallel using variables set of data for the set of Matlab workers. Then many solutions of the same problem are calculated for the various sets of data in a short time.

Chapter 7 solves the complex interconnected problem using parallel solution of their sub-problems.

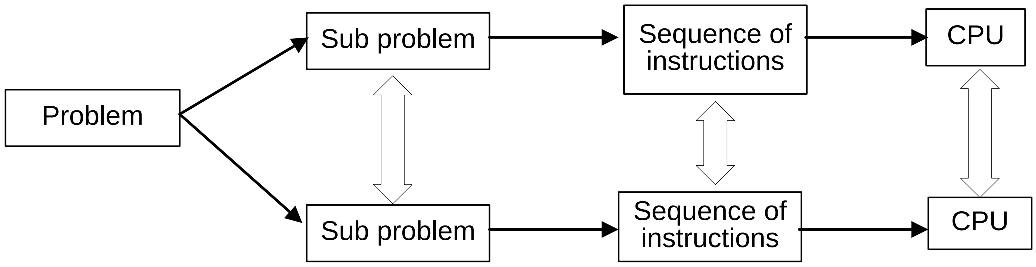


Figure 5.15: Parallel programming block diagram (adapted from Shah et al., 2007)

5.9 Conclusion

A review of the evolution of sequential computers has revealed many ways the parallelism has been introduced into sequential computers over time. While the application of parallelism in sequential machines has been hidden from user program, it has been the key mechanism of delivering speed in computer from the architecture and system design perspective as opposed advance in solid-state electronic technology. Parallel applications are thus the next natural step in computational speed that can accompany and further the process in technology.

Although Mathworks hesitated with the introducing parallelism into Matlab, they finally completed a very good work. The Matlab engineers did their best to design the parallelization of Application Programming Interface (API) in a Matlab. As far as it is in parallel processing possible, black box solutions are provided for inexperienced users, whereas experts can e.g. make use of message passing almost in the same way as in the native Message Passing Interface (MPI). Thus, Parallel Matlab (PCT/MDCS) is an ideal environment for a smooth transition to the world of parallel scientific computing, especially for current Matlab users. The programs that are developed using PCT/MDCS in parallel way for multiarea economic dispatch problem are given in Appendices I to J.

Chapter 6 solves the single area CEED problem using Lagrange's and PSO algorithms in a data-parallel way.

CHAPTER SIX

INVESTIGATION OF THE SINGLE AREA COMBINED ECONOMIC EMISSION DISPATCH PROBLEM CAPABILITIES OF THE DEVELOPED ALGORITHM'S USING PARALLEL COMPUTING

6.1 Introduction

This chapter aim is to develop algorithms and software leading to fast investigation of the capabilities of the developed method, algorithms and Matlab programmes in Chapters 3 and 4 to solve the combined Economic Emission Dispatch problem for various sets of input data and parameters. Normally this investigation requires running of the developed software many times, changing the data and parameters for every run. This process requires a lot of time and efforts if it is done sequentially. The data-parallel capabilities of the parallel Matlab in a Cluster of computers allow the speed of the investigations to be increased many times. The chapter describes the parallel computing toolbox configuration setting and programmes for data-parallel calculation of the Combined Economic Emission Dispatch (CEED) problem. The knowledge from the Chapters three and four is used to implement the sequential Lagrange's and PSO methods, algorithm and programs in a data-parallel ways. The capabilities of Matlab data parallelism software is used to solve the CEED problem in a Cluster of Computers (CC) by Matlab Distributed Computing Engine (MDCE). Parallel computing toolbox configuration setting is described in part 6.2, data parallel implementation of a Lagrange's and PSO methods is described in part 6.3 and 6.4 respectively. Two case studies of data-parallel solution of the CEED problem for i) IEEE 30 bus, and ii) IEEE 118 bus system are implemented using the developed Lagrange's algorithm and one case study for IEEE 30 bus system is implemented in a data-parallel way using the developed PSO method. Comparison of the data-sequential and data-parallel computing results from the solution of the CEED problem are presented in part 6.5.

6.2 Parallel computing toolbox configuration setting to solve the CEED problem in a data-parallel way

This section describes the parallel computing set-up procedure to implement the sequential code in data-parallel way. Parallel computing has local configuration with 8 workers default or user can specify their own configurtion name and assign any number of workers to that configuration using Matlab Cluster of Computers. The configuration defining the certain parameters and properties, and then these settings are applied when creating objects in the MATLAB client. The functions that support the use of configurations are *batch ,createJob, createMatlabPoolJob, createParallelJob, createTask, dfeval, dfevalasync, findResource, matlabpool, pmode* and *set*

One can create and modify configurations through the Configurations Manager. To access the Configurations Manager using the *Parallel* pull-down menu on the MATLAB desktop. select *Parallel > Manage Configurations* to open the Configurations Manger as shown in Figure 6.1.

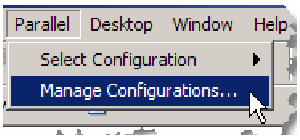


Figure 6. 1: **Manage configuration setting of parallel computing toolbox**

When the first time it opens, the Configurations Manager lists only one configuration called local, which is the default configuration as shown in Figure 6.2

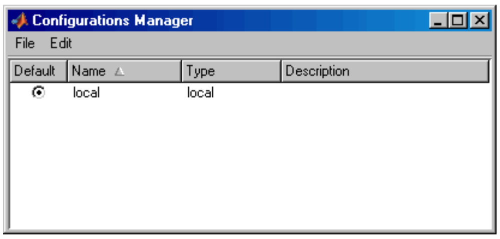


Figure 6. 2: **Local configuration setting of parallel computing toolbox**

To create own configuration setting in the remote Cluster of Computers (CC) using Matlab Distributed Computing Engine (MDCE) the following Procedure should be followed:

1. Start the workers from the Matalab desktop.

   The following steps have to be followed:

   - Execute the m-file *startcluster.m* which is given in Appendix H
   - The waitbar shows the status of starting the workers as shown in Figure 6.3. The DOS window of the MDCE service on the host is given in Figure 6.4.
   - The *nodestatus* commmand in the Matlab desktop is used to view the status of the MDCE service on the *host, jobmanager* and *workers* as shown in Figure 6.5

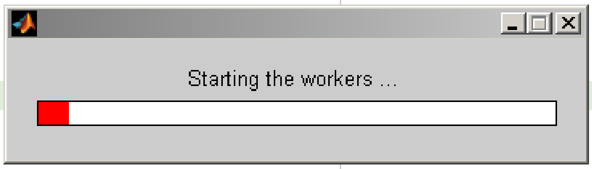


Figure 6. 3: **Waitbar status of starting workers**

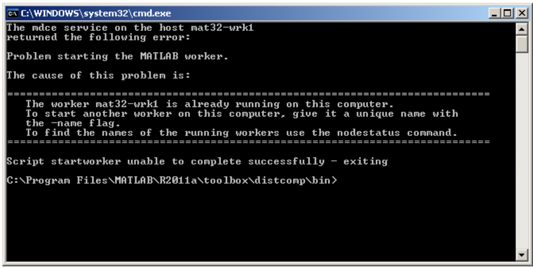


Figure 6. 4: **DOS window of the MDCE service on the host**

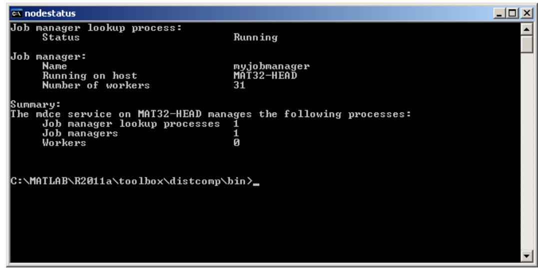


Figure 6. 5: **Node status of the DOS window**

2. Create and modify configurations using the Configurations Manager and its menus and dialog boxes.

   Figure 6.6 shows how to create the own configuration (*jobmanagerconfig1*).

   - Right click the *jobmanagerconfig1* and select the properties menu as shown in Figure 6.7.
   - Set the *jobmanagername* as *myjobmanager* in the scheduler tab.
   - Set the minimum and maximum number of workers that can run the job in the Job tab. The maximum number of workers depends on the available number of workers in used Cluster Computer Laboratory. Cape Peninsula University of Technology (CPUT) - research center "Real Time Distributed System" (RTDS) have 32 workers in the Cluster Computer laboratory, that are used for the thesis investigations.
   - Select the *True* option in return command window output of the Task tab.

     The Figures 6.9 – 6.11 show the settings of *Scheduler, Jobs* and *Tasks* respectively.

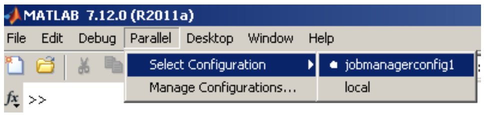


Figure 6. 6: **Creating the own configuration (jobmanagerconfig1)**

3. Validating Configuration

   The Configurations Manager includes a tool for validating configurations.

To validate a configuration, following steps are to be followed:

i. Open the Configurations Manager by selecting on the desktop ***Parallel > Manage Configurations.***

ii. In the Configurations Manager, click the name of the configuration you necessary to be tested in the list of those available. Note that the configuration can be highlighted this way without changing the selected default configuration. So a configuration selected for validation does not need to be the default configuration.

iii. Click *Start Validation*.

The Configuration Validation tool attempts four operations to validate the chosen configuration:

- Uses *findResource* to locate the scheduler
- Runs a *distributed job* using the configuration
- Runs a *parallel job* using the configuration
- Runs a *MATLAB pool job* using the configuration

While the tests are running, the Configurations Manager displays their progress as shown in Figure 6.7.

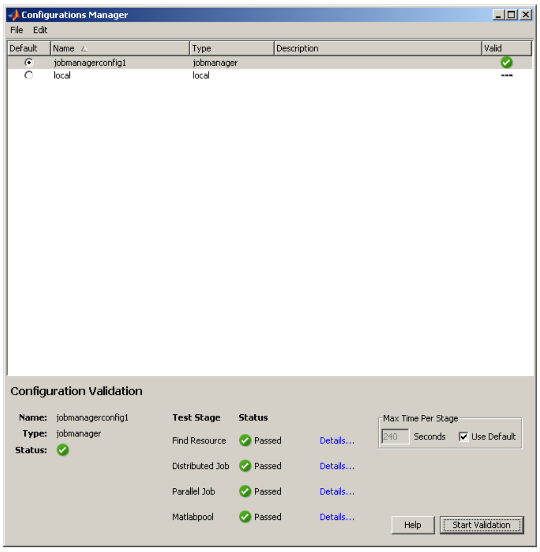


Figure 6.7: Progress display of the Configurations Manager

The photograph of the Cluster Computer Laboratory shown in *Appendix L*.

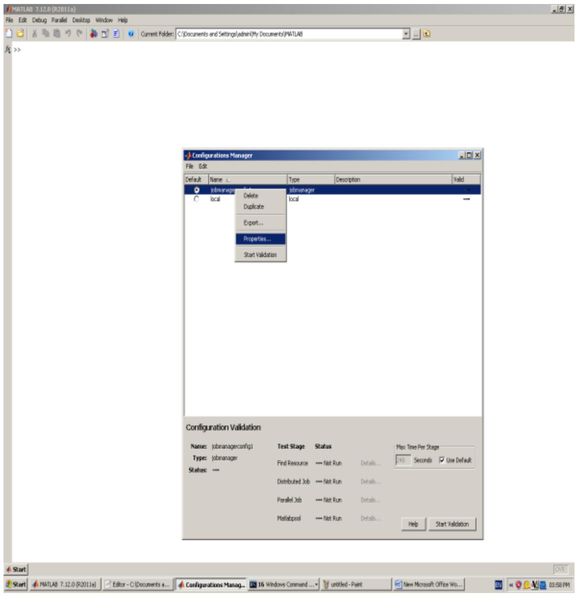


Figure 6. 8: Properties menu of the jobmanagerconfig1

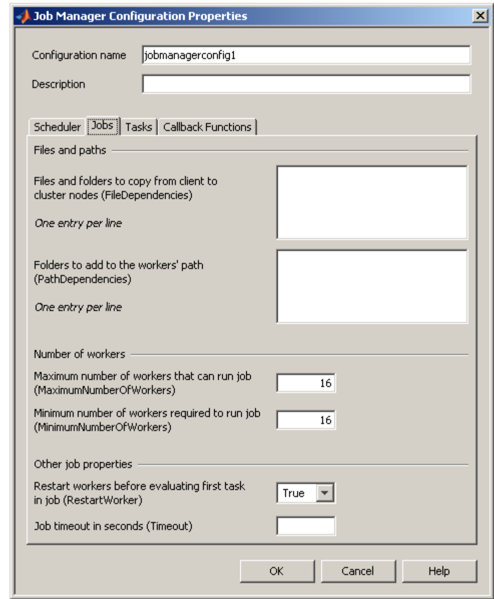


Figure 6. 10: Jobs setting

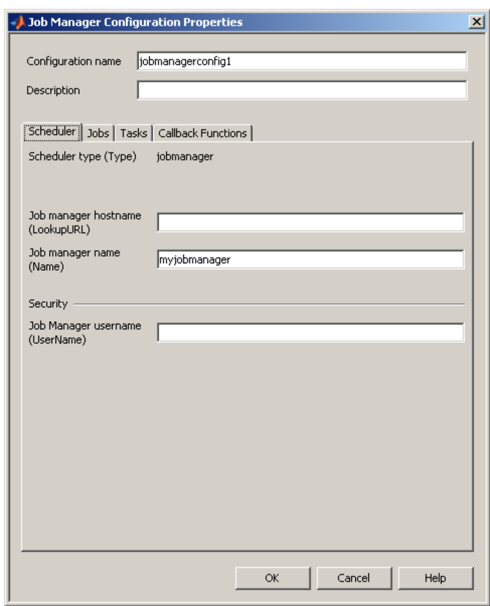


Figure 6. 9: Scheduler setting

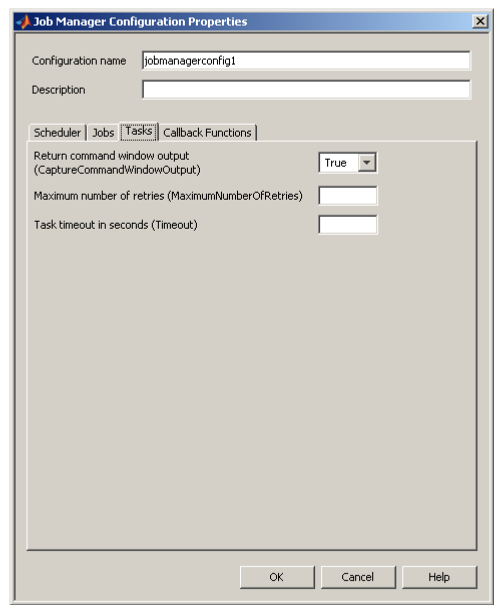


Figure 6. 11: Tasks setting

6.3 Matlab data-parallel program for CEED problem solution using Lagrange's algorithm

Data parallelism is a form of parallelization of computing across multiple workers in parallel computing environments. Data-parallelism focuses on distributing the data across different parallel computing nodes. It contrasts to task parallelism as another form of parallelism. In a Cluster of computer system executing a single set of instructions over the multiple set of data, parallelism is achieved when each workers performs the same task on different pieces of distributed data. For instance, consider 6 workers namely (mat32-wrk1 to mat32-wrk6) in a parallel environment, and the aims to assign a task of different data or power demands. It is possible to tell mat32-wrk1 to do the task on power demand of 125 [MW] and mat32-wrk2 to work on power demand of 150 [MW] simultaneously and so on., thereby reducing the duration of the sequentially execution of the same task for different data. The data can be assigned using conditional statements as described below.

This part of the chapter considers a specific example to solve the data-parallel CEED problem: data-parallel implementation of the CEED problem on the six workers for IEEE 30 bus system as follows:

- mat32-wrk1 solve the CEED problem for power demand of 125 [MW],
- mat32-wrk2 solve the CEED problem for power demand of 150 [MW],
- mat32-wrk3 solve the CEED problem for power demand of 175 [MW],
- mat32-wrk4 solve the CEED problem for power demand of 200 [MW],
- mat32-wrk5 solve the CEED problem for power demand of 225 [MW],
- mat32-wrk6 solve the CEED problem for power demand of 250 [MW],

Since the six workers, distribute the data in parallel way, the job manager would take only one-sixth the time to solve the data parallel CEED problem of performing the same operation sequentially using one CPU only.

The algorithm for calculation of the optimal generation is programmed in a sequential way and is implemented in a data-parallel way. The goal of data parallelization is to obtain high performance and increased speed over the sequential program that solves the same problem. This is to ensure efficient load balancing among processors, reduction in communication overhead and synchronization. The data parallel implementation and the results from the calculations are described using

Lagrange's and PSO methods in parts 6.3 and 6.4 respectively. Figure 6.12 shows the Implementation of Lagrange's data parallel method for CEED problem solution for six different power demands in a Cluster of Computers.

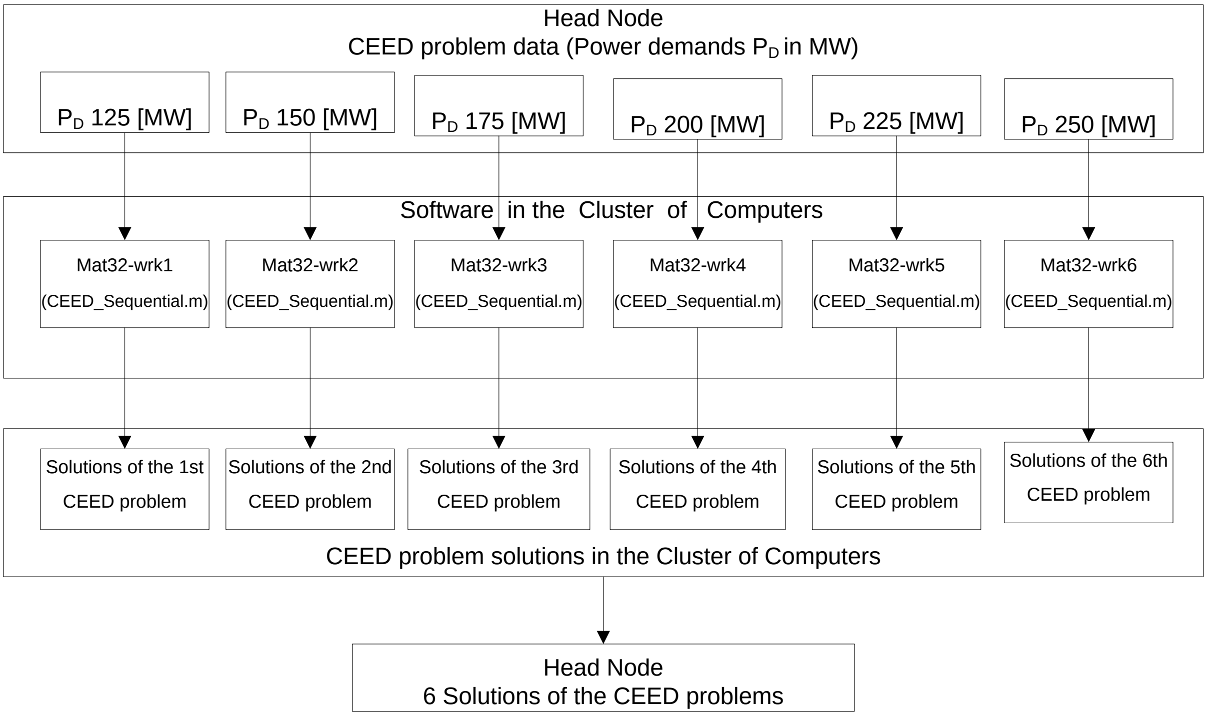


Figure 6.12: **Data parallel implementation of Lagrange's method in a Cluster of Computers**

The program *CEED_Casestudy1_DataParallel.m* transforms the developed Lagrange's sequential program for CEED problem solution using special functions and commands from the Parallel Computing Toolbox. The inputs and outputs are the same as for the sequential program. The program *CEED_Casestudy1_DataParallel_funct.m* is the program that is used to send various power demand to the workers for data parallel calculation of the CEED problem with the *Filedependencies* function from Parallel Computing Toolbox (PCT). The definition of variables in a Matlab script-file *CEED_Casestudy1_DataParallel.m* is as follows:

n ,m, lambda, epsilon, alfa, a, b, c, d, e, f, P_{min}*,* P_{max}*, B, B01, B00*

This variables are defined as local ones since they have to be accessible at a shared memory which in this case is used as jobmanager. They are defined as local variables in the script file *CEED_Casestudy1_DataParallel.m*

6.3.1 Starting parallel computation for CEED problem in a data-parallel way

The parallel computing configuration setting procedure described in part 6.2 is used for data parallelization of the CEED problem. The following commands are used:

```
jm = findResource('scheduler','configuration','jobmanagerconfig1')
pjob1=createParallelJob(jm,'Configuration','jobmanagerconfig1');
set(pjob1,'Configuration','jobmanagerconfig1')
set(pjob1,'MinimumNumberOfWorkers',6);
set(pjob1,'MaximumNumberOfWorkers',6);
```

The function file *CEED_Casestudy1_DataParallel_funct.m* is given in *Appendix E2* on the client path, but it has to be accessible to the workers. The *FileDependencies* property of the *pjob1* is used to transfer this function to all the workers that are available on the computer cluster.

```
set(pjob1,'FileDependencies',{'CEED_Casestudy1_DataParallel_funct.m'})
```

Starting parallel computing is done by creating a task which is the same for every worker. The task has 14 outputs arguments and 12 input arguments to the function script file *CEED_Casestudy1_DataParallel_funct.m*

```
task1=createTask(pjob1,@CEED_Casestudy1_DataParallel_funct,13,{Pmax,Pmin,f,e
,d,c,b,a,lambda,B,B00,B01})
submit(pjob1)
```

The part of the program implemented in a data-parallel way is described by the function file *CEED_Casestudy1_DataParallel_funct.m*

6.3.2 Function for parallel calculation for CEED problem

The function *CEED_Casestudy1_DataParallel_funct.m* is used to calculate the CEED problem in data-parallel way. The *labindex* command is used to send various power demands to different workers. The syntax is given as follows:

```
if labindex ==1
        P_D=125;
end
if labindex==2
        P_D=150;
end
if labindex==3
        P_D=175;
end
if labindex==4
```

```
        PD=200;
end
if labindex==5
        PD=225;
end
if labindex==6
        PD=250;
end
```

The power demand of 125 [MW] is sent to worker 1, 150 [MW] is sent to Worker 2 and so on. When the calculation completes, the output results are transformed to the variables P_D, lambda, P1, P2, P3, P4, P5, P6, P_L, F_C, E_T, F_T, iter time by the worker with *labindex* command

```
waitForState(pjob1,'finished')
results=getAllOutputArguments(pjob1)
```

The parallel version of the programs *CEED_Casestudy1_DataParallel.m and CEED_Casestudy1_DataParallel_funct.m* for solution of the CEED problem using Lagrange's algorithm is given in *Appendix E*. Algorithm given in Figure 3.1 is used.

6.3.3 Results from the calculation of the CEED problem solution using Lagrange's algorithm in the Matlab data-parallel environment

The CEED problem is formulated and solved using Lagrange's algorithm in data-parallel way for two case studies (i) IEEE 30 bus and (ii) IEEE 118 bus system. The data are given in (Gnanadass, 2010) and (Guerrero, 2004) and are described in Table 3.1, 3.10 and 3.11 respectively.

6.3.3.1 Case Study 1: IEEE 30 bus system

The data of the power system are given in (Gnanadass, 2005). It consists of six generators and the constraint of the transmission loss is considered in this case. Software for implementation of the Lagrange’s algorithm is developed for sequential solution in one computer, Chapter 3 (Krishnamurthy and Tzoneva, 2013). This software has to be used for investigation of the problem solution for various data sets of the power demand, Lagrange’s multiplier and the penalty factors.

Requirements towards the solution of the CEED problem are:

1) It is necessary to find solutions of the CEED problem for 6 different values of the power demand P_D

2) Every solution from point 1 above to be calculated for 2 different initial values of the Lagrange's multiplier $\lambda=4$ and $\lambda=10$.

3) Every solution from points 1 and 2 above to be calculated for 2 types of penalty factors Min-Max and Max-Max.

The vector of the power demand to be used in the investigation of the IEEE 30 bus system is distributed between 6 workers from 125 to 250 [MW] with an incremental of 25 [MW] to each workers. i.e., power demand of 125 [MW] is send to worker 1, 150 [MW] is send to Worker 2 and so on as shown in Figure 6.12 and in Table 1. In this way the sequential problem is solved 6 times in parallel. These solutions are for $\lambda=4$ and Min-Max penalty factor. Then the next 6 are for $\lambda=4$ and Max-Max penalty factor. Further the solutions are repeated for $\lambda=10$ and Min-Max and Max-Max penalty factors respectively. The total run of the software is 4 times and the calculated solutions are 24. The Lagrange's parallel solutions of the CEED problem using Min-Max and Max-Max price penalty factors for IEEE 30 bus system are given in Tables 6.1 and 6.2 respectively.

Table 6.1: **Lagrange's data-parallel solution of the CEED problem for six sets of input data and based on Min-Max price penalty factor for IEEE 30 bus system**

Name of the worker	P_D [MW]	Lambda	Generator active power in [MW]						P_L [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]	Initial lambda 4		Initial lambda 10	
													Number of iterations	CT [S]	Number of iterations	CT [s]
mat32-wrk1	125	2.6643	59.2509	20.0000	15.0000	10.0000	10.0000	12.0000	1.2499	308.1632	144.5308	377.1885	195	0.0929	219	0.2305
mat32-wrk2	150	3.0287	78.8322	26.2655	15.0000	10.0000	10.0000	12.0000	2.0967	373.5027	162.2308	448.5180	153	0.0785	180	0.2376
mat32-wrk3	175	3.3683	96.7123	32.8893	16.5113	10.0000	10.0000	12.0000	3.1120	443.9698	190.4200	528.5602	139	0.0777	170	0.1137
mat32-wrk4	200	3.7044	114.0706	39.3658	18.8705	10.0000	10.0000	12.0000	4.3059	519.5062	228.1564	616.9564	129	0.0985	166	0.1103
mat32-wrk5	225	3.9709	127.5970	44.4448	20.7449	14.3841	11.2167	12.0000	5.3865	601.1057	268.0774	713.0565	75	0.0417	134	0.0566
mat32-wrk6	250	4.1411	138.9676	48.7365	22.3455	18.8091	13.7142	13.8596	6.4336	687.0703	310.2824	815.1811	80	0.0443	129	0.0552
Head node total operating time in [S]													17.5630		17.8280	

Table 6.2: **Lagrange's data-parallel solution of the CEED problem for six sets of input data and based on Max-Max price penalty factor for IEEE 30 bus system**

Name of the worker	P_D [MW]	Lambda	Generator active power in [MW]						P_L [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]	Initial lambda 4		Initial lambda 10	
													Number of iterations	CT [S]	Number of iterations	CT [s]
mat32-wrk1	125	3.2435	59.2509	20.0000	15.0000	10.0000	10.0000	12.0000	1.2499	308.1633	144.5308	619.2622	544	0.2880	621	0.4548
mat32-wrk2	150	4.3942	79.5282	25.5778	15.0000	10.0000	10.0000	12.0000	2.1070	373.4808	162.2098	716.1593	360	0.2275	424	0.3035
mat32-wrk3	175	5.0946	91.6415	31.7899	17.0780	13.4044	11.9356	12.0000	2.8502	447.4312	186.5639	835.6383	259	0.0915	283	0.1625
mat32-wrk4	200	5.6416	100.9848	36.5993	19.2509	17.3132	14.8895	14.5195	3.5581	527.9225	214.4258	969.9064	241	0.0813	261	0.1515
mat32-wrk5	225	6.1847	110.1592	41.3373	21.4100	21.1778	17.8128	17.4464	4.3444	612.0153	247.2918	1117.6971	248	0.0837	261	0.1571
mat32-wrk6	250	6.7333	119.3260	46.0868	23.5930	25.0656	20.7566	20.3862	5.2152	699.5437	285.3632	1279.1301	254	0.0860	260	0.1475
Head node total operating time in [S]													17.0650		17.2350	

Figure 6.13 to 6.16 show the comparison of computation time and number of iterations needed to reach the optimum solution of Lagrange's method using Min-Max and Max-Max penalty factors for two different initial selection of Lagrange's multipliers (lambda) of 4 and 10 respectively.

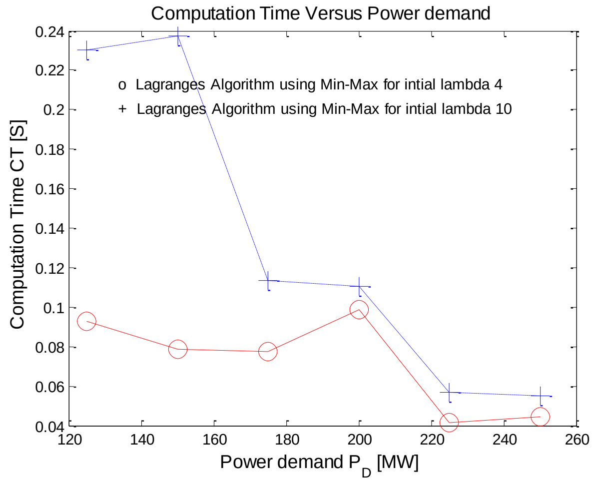


Figure 6.13: **Comparison of Computation time of the Lagrange's method using Min-Max penalty factor for different values of λ**

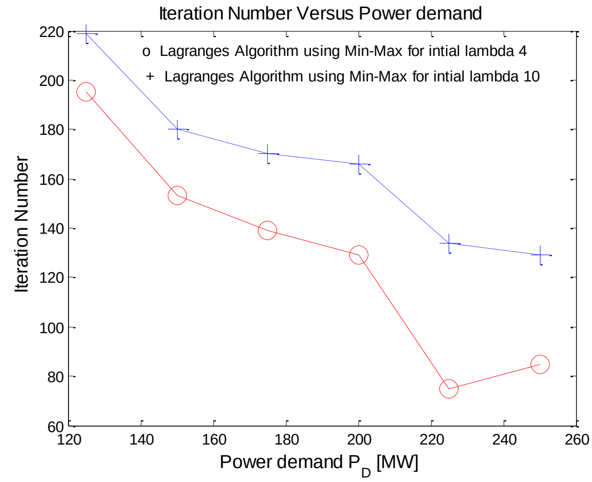


Figure 6.14: **Comparison of the number of iterations of the Lagrange's method using Min-Max penalty factor for different values of λ**

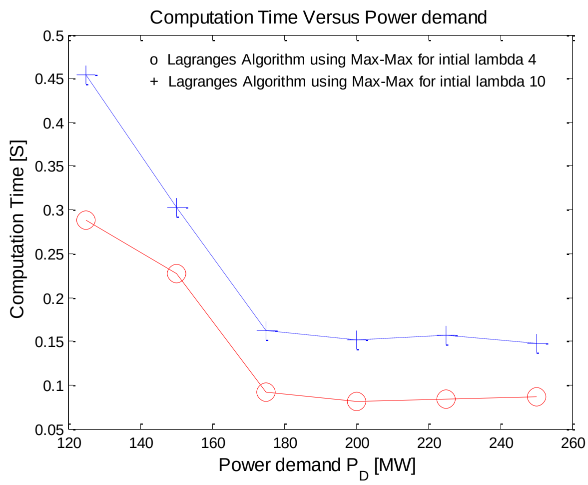


Figure 6.15: **Comparison of Computation time of the Lagrange's method using Max-Max penalty factor for different values of λ**

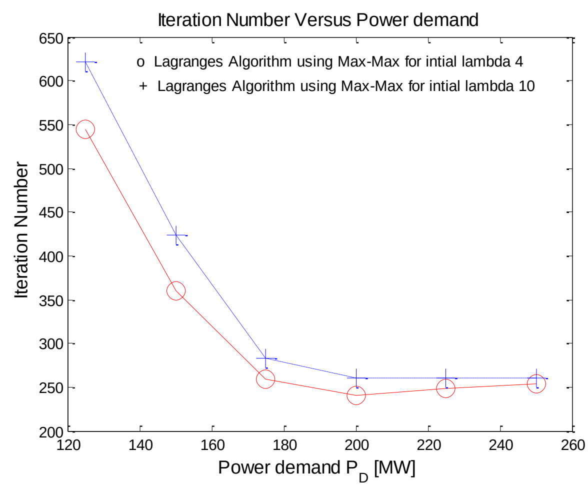


Figure 6.16: **Comparison of the number of iterations of the Lagrange's method using Max-Max penalty factor for different values of λ**

Figures 6.17 and 6.18 show the lambda and deltalambda values over the iteration number as given in Equation (3.23) and (3.22) respectively.

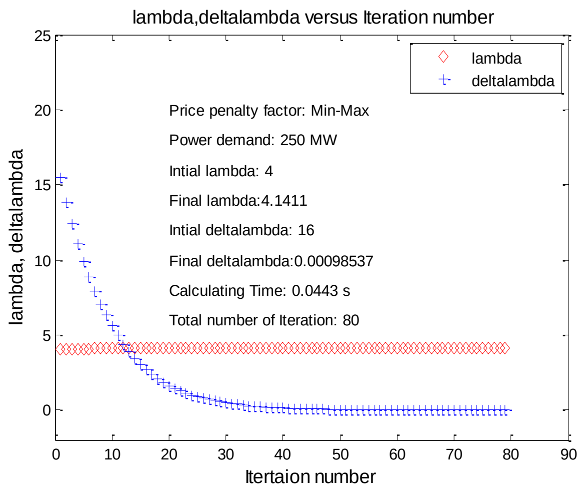


Figure 6.17: **Lambda and deltalambda values of the Lagrange's method based on Min-Max penalty factor for initial lambda of 4**

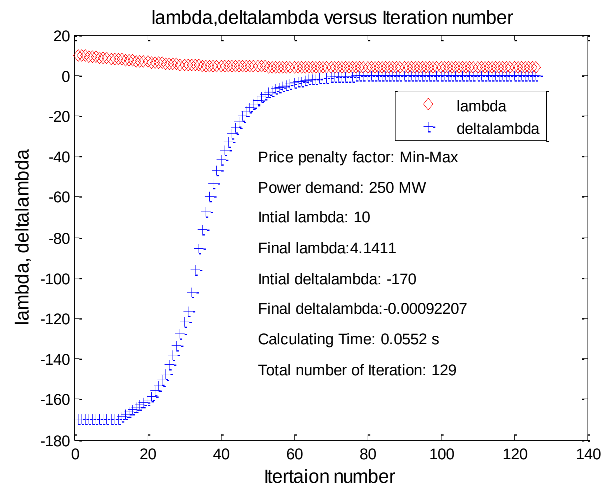


Figure 6.18: **Lambda and deltalambda values of the Lagrange's method based on Min-Max penalty factor for initial lambda of 10**

6.3.3.2 Discussion on Lagrange's parallel computing solution of the CEED problem for the IEEE 30 bus system:

The results from the parallel implementation of the Lagrange's method using Min-Max and Max-Max price penalty factors are given in Tables 6.1 and 6.2 respectively. These solutions of the problem are done by 6 workers. Every worker solves the problem for different power demand and for 2 initial values of λ. This means 6 parallel solutions are calculated for every value of λ. The parallel solution of the Lagrange's algorithm is the same as the sequential one but the data-parallel calculation produces 6 different solutions for the 6 different values of the power demand. Number of iterations used to obtain the optimal solution are different for every worker because the convergence to the optimal solution depends on the initial values of λ and on the required power demand.

The Matlab parallel computing software has capability to implement the sequential code of the CEED problem in data-parallel way using Lagrange's method. The parallel solution of the CEED problem given in Table 6.1 and 6.2 is the same as the sequential obtained one. The Lagrange's method solution depends on the optimal lambda value of the Lagrange's multiplier. It can be seen from the Tables 6.1 and 6.2, that the computation time of the Lagrange's method depends on the initial selection of the Lagrange's variable lambda. It is proved that if the initial Lagrangian variable value is close to the optimal one, then the problem optimal solution requires less computation time, otherwise it requires more computational time.

When initial lambda is 4, computational time and number of iterations needed to obtain the Lagrange's solution using Min-Max penalty factor is less in comparison with the solution using the initial lambda value of 10 as shown in Figure 6.13 and 6.14. It seen that the solution of the Lagrange's method is influenced by the initial selection of the Lagrange's multiplier (lambda). Figures 6.15 and 6.16 show the comparison of computation time and the number of iterations obtained using Max-Max penalty factor for two different initial selections of the Lagrange's multiplier (lambda).

It shows that the initial lambda value of 4 uses less computational time and number of itertaions in comparison with the initial value of lambda of 10. Lagrange's tolerance value $\Delta\lambda$ given by Equation (3.23) and λ value given by Equation (3.22) changes over the iteration numbers as shown in Figure 6.17 and 6.18 respectively. In Table 6.1 for 250 [MW] power demand, the initial value of lambda of 4 reaches the optimal point for the final value of lambda of 4.1411 and takes 0.0443 seconds but the initial value of lambda of 10 reaches the optimal point for the same final value of lambda of 4.1411 but it takes 0.0552 seconds. It is proven that Lagrange's calculation time is less for the initial lambda value of 4 in comparison with the other initial value of lambda of 10.

6.3.3.3 Case Study 2: IEEE 118 bus system

The CEED problem using Lagrange's algorithm is solved in data-parallel way for an IEEE 118 bus system. The data of the power system are given in (Guerrero, 2004). The system consists of fourteen generators and the transmission loss constraint is considered in this case. The power demand of the 14 generators IEEE 118 bus system is distributed between 14 workers from 3668 to 3920 [MW] as follows:

- mat32-wrk1 solve the CEED problem for power demand of 3668 [MW]
- mat32-wrk2 solve the CEED problem for power demand of 3680 [MW]
- mat32-wrk3 solve the CEED problem for power demand of 3700 [MW]
- mat32-wrk4 solve the CEED problem for power demand of 3720 [MW]

- mat32-wrk5 solve the CEED problem for power demand of 3740 [MW]
- mat32-wrk6 solve the CEED problem for power demand of 3760 [MW]
- mat32-wrk7 solve the CEED problem for power demand of 3780 [MW]
- mat32-wrk8 solve the CEED problem for power demand of 3800 [MW]
- mat32-wrk9 solve the CEED problem for power demand of 3820 [MW]
- mat32-wrk10 solve the CEED problem for power demand of 3840 [MW]
- mat32-wrk11 solve the CEED problem for power demand of 3860 [MW]
- mat32-wrk12 solve the CEED problem for power demand of 3880 [MW]
- mat32-wrk13 solve the CEED problem for power demand of 3900 [MW]
- mat32-wrk14 solve the CEED problem for power demand of 3920 [MW]

Different initial lambda values of 4 and 10 are used in Lagrange's data-parallel program of the CEED problem. The Lagrange's data-parallel solutions of the CEED problem using Min-Max and Max-Max price penalty factors for IEEE 118 bus system are given in Table 6.3 and 6.4 respectively.

Table 6.3: **Lagrange's data-parallel solutions of the CEED problem for 14 sets of input data and based on Min-Max price penalty factor for IEEE 118 bus system**

Name of the worker	P_D	lambda	P1	P2	P3	P4	P5	P6	P7	P8	P9
mat32-wrk1	3668	8.4132	53.5381	460.3228	287.2570	179.6683	360.9079	288.8098	169.9043	147.6425	329.8012
mat32-wrk2	3680	8.4887	52.5567	464.9113	290.6906	180.5749	364.5699	291.6788	171.5777	149.2418	332.4505
mat32-wrk3	3700	8.6098	50.9908	472.1960	296.1481	181.9767	370.3777	296.2358	174.2215	151.7704	336.6501
mat32-wrk4	3720	8.7227	50.0000	478.8937	301.1730	183.2248	375.7112	300.4281	176.6388	154.0839	340.5050
mat32-wrk5	3740	8.8253	50.0000	484.9076	305.6911	184.3122	380.4953	304.1946	178.7984	156.1521	343.9613
mat32-wrk6	3760	8.9248	50.0000	490.6719	310.0274	185.3250	385.0766	307.8070	180.8588	158.1263	347.2702
mat32-wrk7	3780	9.0215	50.0000	496.2157	314.2033	186.2718	389.4789	311.2834	182.8318	160.0175	350.4489
mat32-wrk8	3800	9.1159	50.0000	501.5627	318.2364	187.1597	393.7215	314.6384	184.7269	161.8345	353.5117
mat32-wrk9	3820	9.2082	50.0000	506.7326	322.1411	187.9945	397.8206	317.8845	186.5520	163.5847	356.4705
mat32-wrk10	3840	9.2985	50.0000	511.7421	325.9296	188.7811	401.7898	321.0319	188.3138	165.2746	359.3352
mat32-wrk11	3860	9.3871	50.0000	516.6053	329.6125	189.5238	405.6405	324.0895	190.0179	166.9092	362.1142
mat32-wrk12	3880	9.4742	50.0000	521.3344	333.1985	190.2262	409.3827	327.0648	191.6692	168.4932	364.8148
mat32-wrk13	3900	9.5599	50.0000	525.9400	336.6956	190.8915	413.0250	329.9644	193.2721	170.0306	367.4434
mat32-wrk14	3920	9.6443	50.0000	530.4312	340.1104	191.5222	416.5750	332.7939	194.8300	171.5248	370.0053

Continuation of Table 6.3

Name of the worker	P10	P11	P12	P13	P14	P_L [MW]	F_C	E_T	F_T	Initial lambda 4		initial lambda 10	
										Number of iterations	CT [S]	Number of iterations	CT [s]
mat32-wrk1	317.6919	65.8266	162.1047	1000.0000	50.0000	50.447	18046.1791	13577.1113	18251.3046	76	0.1846	72	0.1379
mat32-wrk2	320.0804	65.1526	163.9344	1000.0000	50.0000	217.4207	24535.7394	28752.3313	24976.7155	74	0.1723	70	0.1089
mat32-wrk3	323.8701	64.0773	166.8673	1000.0000	50.0000	235.3826	24780.8642	29184.3579	25227.9988	72	0.1327	66	0.1039
mat32-wrk4	327.3527	63.0841	169.5948	1000.0000	50.0000	250.6910	25008.9760	29588.1862	25461.8649	65	0.1072	59	0.1024
mat32-wrk5	330.4787	62.1896	172.0691	1000.0000	50.0000	263.2510	25217.2834	29956.2061	25675.4150	63	0.1064	57	0.1016
mat32-wrk6	333.4748	61.3310	174.4636	1000.0000	50.0000	274.4335	25418.1626	30313.6911	25881.3850	61	0.1040	55	0.0999
mat32-wrk7	336.3563	60.5047	176.7877	1000.0000	50.0000	284.4009	25612.4911	30661.8872	26080.6708	60	0.0978	53	0.0837
mat32-wrk8	339.1359	59.7081	179.0491	1000.0000	50.0000	293.2858	25800.9817	31001.8032	26273.9994	58	0.0890	51	0.0836
mat32-wrk9	341.8243	58.9388	181.2544	1000.0000	50.0000	301.1990	25984.2336	31334.2902	26461.9825	57	0.0871	50	0.0795
mat32-wrk10	344.4303	58.1949	183.4089	1000.0000	50.0000	308.2331	26162.7457	31660.0548	26645.1293	56	0.0864	48	0.0726
mat32-wrk11	346.9614	57.4748	185.5173	1000.0000	50.0000	314.4673	26336.9447	31979.7011	26823.8749	55	0.0860	47	0.0710
mat32-wrk12	349.4241	56.7770	187.5836	1000.0000	50.0000	319.9696	26507.1967	32293.7480	26998.5931	54	0.0792	46	0.0700
mat32-wrk13	351.8241	56.1002	189.6112	1000.0000	50.0000	324.7988	26673.8214	32602.6489	27169.6102	54	0.0692	44	0.0621
mat32-wrk14	354.1663	55.4433	191.6031	1000.0000	50.0000	329.0064	26837.0922	32906.7892	27337.2051	53	0.0614	43	0.0582
Head node total operating time in [S]]										17.8410		17.5160	

Table 6.4: **Lagrange's data-parallel solutions of the CEED problem for 14 sets of input data and based on Max-Max price penalty factor for IEEE 118 bus system**

Name of the worker	P_D	lambda	P1	P2	P3	P4	P5	P6	P7	P8	P9
mat32-wrk1	3668	17.9645	55.7542	542.9174	355.1844	234.5058	432.9026	335.0527	238.1653	223.4820	374.8714
mat32-wrk2	3680	18.0835	54.9197	546.0604	357.3359	235.0007	435.2273	337.0768	239.1245	224.4482	376.5263
mat32-wrk3	3700	18.2795	53.5515	551.1956	360.8553	235.7921	439.0242	340.3842	240.6852	226.0225	379.2275
mat32-wrk4	3720	18.4725	52.2098	556.2095	364.2968	236.5440	442.7300	343.6139	242.2013	227.5546	381.8621
mat32-wrk5	3740	18.6629	50.8932	561.1103	367.6658	237.2592	446.3510	346.7712	243.6757	229.0472	384.4345
mat32-wrk6	3760	18.8444	50.0000	565.7388	370.8523	237.9166	449.7698	349.7537	245.0616	230.4524	386.8617
mat32-wrk7	3780	19.0093	50.0000	569.9109	373.7288	238.4942	452.8506	352.4426	246.3053	231.7155	389.0479
mat32-wrk8	3800	19.1725	50.0000	574.0059	376.5558	239.0472	455.8737	355.0824	247.5211	232.9519	391.1920
mat32-wrk9	3820	19.3340	50.0000	578.0279	379.3363	239.5769	458.8423	357.6757	248.7103	234.1630	393.2965
mat32-wrk10	3840	19.4940	50.0000	581.9807	382.0726	240.0844	461.7591	360.2250	249.8746	235.3501	395.3635
mat32-wrk11	3860	19.6525	50.0000	585.8676	384.7670	240.5710	464.6269	362.7324	251.0150	236.5144	397.3950
mat32-wrk12	3880	19.8097	50.0000	589.6919	387.4216	241.0374	467.4479	365.2002	252.1328	237.6570	399.3926
mat32-wrk13	3900	19.9657	50.0000	593.4565	390.0384	241.4847	470.2245	367.6301	253.2291	238.7790	401.3580
mat32-wrk14	3920	20.1205	50.0000	597.1639	392.6190	241.9136	472.9585	370.0238	254.3048	239.8811	403.2928

Continuation of Table 6.4

Name of the worker	P10	P11	P12	P13	P14	P_L [MW]	F_C	E_T	F_T	Initial lambda 4		Initial lambda 10	
										Number of iterations	CT [S]	Number of iterations	CT [s]
mat32-wrk1	359.9949	73.9307	182.9901	1000.0000	50.0000	142.9925	17916.1113	14219.3073	25015.1159	136	0.3966	133	0.3057
mat32-wrk2	361.4657	73.2510	184.1926	1000.0000	50.0000	794.6301	28234.0395	34646.8013	47463.0474	134	0.3806	132	0.2929
mat32-wrk3	363.8674	72.1345	186.1720	1000.0000	50.0000	798.9130	28412.1605	34997.8124	47803.4485	133	0.3408	130	0.2877
mat32-wrk4	366.2112	71.0376	188.1224	1000.0000	50.0000	802.5941	28586.8785	35343.8505	48138.0594	131	0.3384	128	0.2853
mat32-wrk5	368.5011	69.9596	190.0460	1000.0000	50.0000	805.7156	28758.4298	35685.2593	48467.2816	130	0.3140	127	0.2836
mat32-wrk6	370.6631	68.9365	191.8782	1000.0000	50.0000	807.8858	28922.0091	36010.6303	48781.0517	119	0.2925	116	0.2660
mat32-wrk7	372.6116	68.0105	193.5430	1000.0000	50.0000	808.6616	29071.9798	36306.3865	49067.4830	117	0.2904	114	0.2605
mat32-wrk8	374.5239	67.0984	195.1891	1000.0000	50.0000	809.0422	29219.7299	36598.8735	49350.1290	115	0.2621	112	0.2579
mat32-wrk9	376.4021	66.1999	196.8177	1000.0000	50.0000	809.0494	29365.3832	36888.2742	49629.2010	114	0.2545	111	0.2508
mat32-wrk10	378.2480	65.3144	198.4297	1000.0000	50.0000	808.7030	29509.0497	37174.7491	49904.8861	112	0.2515	110	0.2472
mat32-wrk11	380.0635	64.4415	200.0260	1000.0000	50.0000	808.0213	29650.8333	37458.4519	50177.3621	111	0.2490	109	0.2340
mat32-wrk12	381.8500	63.5810	201.6075	1000.0000	50.0000	807.0210	29790.8270	37739.5184	50446.7872	110	0.2481	108	0.2339
mat32-wrk13	383.6091	62.7325	203.1749	1000.0000	50.0000	805.7175	29929.1180	38018.0772	50713.3103	110	0.2397	107	0.2292
mat32-wrk14	385.3421	61.8957	204.7289	1000.0000	50.0000	804.1250	30065.7843	38294.2414	50977.0639	109	0.2196	106	0.2079
Head node total operating time in [S]										18.7190		18.1563	

Figure 6.19 and 6.20 show the Lagrange's data-parallel solution of the CEED problem for 6 and 14 sets of input data for IEEE 30 and 118 bus system respectively as obtained in the Matlab environment.

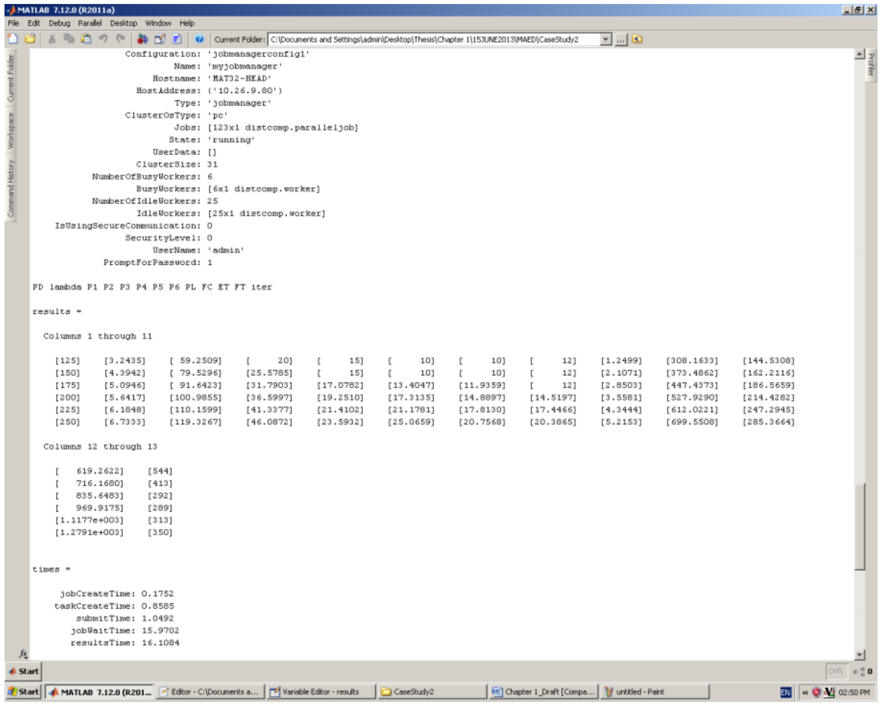


Figure 6. 19: **Lagrange's data-parallel solution of the CEED problem for 6 sets of input data for IEEE 30 bus system**

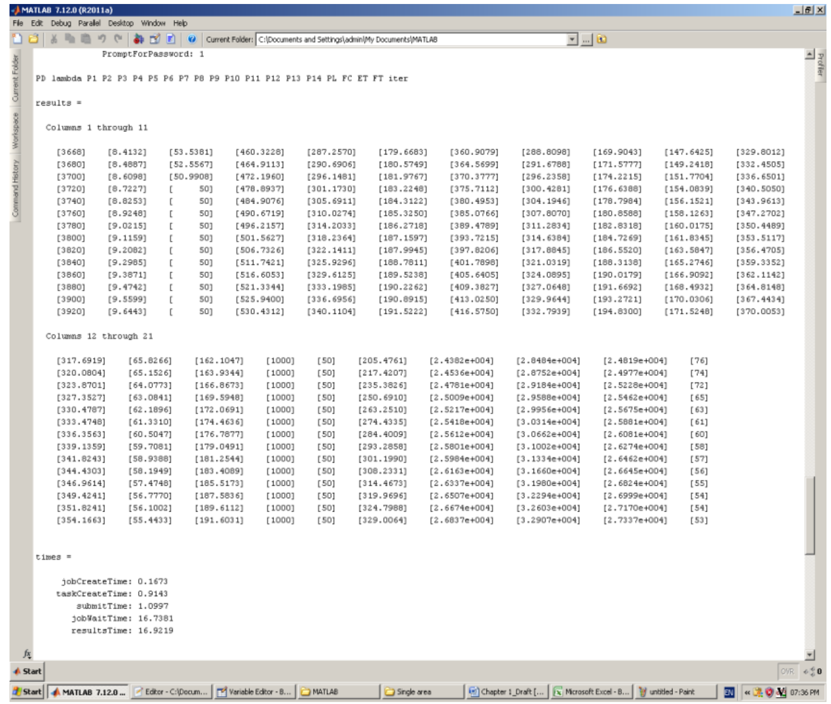


Figure 6. 20: **Lagrange's data-parallel solution of the CEED problem for 14 sets of input data for IEEE 118 bus system**

Figure 6.21 to 6.24 show the comparison of computation time and number of iterations needed to reach the optimum solution of Lagrange's method based on Min-Max and Max-Max penalty factors for two different initial selections of lambda of 4 and 10 for the IEEE 118 bus system.

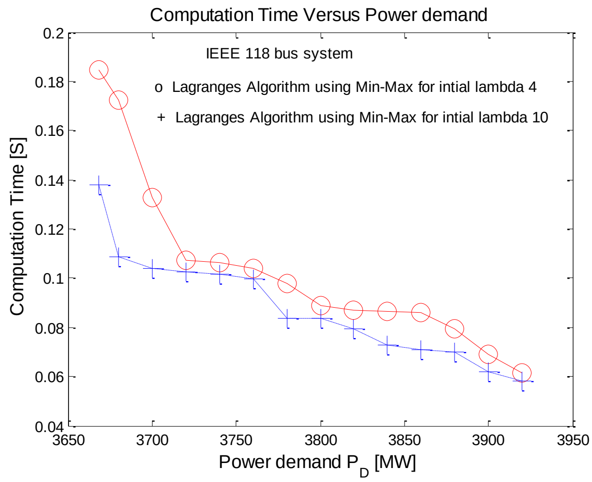


Figure 6.21: **Comparison of the computation time of the Lagrange's method based on Min-Max penalty factor for different values of λ for the IEEE 118 bus system**

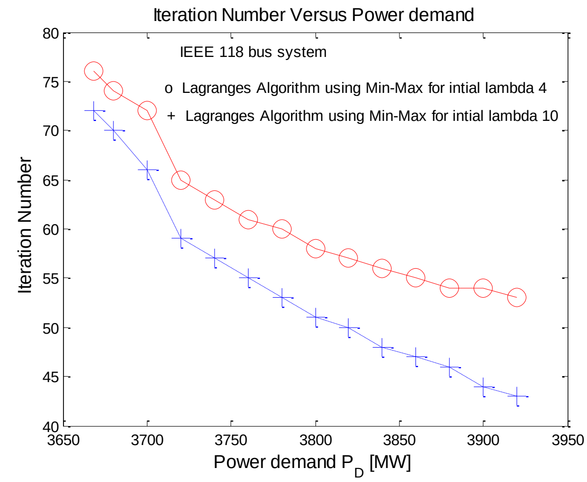


Figure 6.22: **Comparison of the number of iterations used for the Lagrange's method based on Min-Max penalty factor for different values of λ for the IEEE 118 bus system**

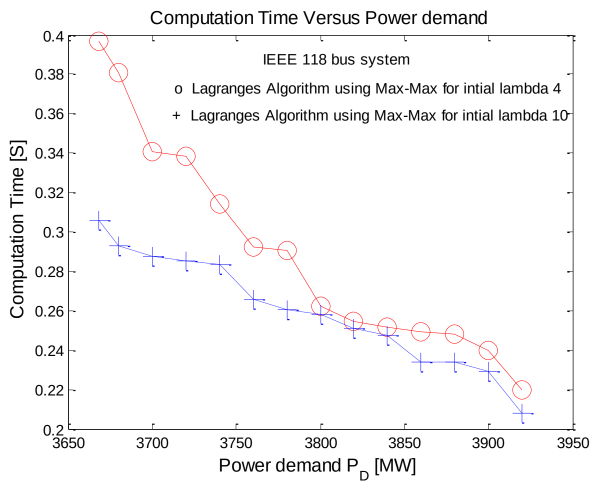


Figure 6.23: **Comparison of the computation time of the Lagrange's method based on Max-Max penalty factor for different values of λ for the IEEE 118 bus system**

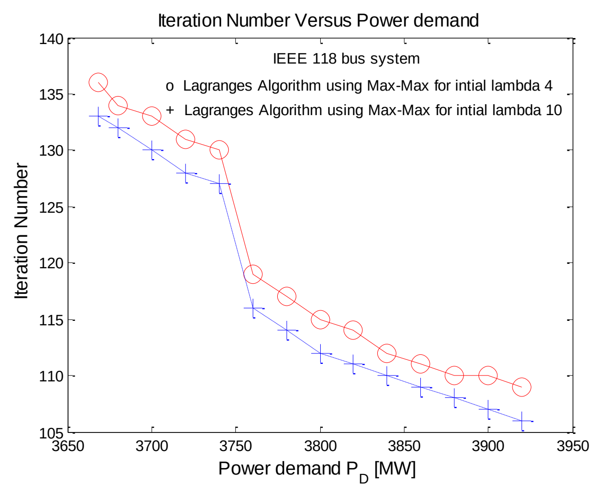


Figure 6.24: **Comparison of number of iterations used for the Lagrange's method based on Max-Max penalty factor for different values of λ for the IEEE 118 bus system**

6.3.3.4 Discussion on Lagrange's data-parallel computing solutions of the CEED problem for the IEEE 118 bus system:

The results from the data-parallel implementation of the Lagrange's method using Min-Max and Max-Max price penalty factors for the IEEE 118 bus system are given in Tables 6.3 and 6.4 respectively for the different load levels of 3668 to 3920 MW. The solutions of the problem are done by 14 workers. Every worker solves the problem for different power demand and for 2 initial values of λ. This means 14 parallel solutions are calculated for every value of λ. The parallel calculation of the Lagrange's algorithm for one worker is the same as the sequential one but the data-parallel calculation produces 14 different solutions for the 14 different values of the power demand. The sequential one produces only one solution and has to be repeated 14 times to obtain all the solutions for the required power demand. Number of iterations used to obtain the optimal solution are different for every worker because the conversion to the optimal solution depends on the initial values of λ and on the required power demand.

The Matlab parallel computing software have capability to implement the sequential code of the CEED problem in data-parallel way using Lagrange's method. The parallel solution of the CEED problem given in Table 6.3 and 6.4 for the load level 3668 MW is the same as for the sequential solution given in Table 3.13. The Lagrange's method solution depends on the optimal lambda value of the Lagrange's multiplier. It can be seen from the Tables 6.3 and 6.4, that the computation time of the Lagrange's method depends on the initial selection of the Lagrange's variable lambda. As in the previous case the initial Lagrangian variable value is close to the optimal one, then it requires less computation time to obtain the global solution otherwise it requires more computational time.

When initial value of lambda is 10, the computational time and number of iterations needed to obtain Lagrange's solution using Min-Max penalty factor is less in comparison with the solution using the initial lambda value of 4 as shown in Figure 6.21 and 6.22. It shows that solution of the Lagrange's method is influenced by the initial selection of the Lagrange's multiplier (lambda). Figures 6.23 and 6.24 show the comparison of computation time and the number of iterations obtained using Max-Max penalty factor for two different initial selection of Lagrange's multiplier (lambda).

It shows that the initial lambda value of 10 uses less computational time and the number of iterations in comparison with the case when the initial value of lambda is 4.

In Table 6.3 for 6920 [MW] power demand, the initial value of lambda of 10 reaches the optimal point for the final value of lambda of 9.6443 and takes 0.0582 seconds but the initial value of lambda of 4 reaches the optimal point for the same final value of lambda of 9.6443 but it takes 0.0614 seconds. It is proven that Lagrange's calculation time is less for the initial lambda value of 10 in comparison with the other initial value of lambda of 4.

The Matlab parallel computing software has capability to solve the CEED problem in a data-parallel way for both small scale and large scale power system networks. The data-parallel solution of the CEED problem for IEEE 30 bus and 118 bus systems are the same as the sequential ones given in Table 3.13 for the load level 3668 MW. The comparison of the sequential and the data-parallel computational times for the used Lagrange's method is given in Table 6.5.

Table 6.5: Comparison of computation time of sequential and data-parallel solution of the CEED problem based on Lagrange's method

Type of price penalty factor	Computation method	IEEE 30 bus system			IEEE 118 bus system		
		initial lambda	Computation Time of one problem	Computation Time of 6 problems	initial lambda	Computation Time of one problem	Computation Time of 14 problems
Min-Max	Sequential (Krishnamurthy and Tzoneva,2013)	4	3.2542	19.5252	4	46.2312 (Refer Table 3.13)	647.2368
	Data-parallel	4	2.9271	17.5626	4	1.2743	17.8402
	Data-parallel	10	2.9713	17.8278	10	1.2511	17.5154
Max-Max	Sequential (Krishnamurthy and Tzoneva,2013)	4	3.5987	21.5922	4	56.6230 (Refer Table 3.13)	792.7220
	Data-parallel	4	2.8441	17.0646	4	1.3369	18.7166
	Data-parallel	10	2.8725	17.2350	10	1.2968	18.1552

In parallel computation the total operating time includes time spend on job creation, task creation, job submission, wait for state, and fetch results time. Hence, it is necessary to use parallel computing for large complex power system problems where the calculation time is more in comparison with the communication time. It is proved that implementation of data-parallel computation reduces the computation time of the CEED problem of IEEE 30 and 118 bus system in comparison with the sequential one.

6.4 Data-parallel results for the CEED problem solution using the PSO method

PSO is a random search heuristic method and it generates random solutions for each execution. So it is necessary to apply Chebyshev’s theorem to PSO method results in order, to find near optimum solution range around the mean value of all obtained solutions.

Chebyshev's theorem states that for any set of numbers, the fraction of these numbers that will lie within K standard deviations of the mean can be calculated as follows:

This formula can be used also for calculation of the fraction of data between the minimum $K1$ and maximum $K_1^{|}$ values of the elements of a given data set or between , $K_2^{|}$ and $K2$ for the maximum values of the elements as follows

$$\text{Fraction of data} = 1 - \frac{1}{(K)^2} \tag{6.1}$$

Where

K	Number of the standard deviations from both sides of the mean value of the considered set of data.
$K1$	(mean – minimum value of the given data set) / Standard deviation
$K2$	(maximum value of the given data set – mean) / Standard deviation

The condition for the application of the Chebyshev's theorem is $K > 1$, which means to consider more than one standard deviation from the mean. This theorem applies to all type distributions of data.

The mean and standard deviation of any data is given by Equation (6.2) and (6.3) respectively.

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} \tag{6.2}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x - \bar{x}\right)^2} \tag{6.3}$$

Where

$\bar{x}$	Mean of any given data
σ	Standard deviation
N	Number of data's

The mean, standard deviation, and the fraction of data are calculated for data-parallel PSO solutions of the CEED problem and are given in Table 6.7, 6.9, 6.11 and 6.13 respectively. The developed PSO algorithm in Chapter 4 is run in a data-parallel way for IEEE 30 bus system. Two subcases are considered: the same power demand for all workers and different power demands to the different workers.

Subcase 1: The same power demands are sent to six different workers as given below: (Table 6.6 to 6.13)

Worker 1:	250 [MW]	Worker 2:	250 [MW]	Worker 3:	250 [MW]
Worker 4:	250 [MW]	Worker 5:	250 [MW]	Worker 6:	250 [MW]

Hence, the capability of the PSO algorithm using data-parallel calculations is investigated for the considered below four scenarios, as follows:

i. The constant power demand of 250 [MW] is sent to six Matlab workers using Min-Max price penalty factor for a number of particles in the swarm equal to 10.

ii. The constant power demand of 250 [MW] is sent to six Matlab workers using Min-Max price penalty factor for a number of particles in the swarm equal to 30.

iii. The constant power demand of 250 [MW] is sent to six Matlab workers using Max-Max price penalty factor for a number of particles in the swarm equal to 10.

iv. The constant power demand of 250 [MW] is sent to six Matlab workers using Max-Max price penalty factor for a number of particles in the swarm equal to 30.

Subcase 2: Various power demands are sent to different workers as given below: (Table 6.14 to 6.17)

Worker 1:	125 [MW]	Worker 2:	150 [MW]	Worker 3:	175 [MW]
Worker 4:	200 [MW]	Worker 5:	225 [MW]	Worker 6:	250 [MW]

v. Six different power demands are sent to six different workers using Min-Max price penalty factor for a number of particles in the swarm equal to 10.

vi. Six different power demands are sent to six different workers using Min-Max price penalty factor for a number of particles in the swarm equal to 30.

vii. Six different power demands are sent to six different workers using Max-Max price penalty factor for a number of particles in the swarm equal to 10.

viii. Six different power demands are sent to six different workers using Max-Max price penalty factor for a number of particles in the swarm equal to 30.

The data-parallel solutions of the CEED problem using the PSO method for the considered first four scenarios are given in Table 6.6 to 6.13.

Table 6.6: **Data-parallel PSO solutions of the CEED problem using the Min-Max penalty factor for 10 particles in the swarm**

Name of the worker	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	P_L [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]	Number of iterations per worker	Computation Time per worker [s]
mat32-wrk1	135.0281	42.3529	24.1501	24.1364	16.4158	13.8466	5.9299	690.1578	300.9827	817.5432	25	4.2288
mat32-wrk2	135.8212	50.1971	25.6961	13.2141	13.5214	17.9177	6.3675	691.0407	307.7463	817.6888	9	3.5710
mat32-wrk3	128.0193	47.1493	24.6530	24.5294	14.8065	16.5533	5.7108	692.6998	295.5579	817.8253	18	3.5822
mat32-wrk4	141.6731	47.0501	21.7265	20.4280	14.9374	10.6903	6.5055	686.1100	313.8531	815.8826	18	3.9623
mat32-wrk5	146.6257	45.0698	23.1155	18.5360	10.6700	12.7818	6.7988	685.1972	320.6730	816.5491	7	1.9336
mat32-wrk6	132.4733	49.4452	22.9099	23.3317	10.0836	17.8566	6.1003	690.4857	303.7451	817.3051	21	1.8457
Head node total operating time in [s]											21.1720	

Table 6.7: **Standard deviation and fraction of data calculation for the data-parallel PSO solutions of the CEED problem using the Min-Max penalty factor for 10 particles in the swarm**

Probability analysis	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	P_L [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]
Mean	136.6068	46.8774	23.7085	20.6959	13.4058	14.9411	6.2355	689.2819	307.0930	817.1324
SD	6.6330	2.8795	1.4111	4.3404	2.5260	2.9627	0.3987	2.9576	9.0790	0.7603
K1	1.2947	1.5713	1.4046	1.7238	1.3152	1.4348	1.3160	1.3811	1.2705	1.6438
K2	1.5105	1.1529	1.4085	0.8832	1.1916	1.0047	1.4128	1.1556	1.4958	0.9114
Fraction of data %	40.3396 ~ 56.1691	59.4965 ~ 24.7621	49.3116 ~ 49.5969	66.3452 ~ -28.1943	42.1884 ~ 29.5739	51.4226 ~ 0.9318	42.2609 ~ 49.9029	47.5726 ~ 25.1210	38.0511 ~ 55.3032	62.9926 ~ -20.4007

Table 6.8: **Data-parallel PSO solutions of the CEED problem using the Min-Max penalty factor for 30 particles in the swarm**

Name of the worker	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	P_L [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]	Number of iterations per worker	Computation Time per worker [s]
mat32-wrk1	133.8061	44.4590	22.1976	24.5246	16.2150	14.7381	5.9405	689.7240	300.4373	816.9681	11	9.1196
mat32-wrk2	141.1899	52.6787	21.5448	14.6853	10.1424	16.5567	6.7978	687.4895	318.2968	817.0252	37	8.6061
mat32-wrk3	134.3973	48.1086	21.3595	21.1707	16.5182	14.5583	6.1126	688.9882	302.9655	815.9464	16	9.2774
mat32-wrk4	134.0400	46.0172	21.6080	24.4891	14.1541	15.7190	6.0274	689.2289	302.0160	816.5401	16	9.0428
mat32-wrk5	130.9404	49.1524	26.3131	22.5934	11.5300	15.4409	5.9704	691.9420	301.6774	817.5288	32	8.2211
mat32-wrk6	130.5231	51.5905	21.3481	25.1292	12.0923	15.3395	6.0227	690.3952	303.3248	817.2139	4	8.3061
Head node total operating time in [s]											21.8440	

Table 6.9: **Standard deviation and fraction of data calculation for the data-parallel PSO solutions of the CEED problem using the Min-Max penalty factor for 30 particles in the swarm**

Probability analysis	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	P_L [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]
Mean	134.1495	48.6677	22.3952	22.0987	13.4420	15.3921	6.1452	689.6280	304.7863	816.8704
SD	3.8264	3.1596	1.9443	3.9204	2.6086	0.7199	0.3251	1.4896	6.6968	0.5564
K1	0.9477	1.3320	0.5385	1.8910	1.2649	1.1582	0.6297	1.4356	0.6494	1.6607
K2	1.8400	1.2695	2.0151	0.7730	1.1793	1.6177	2.0074	1.5534	2.0175	1.1833
Fraction of data %	-11.3344 ~ 70.4617	43.6404 ~ 37.9476	-244.7865 ~ 75.3725	72.0343 ~ -67.3525	37.4982 ~ 28.0906	25.4546 ~ 61.7888	-152.2309 ~ 75.1835	51.4800 ~ 58.5607	-137.1133 ~ 75.4308	63.7398 ~ 28.5842

Table 6.10: **Data-parallel PSO solutions of the CEED problem using the Max-Max penalty factor for 10 particles in the swarm**

Name of the worker	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	P_L [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]	Number of iterations per worker	Computation Time per worker [s]
mat32-wrk1	113.2991	42.6046	25.5110	27.9540	23.1848	22.2469	4.8004	706.7198	280.4285	1282.6440	18	23.5242
mat32-wrk2	121.6029	45.6087	24.8056	24.7609	22.1232	16.3728	5.2740	698.1503	287.6332	1281.1256	7	23.2108
mat32-wrk3	121.5281	45.5910	24.7866	24.8191	22.1700	16.3744	5.2693	698.2076	287.5759	1281.1184	7	23.4145
mat32-wrk4	122.1090	46.6081	22.6695	24.6816	24.0484	15.2166	5.3333	697.8389	289.2533	1282.9118	18	23.4815
mat32-wrk5	117.0564	44.8636	21.3375	22.7258	22.8967	26.2662	5.1462	705.0327	283.2695	1283.9704	11	23.0988
mat32-wrk6	113.5366	42.8675	27.2924	31.5086	22.7597	16.7967	4.7616	705.7920	283.4777	1286.6840	7	23.0676
Head node total operating time in [s]											39.6100	

Table 6.11: **Standard deviation and fraction of data calculation for the data-parallel PSO solution of the CEED problem using the Max-Max penalty factor for 10 particles in the swarm**

Probability analysis	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	P_L [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]
Mean	118.1887	44.6906	24.4004	26.0750	22.8638	18.8789	5.0975	701.9569	285.2730	1283.0757
SD	4.1233	1.6146	2.1113	3.1470	0.7139	4.3867	0.2529	4.2979	3.3889	2.0817
K1	1.1858	1.2920	1.4507	1.0643	1.0374	0.8349	1.3282	0.9581	1.4295	0.9402
K2	0.9508	1.1876	1.3698	1.7266	1.6593	1.6840	0.9324	1.1082	1.1745	1.7333
Fraction of data %	28.8879 ~ -10.6245	40.0897 ~ 29.0979	52.4846 ~ 46.7029	11.7100 ~ 66.4557	7.0804 ~ 63.6812	-43.4723 ~ 64.7382	43.3138 ~ 15.0297	-8.9281 ~ 18.5728	51.0650 ~ 27.5087	-13.1153 ~ 66.7163

Table 6.12: Data-parallel PSO solutions of the CEED problem using the Max-Max penalty factor for 30 particles in the swarm

Name of the worker	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	P_L [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]	Number of iterations per worker	Computation Time per worker [s]
mat32-wrk1	117.9292	45.1501	25.7970	23.1097	20.2772	22.8691	5.1323	702.4412	283.4405	1280.7217	35	101.5935
mat32-wrk2	119.6353	45.8059	24.8352	26.6139	18.6795	19.6412	5.2109	698.9373	286.1902	1279.9195	34	101.5935
mat32-wrk3	114.9600	48.9911	21.7455	28.1325	19.8861	21.3864	5.1017	701.6854	285.3008	1281.2860	20	101.5735
mat32-wrk4	119.9959	45.0122	23.7258	27.7159	21.4137	17.3176	5.1810	698.7622	286.6683	1280.5410	32	101.3493
mat32-wrk5	113.0798	51.6118	21.6592	25.6241	23.2214	19.8718	5.0680	703.4137	285.3244	1282.8801	37	101.2399
mat32-wrk6	126.6020	43.7145	24.9202	25.3957	16.7838	18.1017	5.5180	694.7909	291.4346	1283.0721	25	101.2669
Head node total operating time in [s]											132.0940	

Table 6.13: Standard deviation and fraction of data calculation for the data-parallel PSO solution of the CEED problem using the Max-Max penalty factor for 30 particles in the swarm

Probability analysis	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	P_L [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]
Mean	118.7004	46.7143	23.7805	26.0986	20.0436	19.8646	5.2020	700.0051	286.3931	1281.4034
SD	4.7152	2.9778	1.7391	1.8260	2.2155	2.0488	0.1632	3.1702	2.7051	1.2953
K1	1.1920	1.0074	1.2198	1.6369	1.4714	1.2432	0.8211	1.6448	1.0915	1.1456
K2	1.6758	1.6447	1.1595	1.1139	1.4343	1.4665	1.9363	1.0752	1.8637	1.2883
Fraction of data %	29.6223 ~ 64.3901	1.4614 ~ 63.0306	32.7883 ~ 25.6206	62.6768 ~ 19.3986	53.8085 ~ 51.3939	35.2945 ~ 53.49981	-48.3306 ~ 73.3273	63.0344 ~ 13.4989	16.0622 ~ 71.2096	23.8041 ~ 39.7462

Figures 6.25 and 6.26 show the comparison of active power of P1 and CEED fuel cost with 6 workers. Figures 6.27 and 6.28 show the comparison of computation time needed to obtain the near optimum solution of PSO method using Min-Max and Max-Max penalty factors for two different selections of the number of particles in the swarm of 10 and 30 and with the same power demand assigned to each individual worker.

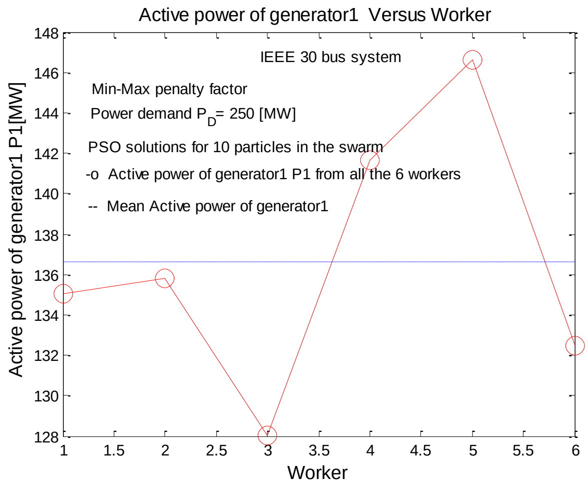


Figure 6.25: Comparison of the active power P1 with worker for the PSO solution using the Min-Max penalty factor

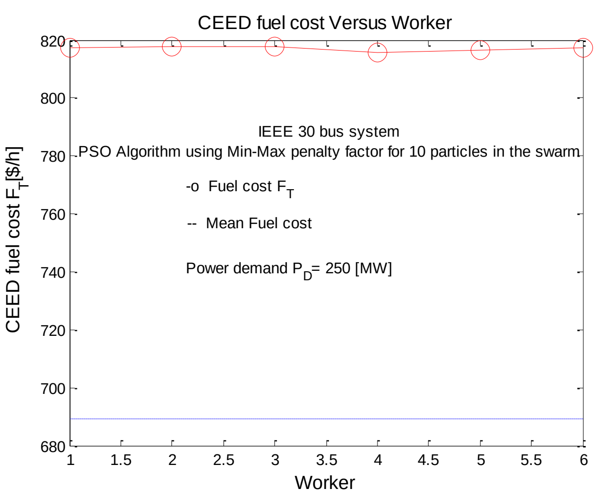


Figure 6.26: Comparison of the CEED fuel cost F_T with worker for the PSO solution using the Min-Max penalty factor

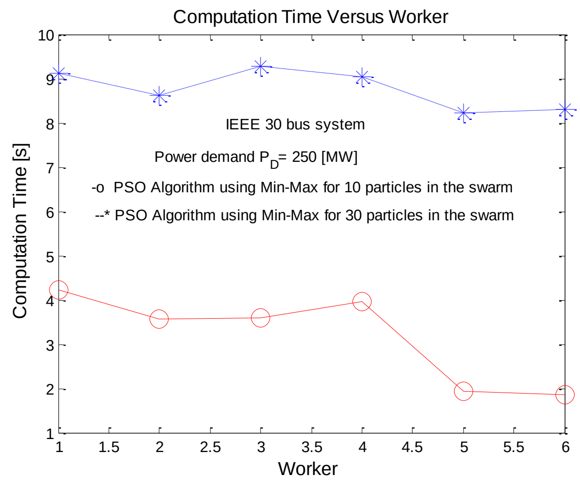


Figure 6.27: Comparison of the computation time for the PSO solution using the Min-Max penalty factor for two different numbers of particles of 10 and 30 and the same power demand assigned to every workers

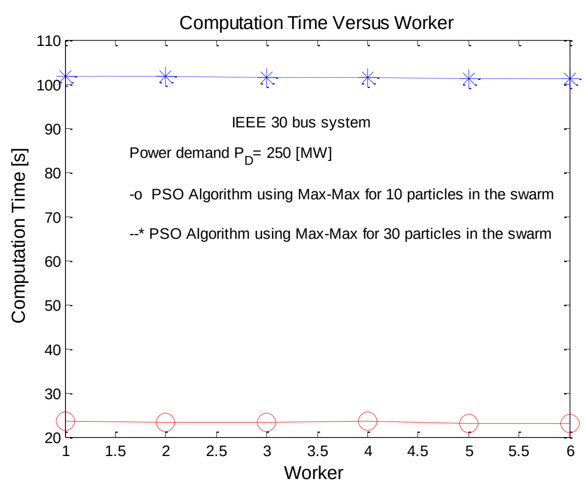


Figure 6.28: Comparison of the computation time for the PSO solution using the Max-Max penalty factor for two different numbers of particles of 10 and 30 and the same power demand assigned to every workers

6.4.1 Discussion on the data-parallel computing solutions of the CEED problem using PSO algorithm with the same power demand of 250 MW in all the workers

In Table 6.6 to 6.13, workers 1 to 6 were assigned to have constant power demand of 250 [MW] to each workers and two different numbers of particles of 10 and 30 were used to compare the obtained solutions. The data-parallel solutions of the CEED problem using the PSO algorithm are not the same as the sequential ones, since metaheuristic PSO algorithm generates random values for each iteration. The computational time for the CEED problem solution using the metaheuristic method (PSO) depends on the initial selection of the number of particles in the swarm as given in Tables 6.6 to 6.13. It is proven that if the initial selection of the number of the particles in the swarm is less, it requires less computation time to obtain the near global solution, otherwise the bigger number of particles in the swarm needs more computational time as shown in Figure 6.27 and 6.28. When initial number of particles in the swarm is 10, computational time needed to obtain PSO solution using the Min-Max penalty factor is less in comparison with the solution for 30 particles in the swarm. It shows that the solution obtained with the PSO algorithm is influenced by the initial selection of the number of particles in the swarm.

PSO algorithm generates random solutions in all the 6 workers. The Matlab cluster of computers is used to generate different PSO solution in 6 workers for the same power demand P_D 250 [MW] which is given in Table 6.7. The mean and standard deviation are used to find the fraction of the data set based on the Chebyshev's theorem. In Table 6.7, PSO solution for the IEEE 30 bus system using Min-Max penalty factor is given. It has a mean and standard deviation of 136.60 and 6.63 respectively. The value of *K* lies between 1.2946 and 1.5104 refer Table 6.7.

- For K1=1.2946, then 40.33% of the active power of generator 1 lies between 128.01 to 145.19 [MW]
- For K2=1.51, then 56.16% of the active power of generator 1 lies between 126.58 to 146.62 [MW].

It is proven that the fraction of data of the active power of generator 1 which is calculated using Chebyshev's theorem for the PSO solution based on Min-Max penalty factor is close to the mean value of the active power of generator 1.

The PSO algorithm implemented for solution of the CEED problem for various values of the power demand is applied in a data-parallel way using the Min-Max and Max-Max penalty factors and 10 or 30 particles in the swarm (scenarios 5 to 8). The results are given in Tables 6.14 to 6.17.

Table 6.14: **Data-parallel PSO solutions of the CEED problem using a Min-Max penalty factor for 10 particles in the swarm and with different power demands assigned to every workers**

Name of the worker	P_D [MW]	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	P_L [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]	Number of iterations per worker	Computation Time per worker [s]
mat32-wrk1	125	50.0000	26.1487	17.5301	12.8406	13.0927	12.5408	1.1390	332.0623	153.1798	404.8368	21	23.3116
mat32-wrk2	150	64.9339	25.8405	18.8241	13.4284	10.9185	17.7294	1.6747	385.4845	162.4775	462.1604	18	23.4507
mat32-wrk3	175	92.7815	22.0565	24.2576	10.9404	13.0204	14.6567	2.7131	455.1861	186.6582	539.9975	2	23.5443
mat32-wrk4	200	111.0379	29.0612	18.3209	18.2135	15.1143	12.0597	3.8074	526.0763	219.5829	623.6578	2	23.6474
mat32-wrk5	225	116.3730	47.3908	21.5058	13.9112	14.9641	15.7271	4.8720	606.8608	256.9177	715.8499	17	23.1760
mat32-wrk6	250	143.4226	46.8939	20.0528	12.3018	18.2169	15.7907	6.6788	687.5114	315.1698	818.0181	20	23.1545
Head node total operating time in [S]												41.0630	

Table 6.15: **Data-parallel PSO solutions of the CEED problem using a Min-Max penalty factor for 30 particles in the swarm and with different power demands assigned to every workers**

Name of the worker	P_D [MW]	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	P_L [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]	Number of iterations per worker	Computation Time per worker [s]
mat32-wrk1	125	50.0000	18.8701	20.2577	9.8370	12.8358	21.0525	1.1164	344.1766	155.4164	419.3112	1	108.1828
mat32-wrk2	150	64.5041	27.7569	18.5799	14.7255	12.0642	14.0306	1.6612	383.3349	162.5006	459.6290	15	108.7903
mat32-wrk3	175	88.0883	27.3660	21.0198	13.9297	15.0685	12.1257	2.5979	452.7297	184.1196	536.9326	7	108.6674
mat32-wrk4	200	104.1440	38.8926	19.5248	12.5571	13.7804	14.8899	3.7888	525.2711	217.9137	620.6234	15	108.3768
mat32-wrk5	225	127.5704	45.1852	19.3856	13.1075	11.6371	13.5386	5.4244	601.3734	268.2408	713.4549	4	101.7679
mat32-wrk6	250	138.5568	50.2854	23.1532	13.3596	16.5351	14.5967	6.4870	688.4791	310.9740	816.4723	20	101.9997
Head node total operating time in [S]												126.7660	

Table 6.16: **Data-parallel PSO solutions of the CEED problem using a Max-Max penalty factor for 10 particles in the swarm and with different power demands assigned to every workers**

Name of the worker	P_D [MW]	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	P_L [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]	Number of iterations per worker	Computation Time per worker [s]
mat32-wrk1	125	50.0000	22.2047	21.7644	13.9499	12.0790	25.9915	1.2901	389.9397	167.9605	755.4505	2	23.5839
mat32-wrk2	150	55.5870	33.9910	21.7091	12.1905	12.0151	16.0599	1.5525	388.7696	168.6214	748.6597	10	23.3688
mat32-wrk3	175	75.7889	26.2902	19.0348	17.8501	19.9147	18.2945	2.1731	466.4808	182.6376	855.9366	9	23.5126
mat32-wrk4	200	108.0520	35.6919	21.1763	12.9948	10.4145	15.5774	3.9068	524.2388	219.9189	975.1413	13	23.5691
mat32-wrk5	225	117.3134	37.3643	16.9988	25.5600	17.8374	14.5107	4.5847	609.8953	253.3740	1124.5010	25	23.1462
mat32-wrk6	250	125.9431	44.7433	22.8955	21.3733	17.5806	23.0478	5.5835	696.5256	290.1646	1283.0978	12	22.9976
Head node total operating time in [S]												39.5160	

Table 6.17: **Data-parallel PSO solutions of the CEED problem using a Max-Max penalty factor for 30 particles in the swarm and with different power demands assigned to every workers**

Name of the worker	P_D [MW]	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	P_L [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]	Number of iterations per worker	Computation Time per worker [s]
mat32-wrk1	125	50.0000	25.3480	15.2514	21.0602	14.9676	12.5719	1.1775	359.0826	160.1855	702.6658	39	101.8074
mat32-wrk2	150	66.1286	21.7591	18.9441	13.7986	16.1641	14.8096	1.6040	387.7633	161.4122	734.7592	30	103.1258
mat32-wrk3	175	83.4624	33.4963	20.6183	10.5264	16.3644	13.1054	2.5731	453.0214	185.0358	841.7756	32	102.8696
mat32-wrk4	200	103.3159	34.0308	20.2883	17.2386	11.7279	17.0248	3.6262	527.9411	215.2338	972.0374	12	102.2761
mat32-wrk5	225	113.0392	42.7653	22.8081	15.3697	17.4873	18.0817	4.5513	610.6085	250.1517	1120.7817	5	101.0463
mat32-wrk6	250	116.4857	44.2894	24.1416	27.2434	19.5923	23.2882	5.0407	702.7840	283.0386	1280.7349	7	100.7420
Head node total operating time in [S]												121.1720	

Figures 6.29 and 6.30 show the comparison of the computation time needed to obtain the near optimum solution of the CEED problem using the PSO method based on Min-Max and Max-Max penalty factors for two different selections of the number of particles in the swarm of 10 and 30 and with different power demands assigned to every workers.

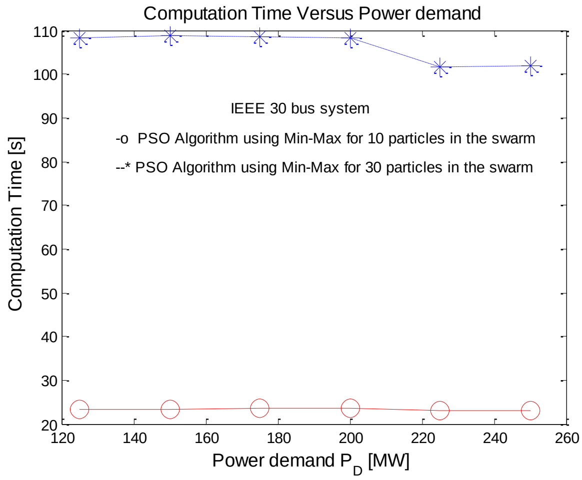


Figure 6.29: Comparison of the Computation time for the PSO solution of the CEED problem using the Min-Max penalty factor and different power demands assigned to every workers

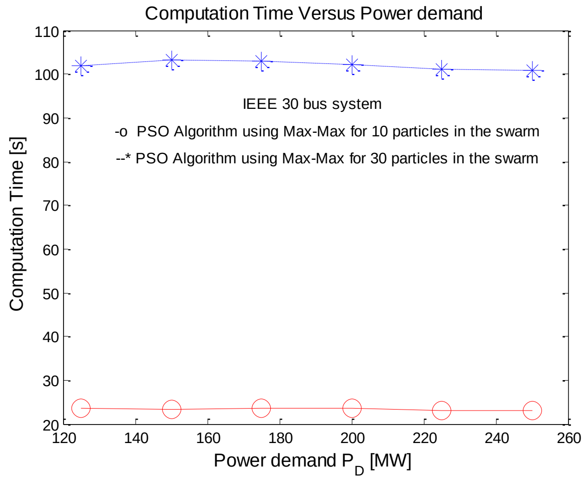


Figure 6.30: Comparison of the computation time for the PSO solution of the CEED problem using the Max-Max penalty factor and different power demands assigned to every workers

6.4.2 Discussion on the data-parallelly computed solutions for the CEED problem using the PSO algorithm with different power demands assigned to each individual worker

In Tables 6.14 to 6.17, workers 1 to 6 were assigned with different power demands of 125 to 250 [MW], with an incremental of 25 [MW]. Two different numbers of particles in the swarm of 10 and 30 are used to compare the parallel PSO solution using Min-Max and Max-Max penalty factors. The computational time of the metaheuristic (PSO) solution depends on the initial selection of the number of particles in the swarm. It can be seen from the Figures 6.29 and 6.30 that the computation time of the PSO algorithm is dependent on the initial selection of the number of particles in the swarm. It is proven that if the initial number of particles in the swarm is less, it requires less computation time to obtain the near global solution, otherwise bigger number of particles in the swarm needs more computational time. When initial number of particle is 10, computational time needed to obtain the PSO solution using Min-Max penalty factor is less in comparison with the solution for 30 particles. It shows that the solution of the PSO algorithm is influenced by the selection of the number of particles in the swarm.

6.5 Comparison of the results from the sequential and data-parallel solutions

The results of sequential program are compared with the results of data-parallel one. Calculation of the CEED problem using the two developed Lagrange's (classical) and PSO (heuristic) methods are applied in a data-parallel way. The solutions of the sequential and data-parallel methods are the same for the Lagrange's but different for the PSO method. The comparison of computation time of sequential and data-parallel solution of Lagrange's and PSO methods of the CEED problem is given in Table 6.5. The conclusion is that data-parallel solutions need less computational time in comparison with the sequential solutions.

6.6 Conclusion

This chapter describes the programs for CEED problem solutions in the Parallel Computing Toolbox. The input and output variables of the programs and the used functions and commands are described. The Results from calculations with the Matlab sequential program and the data-parallel calculations are described. The results of a sequential program (Krishnamurthy and Tzoneva, 2013) are compared with the results of a data-parallel program problem using the two developed Lagrange's (Classical) and PSO (Metaheuristic) algorithms. The solutions of the sequential and data-parallel methods are the same for the Lagrange's but different for the PSO method, hence metaheuristic PSO generates different solution for each simulation run. The comparison of the computation time of sequential and data-parallel solution of the Lagrange's and the PSO method are based on Min-Max and Max-Max penalty factors. It is clear that, less computation time is needed for data-parallel calculation of the CEED problem using Lagrange's method in comparison with the PSO data-parallel solutions. The fuel cost values are less for the Min-Max penalty factor in comparison with these for the Max-Max one.

Multi-area economic dispatch problem formulation and solution using Lagrange's decomposition-coordinating method is described in Chapter 7.

CHAPTER SEVEN

MULTI AREA ECONOMIC EMISSION DISPATCH PROBLEM FORMULATION AND SOLUTION BY A DECOMPOSITION-COORDINATING METHOD

7.1 Introduction

Large interconnected power systems are usually decomposed into areas or zones based on criteria, such as the size of the electric power system, network topology and geographical location. Multi Area Economic Emission Dispatch (MAEED) problem is an optimisation task in power system operation for allocating amount of generation to the committed units within these areas. Its objective is to minimize the fuel cost subject to the power balance, generator limit, transmission lines and tie-line constraints. The solution of the MAEED problem determines the amount of power that can be economically generated in the areas and transferred to other areas if it is needed without violating tie-line capacity constraints and the whole power network constraints.

The solutions of the MAEED problem in the conditions of deregulation are difficult, due to the model size, nonlinearity, and interconnections, and require intensive computations in real-time. High-performance computing (HPC) gives possibilities for reduction of the problem complexity and the time for calculation by the use of parallel processing for running advanced application programs efficiently, reliably and quickly. A cluster of computers working in Matlab software environment is used to implement the optimisation algorithms. Parallelization of the solution is done through decomposition of the MAEED problem according to the power system interconnected areas and coordination of the obtained solutions for every area by a coordinator. Classical (Lagrange) decomposition-coordinating method for HPC is developed and implemented using IEEE benchmark power system models.

This chapter formulates the MAEED problem and develops Lagrange's decomposition coordination method and software to solve the MAEED problem. The Matlab parallel computing toolbox setup procedure is described, how to start the workers from the head node, and test the configuration validation to find resource, distributed job, parallel job and Matlab pool. IEEE 4 area 40 generators system without considering transmission line losses and 4 area 15 generator system by considering transmission line losses are used to test the developed algorithm in the Cape Peninsula University of Technology (CPUT's) research center Real Time Distributed Systems (RTDS) Cluster Computer laboratory.

This chapter formulate MAEED problem solution in part 7.2, Lagrange's decomposition-coordinating method and algorithm for solution of the MAEED problem is given in part 7.3 and 7.4 respectively, single area and multi-area CEED problem solutions for 4 area 40 generator system and 4 area 15 generator system are given in part 7.5 and 7.6 respectively. Profile viewer of the Matlab distributed computing engine is described in part 7.7, the discussion and conclusion on the simulation results are given in part 7.8 and 7.9 respectively.

7.2 Formulation of the MAEED problem solution

Implementation of the solution of the multi-area dispatch problem leads to dispatch of the generator's power within multiple areas. Every area has its own set of generators. The areas are interconnected by tie-lines as shown in Figure 7.1. Single criterion or bi-criteria dispatch problems can be formulated for every area. The solutions of these problems determine the optimal power to be produced by the generators in every area and the optimal values of the power transferred between the areas through the tie-lines. The dispatch problem for the whole power system is a multi-criteria one.

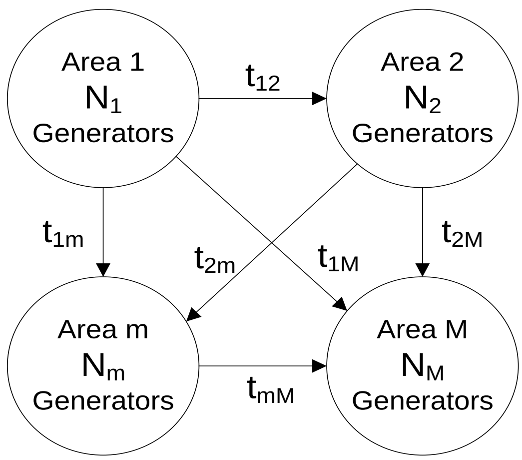


Figure 7.1: **Model of a Multi-Area power system with tie-line power transfer**

The mathematical formulation of the MAEED problem is as follows:

Find the vector of the real power P and the vector of the power transmitted through the tie-lines P_T such that the operational cost for power production

$$F_C(P) = \sum_{m=1}^{M} \sum_{n=1}^{N_m} \left(a_{mn} P_{mn}^2 + b_{mn} P_{mn} + c_{mn} \right) \tag{7.1}$$

where

M Number of the interconnected areas

N_m Number of generators belonging to area m and committed to the power production in this area

a_{mn},b_{mn},c_{mn} Cost coefficients of the n^{th} generator in the m^{th} area

P_{mn} Real power produced by the n^{th} generator in the m^{th} area

$P_m = \left[P_{m1} \; P_{m2} \; P_{m3} \ldots\ldots P_{mN_m} \right]^T$ Vector of the real power produced in the m^{th} area, and $m = \overline{1,M}$

$P = \left[P_1 \; P_2 \; P_3 \ldots\ldots P_M \right]^T$ Vector of the real power produced in the whole power system

The operational cost for transmission of the power through the tie-lines is given as:

$$F_T\left(P_T\right) = \sum_{m=1}^{M} \sum_{\substack{j=1 \\ j \neq m}}^{M} \left(q_{mj} P_{Tmj} + q_{jm} P_{Tjm} \right) \tag{7.2}$$

Where

P_{Tmj} Tie-line power flow from area m to area j

q_{mj} Transmission coefficient for the cost of transmission of the power from area m to area j.

$P_{Tm} = \left[P_{Tm,m+1} \; P_{Tm,m+2} \; P_{Tm,m+3} \ldots\ldots P_{Tm,M} \right]^T, m = \overline{1,M-1}$ Vector of the power transmission between the m^{th} area and all other area

$P_T = \left[P_{T1} \; P_{T2} \; P_{T3} \ldots\ldots P_{T,M-1} \right]^T$ Vector of the power transmission between all areas.

Equation (7.1) and (7.2) are minimised under the following constraints:

i. Minimum and maximum produced active power for every generator

$$P_{mn,\min} \leq P_{mn} \leq P_{mn,\max}, \quad m = \overline{1,M}, \quad n = \overline{1,N_m} \tag{7.3}$$

where

$P_{mn,\min}$ and $P_{mn,\max}$ Minimum and maximum power that can be produced by the n^{th} generator in the m^{th} area

ii. Minimum and maximum active power sent through the tie-lines

$$P_{Tmj,\min} \leq P_{Tmj} \leq P_{Tmj,\max} \tag{7.4}$$

These limits are valid for the two directions of the power flow and can be written as

$$\begin{aligned} P_{T\min} &\leq P_{Tmj} \leq P_{T\max} \quad m = \overline{1,M}, \; j = \overline{1,M}, \; j \neq m \\ P_{T\min} &\leq P_{Tjm} \leq P_{T\max} \quad m = \overline{1,M}, \; j = \overline{1,M}, \; j \neq m \end{aligned} \tag{7.5}$$

iii. Power balance constraint

Balance between the power production and the power demand for the m^{th} area and for the whole system is given by Equation (6.6),

$$\sum_{n=1}^{N_m} P_{mn} = P_{Dm} + P_{Lm} + \sum_{\substack{j=1 \\ j \neq m}}^{M} \left[P_{Tmj} - \left(1 - \rho_{jm}\right) P_{Tjm} \right], \qquad m = \overline{1, M} \tag{7.6}$$

Where

$$P_{Lm} = \sum_{n=1}^{N_m} \sum_{r=1}^{N_m} \left(P_{mn} B_{mnr} P_{mr} \right) + \sum_{n=1}^{N_m} B_{0n} P_{mn} + B_{m00}, \quad m = \overline{1, M} \tag{7.7}$$

$$\sum_{n=1}^{N_m} P_{mn} = \begin{bmatrix} P_{Dm} + \sum_{n=1}^{N_m} \sum_{r=1}^{N_m} \left(P_{mn} B_{mnr} P_{mr} \right) + \sum_{n=1}^{N_m} B_{m0n} P_{mn} + \\ + B_{m00} + \sum_{\substack{j=1 \\ j \neq m}}^{M} \left[P_{Tmj} - \left(1 - \rho_{jm}\right) P_{Tjm} \right] \end{bmatrix}, \qquad m = \overline{1, M} \tag{7.8}$$

Where

$B_{mnr}, B_{m0n}, B_{m00}$	Transmission loss coefficients of the interconnected power system
ρ_{jm}	Fractional loss rate from area j to area m
P_{Tmj}	Power flow from area m to area j and
P_{Tjm}	Power flow from area j to area m

The total operational cost is the sum of the generation cost and of the cost for transmission of the power between the areas. An additional criterion expressing minimisation of the pollutant emission can be also considered. This criterion refers only to the power generation. The tie-lines are not considered in this case because the transmission of power does not create chemical pollution. The mathematical expression of the criterion for the interconnected power system is:

$$F_e = \sum_{m=1}^{M} \sum_{n=1}^{N_m} \left(d_{mn} P_{mn}^2 + e_{mn} P_{mn} + f_{mn} \right) \tag{7.8a}$$

Where

d_{mn}, e_{mn}, f_{mn}	Emission coefficients of the n^{th} generator in the m^{th} area

The multi-area combined economic emission dispatch fuel cost is given by Equation 6.8b

$$F_T = \sum_{m=1}^{M} \sum_{n=1}^{N_m} \left[\left(a_{mn} P_{mn}^2 + b_{mn} P_{mn} + C_{mn} \right) + h_{mn} \left(d_{mn} P_{mn}^2 + e_{mn} P_{mn} + f_{mn} \right) \right] \tag{7.8b}$$

Unknown variables are the values of the active power P_{mn} .

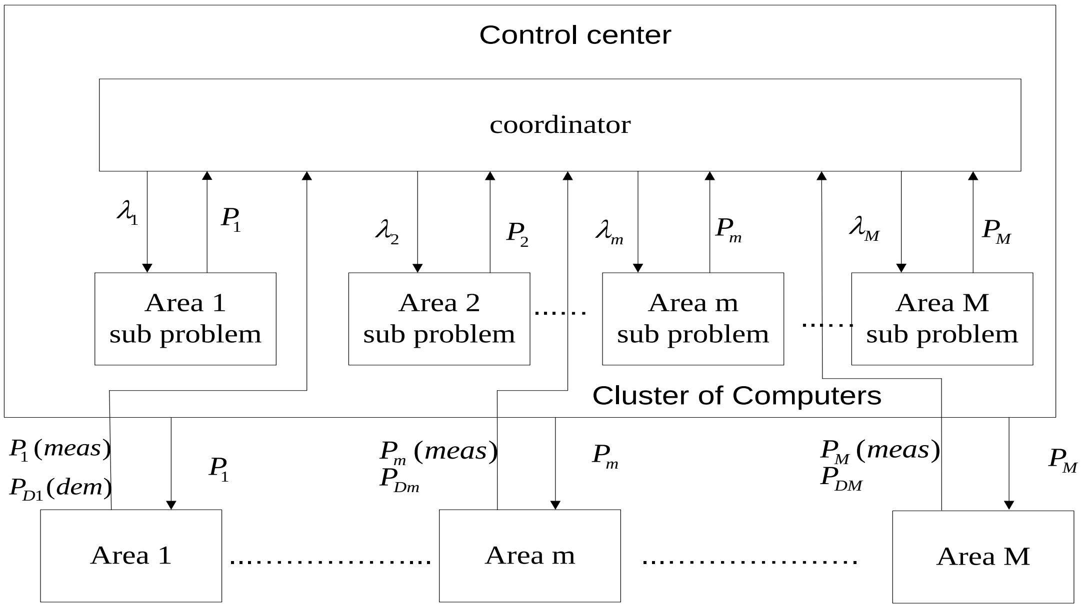
The formulated problem given by Equation (7.1 to 7.8) is characterised as:

- A bi-criteria optimisation problem dispatch problem for every area
- The area are interconnected by tie-lines
- The optimisation dispatch problem for the interconnected power system is Multicriteria
- Dimension of the interconnected problem depends on the number of areas, and tie-lines.

Calculation of the solution of the multicriteria interconnected problem is difficult and time consuming. If the solution has to be done often and in real-time, then better computational approaches can be proposed. One of them is a parallel computing of the area's sub-problems and coordination of the obtained solution using a High Performance Parallel Computing (HPPC) environment (Cluster of computers). This approach requires:

i. A decomposition-coordinating method to be developed in order to obtain an algorithm for parallel calculation.
ii. Software development based on the above algorithm allowing allocation of the separate sub-problems to the structure of the HPPC environment – A Cluster of Computers.
iii. Implementation of the developed software in the HPPC environment, investigations of the capabilities of the developed algorithm.
iv. Evaluation and verification of the obtained solution by comparison of the sequential and parallel ones

The method is based on classical Lagrange’s optimisation (Okada and Asano, 1995), (Danaraj and Gajendran, 2005), and (Hemamalini and Simon, 2009). The literature review found that the classical Lagrange's method has been used till now only for sequential solution of the MAEED problem (Yingvivatanapong et al., 2008), and (Yu and Hang, 2010). The thesis introduces a parallel solution of the MAEED problem by considering two-level calculation structure, Figure 7.2, where the initial optimisation problem is decomposed into sub-problems (for every area one), solved on the first level, and the obtained solutions are coordinated by a coordinating sub-problem at the second level - in order to obtain the global solution of the whole problem.



where : λ_1 – coordinating variable; (meas) - measured; (dem) - demand

Figure 7. 2: **Real-time implementation structure of the MAEED problem solution**

Implementation of the MAEED problem in real-time is done in the following way, Figure 7.2: All data from the separate areas are sent using the communication system to the main control center where the Cluster of computers is located. The problem is solved and the optimal solutions are sent to the areas to be used for control of the generators power production. This scenario can be repeated through some selected periods of time, for example every hour.

The software for parallel calculation is developed for the following IEEE bench mark models:

(i) 4 areas: 4 generators in each area and

(ii) 4 areas: 3 generators in each area.

Figure 7.3 shows the process block diagram of the MAEED problem solution which includes problem formulation, algorithm and software development and task-parallel implementation of the calculations.

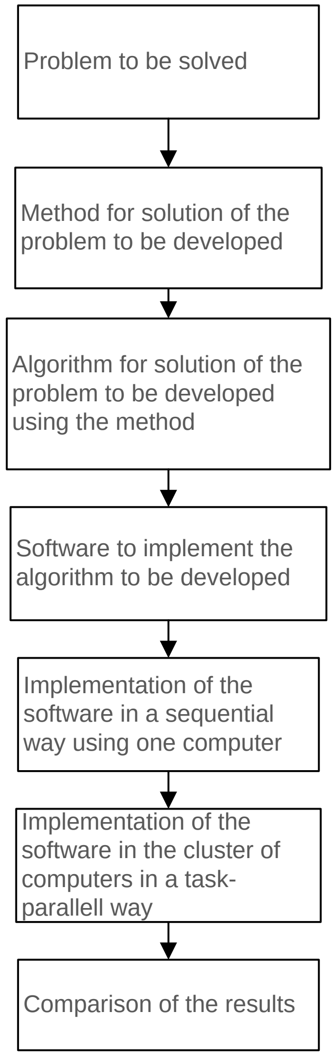


Figure 7.3: **The process block diagram for the MAEED problem solution**

7.3 Development of Lagrange's decomposition-coordinating method for solution of the MAEED problem

Development of a Lagrange's decomposition-coordinating algorithm for solution of the multi-area combined economic and emission dispatch problem is described in this section. Development of the method is based on construction of a function of Lagrange for the multi-area dispatch problem.

The function of Lagrange's is formulated as:

$$L=\left[\begin{aligned}&\sum_{m=1}^{M}\sum_{n=1}^{N_m}\left(a_{mn}P_{mn}^2+b_{mn}P_{mn}+C_{mn}\right)+\sum_{m=1}^{M}\sum_{n=1}^{N_m}h_{mn}\left(d_{mn}P_{mn}^2+e_{mn}P_{mn}+f_{mn}\right)+\sum_{m=1}^{M}\sum_{\substack{j=1\\ j\neq m}}^{M}\left(q_{mj}P_{Tmj}+q_{jm}P_{Tjm}\right)+\\&+\sum_{m=1}^{M}\left\{\lambda_m\left\{-\sum_{n=1}^{N_m}P_{mn}+P_{Dm}+\sum_{n=1}^{N_m}\sum_{r=1}^{N_m}P_{mn}B_{mnr}P_{mr}+\sum_{n=1}^{N_m}B_{m0n}P_{mn}+B_{m00}+\sum_{\substack{j=1\\ j\neq m}}^{M}\left[P_{Tmj}-\left(1-\rho_{jm}\right)P_{Tjm}\right]\right\}\right\}\end{aligned}\right] \quad (7.9)$$

Where:

$\lambda=\left(\lambda_1 \quad \lambda_2 \quad ....\lambda_m....\lambda_M\right)$ Vector of the Lagrange's multipliers.

h_{mn} Price penalty factor for every area

It is necessary to find the values of P_{mn}, P_{Tmj}, P_{Tjm} *and* λ such that the function of Lagrange has an optimum (minimum) according to P_{mn}, P_{Tmj}, P_{Tjm} , and maximum according to ***λ,*** under the constraints.

$$P_{mn,\min}\le P_{mn}\le P_{mn,\max}, m=\overline{1,M}, n=\overline{1,N_m} \quad (7.10)$$

$$P_{Tm,\min}\le P_{Tmj}\le P_{Tm,\max}, m=\overline{1,M}, j=\overline{1,M}, j\neq m \quad (7.11)$$

$$P_{Tm,\min}\le P_{Tjm}\le P_{Tm,\max}, m=\overline{1,M}, j=\overline{1,M}, j\neq m \quad (7.12)$$

The optimal solution is found on the basis of the necessary conditions for optimality, as follows:

$$\frac{\partial L}{\partial P_{mn}}=2a_{mn}P_{mn}+b_{mn}+2h_{mn}d_{mn}P_{mn}+e_{mn}+\lambda_m\left\{-1+2\sum_{r=1}^{N_m}B_{mnr}P_{mr}+B_{m0n}\right\}=0 \quad \begin{matrix}m=\overline{1,M}\\ n=\overline{1,N_m}\end{matrix} \quad (7.13)$$

$$\frac{\partial L}{\partial P_{Tmj}}=q_{mj}+\lambda_m=0=e_{pTmj}, m=\overline{1,M}, j=\overline{1,M}, j\neq m \quad (7.14)$$

$$\frac{\partial L}{\partial P_{Tjm}}=q_{jm}-\lambda_m\left(1-\rho_{jm}\right)=0=e_{pTjm}, m=\overline{1,M}, j=\overline{1,M}, j\neq m \quad (7.15)$$

$$\frac{\partial L}{\partial \lambda_m}=\left\{-\sum_{n=1}^{N_m}P_{mn}+P_{Dm}+\sum_{n=1}^{N_m}\sum_{r=1}^{N_m}P_{mn}B_{mnr}P_{mr}+\sum_{n=1}^{N_m}B_{m0n}P_{mn}+B_{m00}+\sum_{\substack{j=1\\ j\neq m}}^{M}\left[P_{Tmj}-\left(1-\rho_{jm}\right)P_{Tjm}\right]\right\} \quad (7.16)$$

where $m=\overline{1,M}, j=\overline{1,M}, j\neq m$

Solution of the system of Equations (7.13) – (7.16) can be achieved in a parallel way using decomposition approach based on selection of coordinating variables. These are selected to be $\lambda_m, m = \overline{1, M}$

It is supposed that the values of the coordinating variables are given by a coordinator in a two level hierarchical structure of calculations as follows:

$$\lambda_m = \lambda_m^l, m = \overline{1, M} \qquad (7.17)$$

Where

l Index of iterations

Then the Equation (7.13) can be solved in a decentralized way on a first level of the calculation structure and the Equations (7.14) to (7.16) can be solved on the second level.

Solution of equation (7.13) is as follows:

$$\frac{\partial L}{\partial P_{mn}} = 2a_{mn}P_{mn} + 2\lambda_m B_{mnn}P_{mn} + b_{mn} + \lambda_m \left\{ 2\sum_{\substack{r=1 \\ r \neq n}}^{N_m} B_{mnr}P_{mr} + B_{m0n} - 1 \right\} = 0 \qquad (7.18)$$

$$\frac{\partial L}{\partial P_{mn}} = \left(\frac{a_{mn}}{\lambda_m} + B_{mnn} \right) P_{mn} + \sum_{\substack{r=1 \\ r \neq n}}^{N_m} B_{mnr}P_{mr} + \frac{1}{2}\left(\frac{b_{mn}}{\lambda_m} + B_{m0n} - 1 \right) = 0 \qquad (7.19)$$

From here

$$\left(\frac{a_{mn} + h_{mn}d_{mn}}{\lambda_m} + B_{mnn} \right) P_{mn} + \sum_{\substack{r=1 \\ r \neq n}}^{N_m} B_{mnr}P_{mr} = \frac{1}{2}\left(1 - \frac{b_{mn} + h_{mn}e_{mn}}{\lambda_m} - B_{m0n} \right), \begin{matrix} m = \overline{1, M} \\ n = \overline{1, N_m} \end{matrix} \qquad (7.20)$$

Equation (7.20) can be written in a vector matrix form as

$$\begin{bmatrix} \frac{a_{m1} + h_{m1}d_{m1}}{\lambda_m} + B_{m11} & B_{m12} & \ldots\ldots & B_{m1N_m} \\ B_{m21} & \frac{a_{m2} + h_{m2}d_{m2}}{\lambda_m} + B_{m22} & \ldots\ldots & B_{m2N_m} \\ \vdots & \vdots & \vdots & \vdots \\ B_{mN_m1} & B_{mN_m2} & \ldots\ldots & \frac{a_{mN_m} + h_m d_m}{\lambda_m} + B_{nN_mN_m} \end{bmatrix} \begin{bmatrix} P_{m1} \\ P_{m2} \\ \vdots \\ P_{mN_m} \end{bmatrix} = \ldots\ldots\ldots$$

$$\ldots\ldots = \frac{1}{2} \begin{bmatrix} 1 - \frac{b_{m1} + h_{m1}e_{m1}}{\lambda_m} - B_{m01} \\ 1 - \frac{b_{m2} + h_{m2}e_{m2}}{\lambda_m} - B_{m02} \\ \vdots \\ 1 - \frac{b_{mN_m} + h_m e_m}{\lambda_m} - B_{n0N_m} \end{bmatrix}, m = \overline{1, M}$$

(7.21)

In a short form it can be written as

$$E_m P_m = D_m, m = \overline{1,M} \qquad (7.22)$$

If the value of $\lambda_m, m = \overline{1,M}$ is known, then the solution for the active power of the m^{th} area is

$$P_m = E_m \setminus D_m \qquad (7.23)$$

Solution of the equations (7.14) and (7.15) can be done by gradient procedures as follows:

a) Initial values of $P_{Tmj}^{q_m}$ *and* $P_{Tjm}^{q_m}$ are guessed , $q_m = l, q_m = \overline{l, k_m}$

$$P_{Tmj} = P_{Tmj}^{q_m} \text{ and } P_{Tjm} = P_{Tjm}^{q_m}$$

Where

q_m Index of the gradient procedure in the m^{th} area.

b) The improved values of the tie-lines power are

$$P_{Tmj}^{q_m+1} = P_{Tmj}^{q_m} - \alpha_{Tmj}\, e_{PTmj}^{q_m} \qquad (7.24)$$

where $e_{PTmj}^{q_m}$ is given by eq.(7.14), $m = \overline{1,M}$

$$P_{Tjm}^{q_m+1} = P_{Tjm}^{q_m} - \alpha_{Tjm}\, e_{PTjm}^{q_m} \qquad (7.25)$$

where $e_{PTjm}^{q_m}$ is given by eq.(7.14) , $m = \overline{1,M}$

The calculation of the values of the $P_{Tmj}^{q_m+1}$ *and* $P_{Tjm}^{q_m+1}$ stops when

$$e_{PTmj}^{q_m} \le \varepsilon_1 \text{ and } e_{PTjm}^{q_m} \le \varepsilon_2 \qquad (7.26)$$

Where

ε_1 *and* ε_2 Very small positive numbers.

Solutions (7.23), (7.24) and (7.25) depend on $\lambda_m, m = \overline{1,M}$.

When the optimal value of λ_m is obtained, the optimal values of P_{mn}, P_{Tmj} *and* P_{Tjm} will reach their optimum.

A gradient procedure is used to calculate the optimal value of $\lambda_m, m = \overline{1,M}$, as follows:

$$\lambda_m^{l+1} = \lambda_m^{l} + \alpha_{m\lambda} e_{m\lambda}, m = \overline{1,M} \qquad (7.27)$$

Where α_λ is the step of the gradient procedure and e_λ is given by equation (7.16).

The gradient procedure on the second levels stops when $e_\lambda \le \varepsilon_3,\ \varepsilon_3 \ge 0$ (7.28)

The hierarchical structure is

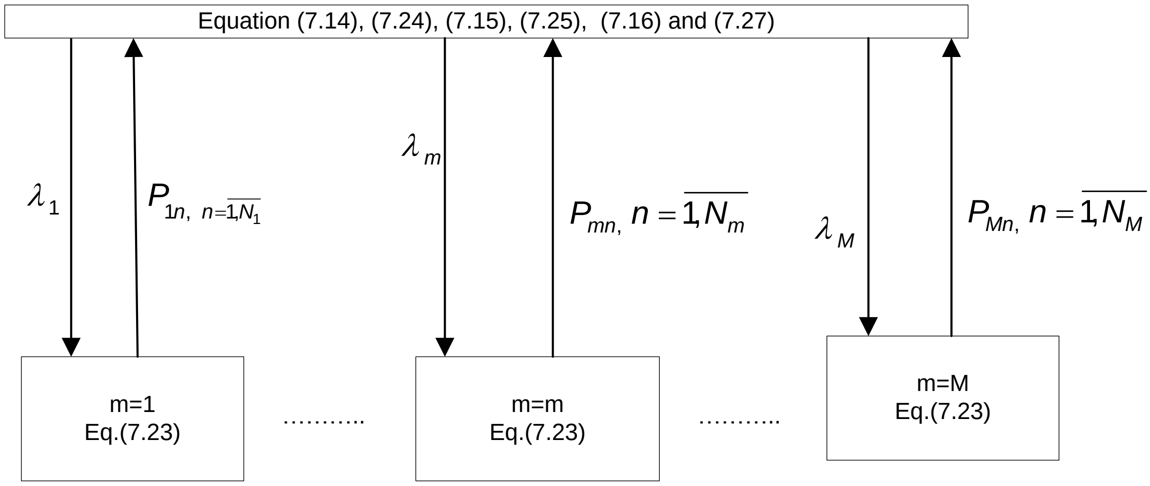


Figure 7.4: **Hierarchical structure for the MAEED problem solution using the developed Lagrange's decomposition-coordinating algorithm**

7.4 Algorithm of the Lagrange's decomposition-coordinating method for calculation of the MAEED problem

1. Values of all coefficients are given and the number of iterations of the second (L) and first level ($k_m, m=\overline{1,M}$) are given.
2. Initial values of the coordinating variables $\lambda_m^l, m=\overline{1,M}$ are guessed, l=1
3. Initial values of the tie-lines active powers are guessed

   $P_{Tmj}^{k_m}, P_{Tjm}^{k_m}, m=\overline{1,M}, j=\overline{1,M}, j \neq m$
4. Calculation on the first level are done for every subsystem $m=\overline{1,M}$ using Equation (7.23).
5. The solutions of the first level $P_{mn}^l,\ m=\overline{1,M},\ n=\overline{1,N_m}$ are checked according to the constraints (7.10) and are sent to the second level.
6. On the second level
   i. Equations (7.14) and (7.24) are solved
   ii. Equations (7.15) and (7.25) are solved

The gradient procedures (7.24) and (7.25) stop when the conditions $e_{PTmj} \leq \varepsilon_1$ *and* $e_{PTjm} \leq \varepsilon_2$ are fulfilled respectively or the maximum number of iterations $k_m, m=\overline{1,M}$ is reached.

The solution of the second level P^{l}_{Tmj} and P^{l}_{Tjm} , $m=\overline{1,M}$, $j=\overline{1,M}$, $j=m$ are checked according to the constraints (7.11) and (7.12)

7. Equations (7.16) and (7.27) are calculated. The gradient procedure (7.27) stops when the condition (7.28) is fulfilled. In this case the optimal solution for $\lambda_m, m=\overline{1M}$ is obtained and the corresponding to it optimal solutions of the problems on the first level are obtained.

   In the above algorithm for one-step of the gradient procedure for optimisation of λ_m on the second level k_m maximum number of iterations are performed to obtain the optimal solution of the MAEED sub-problem in the m^{th} area. The value of k_m can be different for each of the areas.

   The softwares *SACEED_Casestudy1_Sequential.m* and *SACEED_Casestudy2_Sequential.m* of the Lagrange's algorithm for calculation of single area Combined Economic Emission Dispatch (CEED) problem is given in *Appendix I*.

   The softwares *MAEED_Casestudy1_Parallel.m* and *MAEED_Casestudy2_Parallel.m* of the Lagrange's decomposition-coordinating algorithm for calculation of MAEED problem solution is given in *Appendices H to J*.

7.5 Studies of the single area and multi-area CEED Problem solutions

The proposed algorithm is tested on two benchmark models of single and multi-area power systems. They are:

(i) Four area with forty generators in each area without considering of the transmission line losses

(ii) Four area with three generators in each area with considering of the transmission line losses

The two bench mark model are applied for two considered scenarios, they are:

a) The whole power system is considered as a single area one prior to decompose the power system into multi-areas.

b) The whole power system is decomposed into multi-areas with tie-line constraints.

The flowchart of the Lagrange's decomposition-coordinating algorithm for calculation of the MAEED problem is given in Figure 7.5.

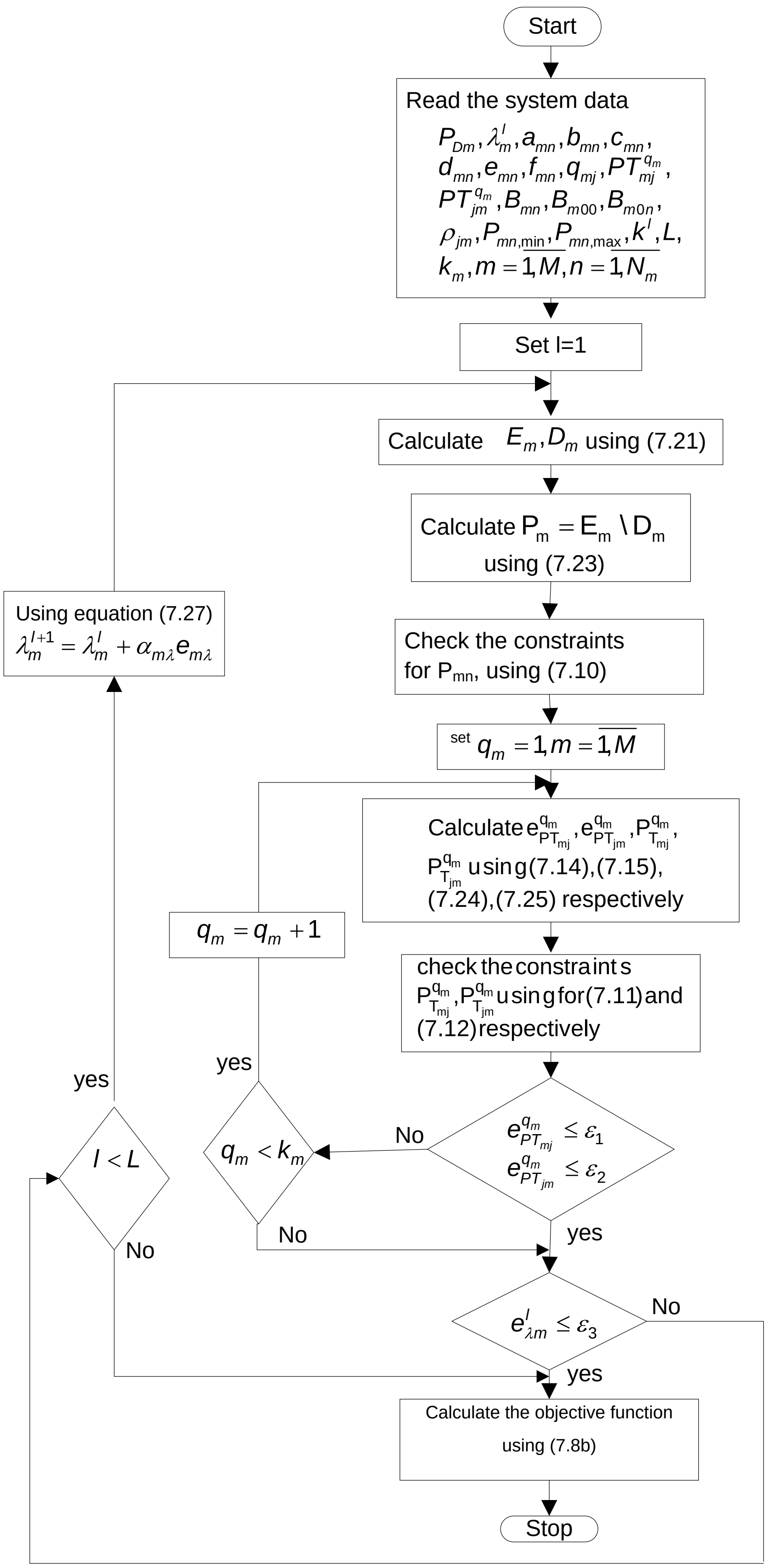


Figure 7. 5: **Flowchart of the multi-area economic emission dispatch problem**

7.5.1 Case study 1:

In this case study, a four area system is used to investigate the effectiveness of the proposed Lagrange's algorithm. The fuel cost and emission data of the system are given in (Basu, 2011), (Basu, 2013) and in the Table 7.1. There are ten generators in each area with different fuel and emission characteristics and tie-line transfer limits, which are given in Table 7.1 and Table 7.2 respectively. The system has a total power demand of 10500 [MW]. The loads of the four area are subdivided into 15%, 40%, 30% and 15% of the total power demand. The transmission cost is neglected in the process of problem solution since it is normally small as compared with the total fuel cost.

Table 7.1: **Multi-area combined economic emission dispatch problem data for the four-area forty-generator system**

Gen_{mn}	$P_{mn,min}$	$P_{mn,max}$	a_{mn}	b_{mn}	c_{mn}	d_{mn}	e_{mn}	f_{mn}
1	36	114	0.0069	6.73	94.705	0.048	−2.22	60
2	36	114	0.0069	6.73	94.705	0.048	−2.22	60
3	60	120	0.02028	7.07	309.54	0.0762	−2.36	100
4	80	190	0.00942	8.18	369.03	0.054	−3.14	120
5	47	97	0.0114	5.35	148.89	0.085	−1.89	50
6	68	140	0.01142	8.05	222.33	0.0854	−3.08	80
7	110	300	0.00357	8.03	287.71	0.0242	−3.06	100
8	135	300	0.00492	6.99	391.98	0.031	−2.32	130
9	135	300	0.00573	6.6	455.76	0.0335	−2.11	150
10	130	300	0.00605	12.9	722.82	0.425	−4.34	280
11	94	375	0.00515	12.9	635.2	0.0322	−4.34	220
12	94	375	0.00569	12.8	654.69	0.0338	−4.28	225
13	125	500	0.00421	12.5	913.4	0.0296	−4.18	300
14	125	500	0.00752	8.84	1760.4	0.0512	−3.34	520
15	125	500	0.00752	8.84	1760.4	0.0496	−3.55	510
16	125	500	0.00752	8.84	1760.4	0.0496	−3.55	510
17	220	500	0.00313	7.97	647.85	0.0151	−2.68	220
18	220	500	0.00313	7.95	649.69	0.0151	−2.66	222
19	242	550	0.00313	7.97	647.83	0.0151	−2.68	220
20	242	550	0.00313	7.97	647.81	0.0151	−2.68	220
21	254	550	0.00298	6.63	785.96	0.0145	−2.22	290
22	254	550	0.00298	6.63	785.96	0.0145	−2.22	285
23	254	550	0.00284	6.66	794.53	0.0138	−2.26	295
24	254	550	0.00284	6.66	794.53	0.0138	−2.26	295

25	254	550	0.00277	7.1	801.32	0.0132	−2.42	310
26	254	550	0.00277	7.1	801.32	0.0132	−2.42	310
27	10	150	0.52124	3.33	1055.1	1.842	-1.11	360
28	10	150	0.52124	3.33	1055.1	1.842	−1.11	360
29	10	150	0.52124	3.33	1055.1	1.842	−1.11	360
30	47	97	0.0114	5.35	148.89	0.085	−1.89	50
31	60	190	0.0016	6.43	222.92	0.0121	−2.08	80
32	60	190	0.0016	6.43	222.92	0.0121	−2.08	80
33	60	190	0.0016	6.43	222.92	0.0121	−2.08	80
34	90	200	0.0001	8.95	107.87	0.0012	−3.48	65
35	90	200	0.0001	8.62	116.58	0.0012	−3.24	70
36	90	200	0.0001	8.62	116.58	0.0012	−3.24	70
37	25	110	0.0161	5.88	307.45	0.095	−1.98	100
38	25	110	0.0161	5.88	307.45	0.095	−1.98	100
39	25	110	0.0161	5.88	307.45	0.095	−1.98	100
40	242	550	0.00313	7.97	647.83	0.0151	−2.68	220

Table 7.2: **Four area- forty generator tie-line transfer limits adopted from (Basu, 2013)**

Tie-line	P_{Tmin}	P_{Tmax}
$P_{T1,2}$	100	200
$P_{T1,3}$	100	200
$P_{T1,4}$	50	100
$P_{T2,3}$	100	200
$P_{T2,4}$	50	100
$P_{T3,4}$	50	100

7.5.2 Results of the solution of the single area 40 generator power system without considering transmission line losses

The whole power system is considered as a single area prior to decompose the Power System (PS) into multi-area. The multi-area power system data are given in Table 7.1. The single area CEED problem with 40 generator and neglecting transmission line losses is considered. The initial lambda is assumed as 4 and maximum number of iteration is set for 10000. The single area economic dispatch problem is solved using Min-Max, Max-Max, Min-Min and Max-Min penalty factors for different power demands from 8000 to 10500 [MW] in a sequential way. The single area CEED problem solution of the generator active powers, fuel cost, emission, CEED fuel cost and computation time based on Min-Max penalty factor method is given in Table 7.3.

Table 7. 3: Results from the single area CEED problem solution using Min-Max penalty factor (prior to decompose the PS into multi-area system)

P_D	Lambda	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22
[MW]		[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]
8000	17.02	114.00	114.00	68.74	113.43	97.00	84.35	232.82	232.33	233.26	130.00	136.80	133.86	182.02	219.01	219.44	219.44	329.42	330.54	359.19	359.19	427.55	427.14
8500	17.81	114.00	114.00	73.23	121.43	97.00	90.44	248.43	248.19	249.02	130.00	152.72	149.70	204.33	238.06	238.33	238.33	351.70	352.92	384.22	384.22	455.32	454.89
9000	18.62	114.00	114.00	77.88	129.70	97.00	96.75	264.59	264.60	265.34	130.00	169.20	166.09	227.41	257.78	257.88	257.88	374.75	376.07	410.11	410.12	484.06	483.60
9500	19.44	114.00	114.00	82.55	138.04	97.00	103.09	280.86	281.12	281.77	130.00	185.79	182.60	250.65	277.63	277.56	277.56	397.95	399.38	436.19	436.19	513.00	512.50
10000	20.28	114.00	114.00	87.35	146.57	97.00	109.60	297.53	298.05	298.60	130.00	202.79	199.51	274.47	297.97	297.73	297.73	421.73	423.27	462.91	462.91	542.65	542.12
10500	21.49	114.00	114.00	94.25	158.86	97.00	118.95	300.00	300.00	300.00	130.00	227.25	223.85	308.74	327.24	326.75	326.75	455.94	457.64	501.35	501.35	550.00	550.00

Continuation of Table 7.3

P_D	P23	P24	P25	P26	P27	P28	P29	P30	P31	P32	P33	P34	P35	P36	P37	P38	P39	P40	F_C	E_T	F_T	Number of iterations	Computation Time
[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[$/h]	[t/h]	[$/h]		[s]
8000	428.93	428.93	402.69	402.69	12.00	12.00	12.00	97.00	168.16	168.16	168.16	90.00	90.00	90.00	102.19	102.19	102.19	359.19	94020.87	41510.21	118657.88	17	0.0121
8500	456.50	456.50	428.67	428.67	12.69	12.69	12.69	97.00	174.43	174.43	174.43	90.00	90.00	90.00	108.87	108.87	108.87	384.22	99211.84	46738.68	127364.86	17	0.0123
9000	485.03	485.03	455.54	455.54	13.40	13.40	13.40	97.00	180.91	180.91	180.91	90.00	90.00	90.00	110.00	110.00	110.00	410.11	104507.57	52309.74	136470.61	18	0.0128
9500	513.75	513.75	482.61	482.61	14.12	14.12	14.12	97.00	187.44	187.44	187.44	90.00	90.00	90.00	110.00	110.00	110.00	436.19	109894.93	58339.47	145985.64	18	0.0130
10000	543.19	543.19	510.34	510.34	14.85	14.85	14.85	97.00	190.00	190.00	190.00	90.00	90.00	90.00	110.00	110.00	110.00	462.91	115419.60	64991.51	155913.27	19	0.0149
10500	550.00	550.00	550.00	550.00	15.91	15.91	15.91	97.00	190.00	190.00	190.00	90.00	90.00	90.00	110.00	110.00	110.00	501.35	121365.32	72403.47	166336.67	35	0.0150

The single area CEED problem solution of the 40 generator active powers, fuel cost, emission and CEED fuel cost values based on Max-Max penalty factor method is given in Table 7.4.

Table 7. 4: Results from the single area CEED problem solution using Max-Max penalty factor (prior to decompose the PS into multi-area system)

P_D	Lambda	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22
[MW]		[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]
8000	25.27	103.81	103.81	77.74	118.61	97.00	92.43	209.83	233.07	235.98	130.00	151.92	149.28	198.66	218.25	218.82	218.82	335.56	336.61	364.48	364.48	435.10	434.64
8500	26.79	110.51	110.51	83.11	126.80	97.00	99.13	223.04	249.60	252.88	130.00	162.87	160.35	214.57	235.87	236.25	236.25	357.87	359.04	389.40	389.40	464.94	464.44
9000	28.32	114.00	114.00	88.54	135.10	97.00	105.91	236.43	266.35	270.00	130.00	173.97	171.56	230.69	253.73	253.91	253.91	380.47	381.76	414.65	414.65	495.17	494.63
9500	29.86	114.00	114.00	93.99	143.42	97.00	112.72	249.86	283.14	287.17	135.69	185.10	182.81	246.86	271.64	271.61	271.61	403.13	404.55	439.96	439.96	525.48	524.91

10000	31.55	114.00	114.00	99.97	152.56	97.00	120.19	264.61	300.00	300.00	148.75	197.32	195.17	264.62	291.31	291.07	291.07	428.03	429.58	467.78	467.78	550.00	550.00
10500	34.09	114.00	114.00	108.98	166.32	97.00	131.45	286.81	300.00	300.00	168.41	215.73	213.77	291.36	320.92	320.35	320.35	465.51	467.27	509.65	509.65	550.00	550.00

Continuation of Table 7.4

P_D	P23	P24	P25	P26	P27	P28	P29	P30	P31	P32	P33	P34	P35	P36	P37	P38	P39	P40	F_C	E_T	F_T	Number of iterations	Computation Time
[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[$/h]	[t/h]	[$/h]		[s]
8000	436.64	436.64	414.11	414.11	10.05	10.05	10.05	97.00	147.40	147.40	147.40	90.00	90.00	90.00	88.59	88.59	88.59	364.48	94434.07	41393.31	147768.77	41	0.0151
8500	466.23	466.23	441.74	441.74	10.74	10.74	10.74	97.00	152.41	152.41	152.41	90.00	90.00	90.00	94.80	94.80	94.80	389.40	99532.04	46675.59	160784.36	41	0.0156
9000	496.21	496.21	469.73	469.73	11.43	11.43	11.43	97.00	157.49	157.49	157.49	90.00	90.00	90.00	101.09	101.09	101.09	414.65	104726.81	52429.09	174559.20	42	0.0157
9500	526.27	526.27	497.80	497.80	12.12	12.12	12.12	97.00	162.58	162.58	162.58	90.00	90.00	90.00	107.40	107.40	107.40	439.96	110043.89	59212.07	189106.79	42	0.0158
10000	550.00	550.00	528.64	528.64	12.88	12.88	12.88	97.00	168.17	168.17	168.17	90.00	90.00	90.00	110.00	110.00	110.00	467.78	115532.36	67169.37	204428.11	60	0.0179
10500	550.00	550.00	550.00	550.00	14.03	14.03	14.03	97.00	176.59	176.59	176.59	90.00	90.00	90.00	110.00	110.00	110.00	509.65	121425.58	76610.17	220809.06	75	0.0191

The single area CEED problem solution of the 40 generator active powers, fuel cost, emission and CEED fuel cost values based on Min-Min penalty factor method is given in Table 7.5.

Table 7.5: **Results from the single area CEED problem solution using Min-Min penalty factor (prior to decompose the PS into multi-area system)**

P_D	Lambda	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22
[MW]		[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]
8000	61.54	91.32	91.32	110.13	122.59	97.00	114.18	113.64	261.96	284.35	156.73	105.05	108.01	146.86	180.32	178.00	178.00	330.08	333.92	369.04	369.04	520.52	517.33
8500	66.49	97.52	97.52	118.84	131.36	97.00	123.15	118.35	282.49	300.00	172.19	108.95	112.62	154.65	194.33	191.51	191.51	352.63	356.88	395.23	395.23	550.00	550.00
9000	74.61	107.67	107.67	120.00	145.73	97.00	137.84	126.05	300.00	300.00	197.50	115.32	120.17	167.42	217.28	213.63	213.63	389.56	394.48	438.12	438.12	550.00	550.00
9500	85.20	114.00	114.00	120.00	164.48	97.00	140.00	136.12	300.00	300.00	230.56	123.64	130.02	184.09	247.25	242.50	242.50	437.78	443.58	494.12	494.13	550.00	550.00
10000	97.15	114.00	114.00	120.00	185.64	97.00	140.00	147.47	300.00	300.00	267.86	133.03	141.15	202.90	281.05	275.09	275.09	492.18	498.97	550.00	550.00	550.00	550.00
10500	133.21	114.00	114.00	120.00	190.00	97.00	140.00	181.74	300.00	300.00	300.00	161.35	174.71	259.66	383.07	373.41	373.41	500.00	500.00	550.00	550.00	550.00	550.00

Continuation of Table 7.5

P_D	P23	P24	P25	P26	P27	P28	P29	P30	P31	P32	P33	P34	P35	P36	P37	P38	P39	P40	F_C	E_T	F_T	Number of iterations	Computation Time
[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[$/h]	[t/h]	[$/h]		[s]
8000	513.05	513.05	477.20	477.20	10.00	10.00	10.00	97.00	81.38	81.38	81.38	90.00	90.00	90.00	76.64	76.64	76.64	369.04	94337.69	47322.89	286458.85	142	0.0334

8500	550.00	550.00	512.61	512.61	10.00	10.00	10.00	97.00	80.96	80.96	80.96	90.00	90.00	90.00	82.57	82.57	82.57	395.23	99496.73	54923.78	318410.07	194	0.0353
9000	550.00	550.00	550.00	550.00	10.00	10.00	10.00	97.00	80.29	80.29	80.29	90.00	90.00	90.00	92.28	92.28	92.28	438.12	104909.47	64177.35	353539.32	275	0.0442
9500	550.00	550.00	550.00	550.00	10.00	10.00	10.00	97.00	79.40	79.40	79.40	90.00	90.00	90.00	104.96	104.96	104.96	494.12	110602.38	75891.80	393445.51	311	0.0509
10000	550.00	550.00	550.00	550.00	10.78	10.78	10.78	97.00	78.41	78.41	78.41	90.00	90.00	90.00	110.00	110.00	110.00	550.00	116597.20	90298.30	438864.19	486	0.1018
10500	550.00	550.00	550.00	550.00	14.82	14.82	14.82	97.00	75.40	75.40	75.40	90.00	90.00	90.00	110.00	110.00	110.00	550.00	123555.26	108481.51	495814.76	1070	0.1999

The single area CEED problem solution of the 40 generator active powers, fuel cost, emission and CEED fuel cost values of the Max-Min penalty factor method is given in Table 7.6.

Table 7.6: **Results from the single area CEED problem solution using Max-Min penalty factor (prior to decompose the PS into multi-area system)**

P_D	Lambda	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22
[MW]		[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]
8000	119.63	74.91	74.91	120.00	124.77	97.00	125.43	110.00	268.45	295.25	170.45	94.00	94.00	125.00	151.27	149.97	149.97	330.80	334.79	364.85	364.85	545.95	542.49
8500	135.16	82.05	82.05	120.00	138.17	97.00	140.00	111.85	300.00	300.00	194.53	96.73	98.11	130.59	167.98	166.06	166.06	364.64	369.27	403.45	403.45	550.00	550.00
9000	158.84	92.94	92.94	120.00	158.60	97.00	140.00	120.93	300.00	300.00	231.23	102.45	104.88	142.23	193.45	190.58	190.58	416.22	421.83	462.28	462.28	550.00	550.00
9500	182.51	103.83	103.83	120.00	179.02	97.00	140.00	130.02	300.00	300.00	267.92	108.16	111.65	153.86	218.91	215.10	215.10	467.80	474.39	521.11	521.12	550.00	550.00
10000	239.92	114.00	114.00	120.00	190.00	97.00	140.00	152.06	300.00	300.00	300.00	122.02	128.07	182.07	280.67	274.56	274.56	500.00	500.00	550.00	550.00	550.00	550.00
10500	352.35	114.00	114.00	120.00	190.00	97.00	140.00	195.21	300.00	300.00	300.00	149.15	160.23	237.32	401.61	391.00	391.00	500.00	500.00	550.00	550.00	550.00	550.00

Continuation of Table 7.6

P_D	P23	P24	P25	P26	P27	P28	P29	P30	P31	P32	P33	P34	P35	P36	P37	P38	P39	P40	F_C	E_T	F_T	Number of iterations	Computation Time
[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[MW]	[$/h]	[t/h]	[$/h]		[s]
8000	539.72	539.72	501.77	501.77	10.00	10.00	10.00	97.00	82.10	82.10	82.10	90.00	90.00	90.00	66.59	66.59	66.59	364.85	94218.02	49763.14	546714.48	316	0.0403
8500	550.00	550.00	550.00	550.00	10.00	10.00	10.00	97.00	81.57	81.57	81.57	90.00	90.00	90.00	74.28	74.28	74.28	403.45	99443.08	58477.22	610074.94	620	0.0780
9000	550.00	550.00	550.00	550.00	10.00	10.00	10.00	97.00	80.76	80.76	80.76	90.00	90.00	90.00	86.01	86.01	86.01	462.28	104932.26	69792.09	683574.27	678	0.1012
9500	550.00	550.00	550.00	550.00	10.00	10.00	10.00	97.00	79.95	79.95	79.95	90.00	90.00	90.00	97.73	97.73	97.73	521.11	110591.53	83078.13	768910.56	694	0.1122
10000	550.00	550.00	550.00	550.00	10.00	10.00	10.00	97.00	78.00	78.00	78.00	90.00	90.00	90.00	110.00	110.00	110.00	550.00	116614.87	97740.12	869466.81	2858	0.4939
10500	550.00	550.00	550.00	550.00	10.00	10.00	10.00	97.00	74.16	74.16	74.16	90.00	90.00	90.00	110.00	110.00	110.00	550.00	123492.31	109405.10	1017533.79	3099	0.5359

Figure 7.6 shows the comparison of the 40 generator single area CEED problem solution based on Min-Max, Max-Max, Min-Min and Max-Min penalty factors.

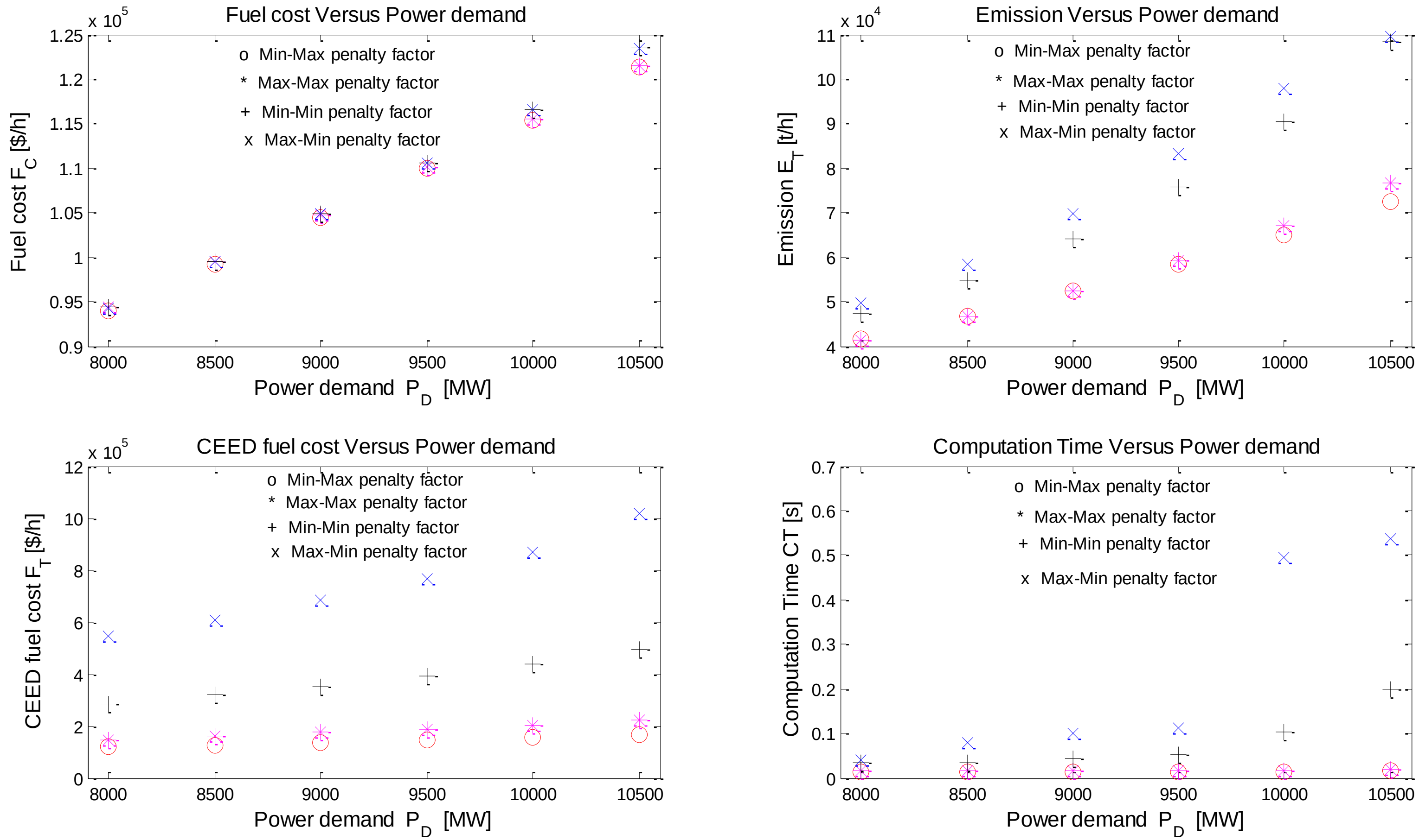


Figure 7.6: **Criteria cost functions and computation time of the 40 generator single area economic emission dispatch problem solutions**

7.5.3 Discussion on the results for Single area CEED problem solution

The 40 generator single area economic emission problem is solved for the considered four types of penalty factors Min-Max, Max-Max, Min-Min, and Max-Min and for the various power demands from 8000 to 10500 [MW]. The results are given in Table 7.3, 7.4, 7.5 and 7.6 respectively. Single area CEED problem solutions: fuel cost, emission, CEED fuel cost and computation time is less for the Min-Max penalty factor in comparison with Max-Max, Min-Min, and Max-Min penalty factors and is given in Figure 7.6.

(i) For 10500 MW power demand, The CEED fuel cost value for Min-Max penalty factor is 166336.67 $/hr (Table 7.3) with a computation time of 0.0150 seconds.

(ii) For the same power demand (10500 MW), The CEED fuel cost value for Max-Min penalty factor is 1017533.79 $/hr (Table 7.6) with a computation time of 0.5359 seconds.

By comparing the Single area CEED fuel cost and computation time of (i) and (ii) results to the deviation of 143.79 % and 189.10 % respectively. It is clear that Min-Max penalty factor provides 43.79 % less CEED fuel cost and 89.10 % less computation time in comparsion with Max – Min penalty factor.

7.5.4 Results of the solution of the four area 40 generator MAEED problem

The initial lambda of each area is assumed as [4 4 4 4], maximum number of iterations is set to 10000. The whole power system is decomposed into a multi-area with considering the tie-line constraints. The multi-area economic emission dispatch problem is solved based on the different types of penalty factors: Min-Max, Max-Max, Min-Min and Max-Min in a task-parallel way. 10 generators are assigned to individual workers using Matlab Distributed Computing Engine (MDCE). Each area has 10 generators. The generator active powers, fuel cost, emission, tie-line powers, and computation time of the MAEED problem solution based on Min-Max penalty factor is given in Table 7.7.

Table 7. 7: Results from the four area – forty generator MAEED problem solution using Min-Max price penalty factor

Worker Name	Area Number	Area P_D [MW]	Lambda	Generator real power values in [MW]										F_C [$/h]	E_T [ton/h]	F_T [$/h]	Tie-line power [MW]	Number of Iterations	Computation Time [s]
Mat32-wrk1	1	1575	16.53	114.00	114.00	65.94	108.45	94.31	80.55	223.09	222.45	223.43	130.00	14999.37	12584.38	19409.15	-89.20	3010	102.25
Mat32-wrk2	2	4200	23.55	269.09	265.49	367.36	377.32	376.40	376.40	500.00	500.00	550.00	550.00	56258.71	36514.40	74802.25	-95.88		
Mat32-wrk3	3	3150	19.37	510.65	510.15	511.42	511.42	480.40	480.40	14.06	14.06	14.06	97.00	33871.94	18539.92	47533.92	-13.70		
Mat32-wrk4	4	1575	18.61	180.84	180.84	180.84	90.00	90.00	90.00	110.00	110.00	110.00	409.82	14901.53	4389.24	20272.87	59.86		
Total		10500												120031.55	72027.94	162018.18	-38.60 29.62		

The generator active powers, fuel cost, emission, tie-line powers, and computation time of the MAEED problem based on Max-Max penalty factor is given in Table 7.8

Table 7. 8: Results from the four area – forty generator MAEED problem solution using Max-Max price penalty factor

Worker Name	Area Number	Area P_D [MW]	Lambda	Generator real power values in [MW]										F_C [$/h]	E_T [ton/h]	F_T [$/h]	Tie-line power [MW]	Number of Iterations	Computation Time [s]
Mat32-wrk1	1	1575	26.77	110.44	110.44	83.05	126.71	97.00	99.05	222.90	249.42	252.69	130.00	16032.35	13852.85	27217.69	56.97	3765	132.13
Mat32-wrk2	2	4200	43.12	281.09	279.83	386.32	426.08	424.35	424.35	500.00	500.00	550.00	550.00	59123.69	42533.18	109912.01	-45.83		
Mat32-wrk3	3	3150	29.31	514.74	514.18	515.62	515.62	487.85	487.85	11.88	11.88	11.88	97.00	34066.52	18591.71	62334.97	-104.44		
Mat32-wrk4	4	1575	30.27	163.93	163.93	163.93	90.00	90.00	90.00	109.08	109.08	109.08	446.71	14914.37	4609.26	26608.50	32.10		
Total		10500												124136.94	79587.00	226073.18	32.96 32.22		

The generator active powers, fuel cost, emission, tie-line powers, and computation time of the MAEED problem based on Max-Max penalty factor is given in Table 7.9

Table 7. 9: **Results from the four area – forty generator MAEED problem solution using Min-Min price penalty factor**

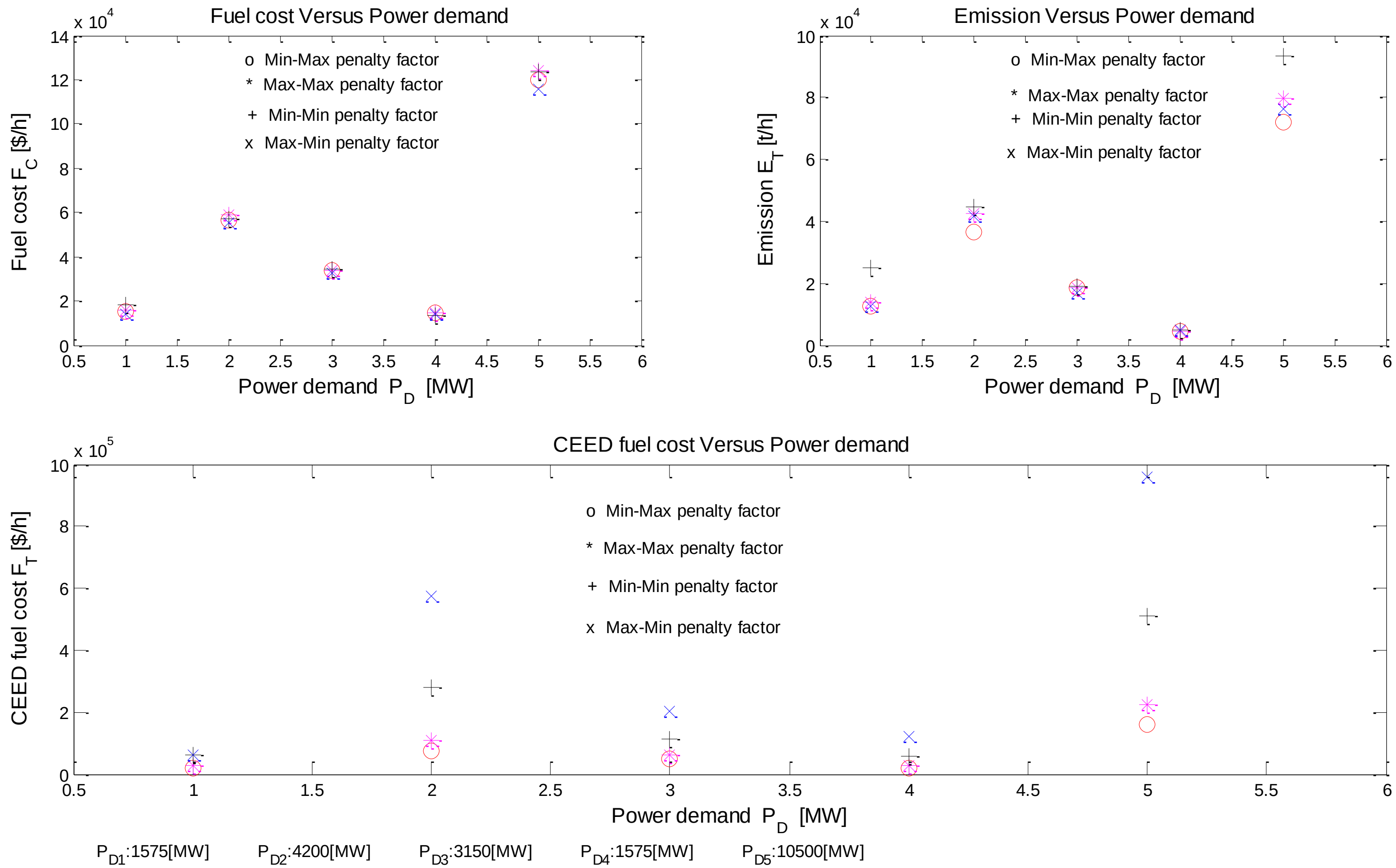
Worker Name	Area Number	Area P_D [MW]	Lambda	Generator real power values in [MW]										F_C [$/h]	E_T [ton/h]	F_T [$/h]	Tie-line power [MW]	Number of Iterations	Computation Time [s]
Mat32-wrk1	1	1575	74.31	107.29	107.29	120.00	145.19	97.00	137.30	125.77	300.00	300.00	196.57	18045.24	25100.89	61115.63	2.0305	4132	161.13
Mat32-wrk2	2	4200	165.68	186.85	204.92	310.75	474.91	461.92	461.92	500.00	500.00	550.00	550.00	57276.39	44497.2	278567.6	83.534		
Mat32-wrk3	3	3150	62.70	529.98	526.72	522.24	522.24	485.49	485.49	10.00	10.00	10.00	97.00	34335.11	18840.1	113211.2	-24.22		
Mat32-wrk4	4	1575	85.56	99.37	105.37	79.37	120.23	132.35	106.23	115.39	105.39	135.39	496.03	13620	4756.795	57285.41	10.153		
Total		10500												123276.7	93194.99	510179.9	-10.91 -44.54		

The generator active powers, fuel cost, emission, tie-line powers, and computation time of the MAEED problem based on Max-Max penalty factor is given in Table 7.10

Table 7. 10: **Results from the four area – forty generator MAEED problem solution using Max-Min price penalty factor**

Worker Name	Area Number	Area P_D [MW]	Lambda	Generator real power values in [MW]										F_C [$/h]	E_T [ton/h]	F_T [$/h]	Tie-line power [MW]	Number of Iterations	Computation Time [s]
Mat32-wrk1	1	1575	96.48	64.26	64.26	105.16	104.79	97.00	103.07	110.00	220.82	241.06	134.57	14004.53	12811.98	62072.33	-96.00	4557	193.08
Mat32-wrk2	2	4200	408.44	162.69	176.27	264.88	461.96	449.10	449.10	500.00	500.00	550.00	550.00	55199.82	41848.56	574342.46	-94.00		
Mat32-wrk3	3	3150	107.96	497.28	494.18	492.23	492.23	459.05	459.05	10.00	10.00	10.00	97.00	32616.81	16779.97	204545.12	-140.00		
Mat32-wrk4	4	1575	188.15	79.76	79.76	79.76	90.00	90.00	90.00	100.53	100.53	100.53	535.14	13931.41	5003.97	119769.39	7.00		
Total		10500												115752.57	76444.47	960729.30	-47.00 -42.00		

Figure 7.7 shows the fuel cost, emission and CEED fuel cost of the four area forty generator MAEED problem solution using various price penalty factors.



Where P_{D1}, P_{D2}, P_{D3}, and P_{D4} are the power demands of the area 1 to 4 respectively.

Figure 7. 7: Comparison of the criterion function values of the four area forty generator MAEED problem solution using various price penalty factors.

The comparison of the single area CEED and multi-area CEED problem solutions obtained by the developed Lagrange's and by the Differential Evolution adopted from the reference paper (Basu,2011) is given in Table 7.11.

Table 7.11: Comparison of the single area CEED and the multi-area CEED problem solutions obtained by the developed Lagrange's algorithm with the Differential Evolution (DE) solution adopted from (Basu, 2011)

Single area CEED problem solution for P_D 10500 [MW] using developed Lagrange's method				Multi-area CEED problem solution using developed Lagrange's decomposition-coordinating technique				DE solution based on Max-Max penalty factor for P_D 10500 [MW] adopted from (Basu,2011)		
Penalty factor/ Criteria	F_C in [$/hr]	E_T in [t/hr]	F_T in [$/hr]	Penalty factor/ Criteria	F_C in [$/hr]	E_T in [t/hr]	F_T in [$/hr]	F_C in [$/hr] (x10^5)	E_T in [t/hr] (x10^4)	F_T in [$/hr]
Min-Max	121365.32	72403.47	166336.67	Min-Max	120031.55	72027.94	162018.18	1.2184	3.7479	------
Max-Max	121425.58	76610.17	220809.06	Max-Max	124136.94	79587	226073.18			
Min-Min	123555.26	108481.51	495814.76	Min-Min	123276.7	93194.99	510179.9			
Max-Min	123492.31	109405.1	1017533.79	Max-Min	115752.57	76444.47	960729.3			

7.5.5 Discussion on the results of the 40 generator MAEED problem solution

The 40 generator multi-area economic emission problem is solved for the considered four types of penalty factors: Min-Max, Max-Max, Min-Min, and Max-Min. The results are given in Table 7.7, 7.8, 7.9 and 7.10 respectively. The forty-generator MAEED problem has four areas and each area has considered power demands of 1575, 4200, 3150 and 1575 [MW] respectively. The MAEED problem solutions: fuel cost, emission, CEED fuel cost and computation time is less for the Min-Max penalty factor in comparison with the other types of penalty factors and is given in Figure 7.7. The comparison of the single area CEED and MAEED solutions are given in Table 7.11. It is proved that MAEED solution is better in comparison with the single area CEED solutions based on Min-Max penalty factor. It is proved that Lagrange's decomposition-coordinating method provide better solution in comparison with the differential evolution solutions adopted from the reference paper (Basu, 2011) and is given in Table 7.11

7.6 Case study 2

In this case study, a four area- three generator system is used to investigate the effectiveness of the proposed Lagrange's algorithm. The fuel cost and emission data of the system are given in (Chen and Wang, 2010). There are three generators in each area with different fuel and emission characteristics and tie-line transfer limits, which are given in Table 7.12 and Table 7.13 respectively. The area power demands are 500, 410, 580 and 600 MW respectively. The transmission loss coefficient is given in Table 7.14.

Table 7. 12: **Multi-area economic emission dispatch data of a four-area three generator system**

Gen_{mn}	a_{mn}	b_{mn}	c_{mn}	d_{mn}	e_{mn}	f_{mn}	$P_{mn,min}$	$P_{mn,max}$
G1,1	0.03546	38.30553	1243.5311	0.00683	-0.54551	40.2669	35	210
G1,2	0.02111	36.32782	1658.5696	0.00461	-0.5116	42.89553	130	325
G1,3	0.01799	38.27041	1356.6592	0.00461	-0.5116	42.89553	125	315
G2,1	0.15247	38.53973	756.7989	0.00484	-0.32767	33.85932	10	150
G2,2	0.02803	40.39655	449.9977	0.00754	-0.54551	50.63931	35	110
G2,3	0.14834	38.34001	558.5696	0.00661	-0.63262	45.83267	125	215
G3,1	0.10587	46.15916	451.3251	0.00914	-0.43211	48.2156	15	175
G3,2	0.07505	43.83562	673.0267	0.00533	-0.61173	52.4521	30	215
G3,3	0.11934	50.63211	530.7199	0.00674	-0.49731	41.1042	50	335
G4,1	0.10587	46.15916	851.3251	0.00728	-0.6821	30.3632	15	175
G4,2	0.13552	41.03782	1038.533	0.00479	-0.5066	25.1765	30	215
G4,3	0.08963	33.56211	1285.907	0.00387	-0.4934	27.7549	50	335

Table 7. 13: **Four area- three generator systems tie-line transfer limits**

Tie-line	P_{Tmin}	P_{Tmax}
$P_{T1,2}$	5	60
$P_{T1,3}$	5	50
$P_{T1,4}$	5	60
$P_{T2,3}$	5	60
$P_{T2,4}$	5	50
$P_{T3,4}$	5	60

Table 7.14: **Transmission loss coefficients of the four areas – three-generator system**

Area 1 B_{mn}	0.000071	0.00003	0.000025	Area 2 B_{mn}	0.000056	0.000045	0.000015
	0.00003	0.000069	0.000032		0.000023	0.000042	0.000047
	0.000025	0.000032	0.00008		0.000032	0.000023	0.000027
Area 3 B_{mn}	0.00002	0.000028	0.000053	Area 4 B_{mn}	0.000074	0.00003	0.000025
	0.000086	0.000034	0.000016		0.000049	0.000069	0.000037
	0.000053	0.000016	0.000028		0.000022	0.000032	0.000083

7.6.1 Results from the solution of the four-area three-generator MAEED problem

The initial lambda is taken as [50 50 50 50], Maximum Number of Iterations is set to 50000 and the initial tie-line values are given in Table 7.15.

Table 7.15: **Initial tie-line values**

P_{T12}	P_{T13}	P_{T14}	P_{T23}	P_{T24}	P_{T34}
15	10	08	11	13	09

Case a) The whole power system is considered as a single area one.

The single area CEED problem is solved using the two considered price penalty factors Min-Max and Max-Max and the solutions are given in Tables 7.16 and 7.17 respectively.

Table 7.16: **Results from the solution of the single area CEED problem using Min-Max penalty factor**

P_D [MW]	Lambda	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	P7 [MW]	P8 [MW]
1500	74.8297	144.6437	184.5802	174.6660	95.0039	110.0000	125.0000	103.5183	137.0107
1600	78.2715	154.1122	195.6567	185.3639	103.1093	110.0000	125.0000	115.1357	149.2558
1700	81.7491	163.3826	206.5893	195.8833	111.2989	110.0000	125.0000	126.8738	161.6282
1800	85.2627	172.4497	217.3744	206.2198	119.5735	110.0000	125.0000	138.7337	174.1289
1900	88.8127	181.3080	228.0079	216.3687	127.9336	110.0000	125.0000	150.7162	186.7590
2000	92.3993	189.9519	238.4859	226.3253	136.3801	110.0000	125.0000	162.8225	199.5195
2090	95.6588	197.5435	247.7799	235.1179	144.0563	110.0000	125.0000	173.8247	211.1162

Continuation of Table 7.16

P9 [MW]	P10 [MW]	P11 [MW]	P12 [MW]	P_L [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]	Number of Iterations	CT [S]
86.4669	95.9171	97.3629	177.4651	31.6360	86784.6361	836.1476	99677.7605	466	0.1064
97.5521	104.9239	105.2306	190.5813	35.9225	93203.6025	941.9621	107353.5833	474	0.1113
108.7525	114.0243	113.1800	203.8339	40.4480	99833.4259	1060.7157	115377.2307	482	0.1130
120.0690	123.2191	121.2119	217.2239	45.2049	106678.3314	1192.5218	123752.7137	489	0.1149
131.5027	132.5090	129.3269	230.7525	50.1856	113742.6299	1337.4971	132484.1385	497	0.1151
143.0544	141.8949	137.5256	244.4208	55.3819	121030.7035	1495.7616	141575.6908	504	0.1181
153.5526	150.4247	144.9767	256.8424	60.2359	127784.9874	1649.6639	150069.5359	510	0.1214

Table 7. 17: **Results from the solution of the single area CEED problem using Max-Max penalty factor**

P_D [MW]	Lambda	P1 [MW]	P2 [MW]	P3 [MW]	P4 [MW]	P5 [MW]	P6 [MW]	P7 [MW]	P8 [MW]
1500	121.0343	128.7165	210.0978	200.5613	87.8812	110.0000	125.0000	89.3108	124.9901
1600	130.7660	138.5650	226.8030	216.3931	95.1668	110.0000	125.0000	98.4578	134.5827
1700	140.6301	148.2814	243.3431	232.0117	102.5514	110.0000	125.0000	107.7293	144.3058
1800	150.5519	157.7877	259.5875	247.2931	109.9793	110.0000	125.7746	117.0551	154.0858
1900	159.9389	166.5378	274.5996	261.3609	117.0068	110.0000	133.2372	125.8782	163.3386
2000	169.4371	175.1532	289.4416	275.2152	124.1175	110.0000	140.7883	134.8058	172.7010
2090	178.0816	182.7879	302.6497	287.4964	130.5891	110.0000	147.6606	142.9309	181.2219

Continuation of Table 7.17

P9 [MW]	P10 [MW]	P11 [MW]	P12 [MW]	P_L [MW]	F_C [$/h]	E_T [kg/h]	F_T [$/h]	Number of Iterations	CT [S]
105.4409	89.1698	100.0139	162.7174	33.9008	86329.9525	847.3549	137468.8623	1295	0.3184
116.2710	95.5106	106.9993	175.3830	39.1332	92524.7627	959.3179	150024.4970	1330	0.3299
127.2483	101.9376	114.0797	188.2208	44.7100	98909.6131	1084.4083	163556.2147	1359	0.3634
138.2900	108.4022	121.2015	201.1339	50.5916	105496.4226	1222.3621	178079.7865	1318	0.3658
148.7364	114.5184	127.9394	213.3508	56.5051	112342.4129	1370.2609	193585.6293	1324	0.3755
159.3066	120.7070	134.7572	225.7125	62.7069	119380.9954	1530.2694	210038.2157	1342	0.3791
168.9267	126.3394	140.9621	236.9631	68.5286	125883.5243	1684.6603	225665.1721	1360	0.3836

7.6.2 Discussion on the results of the four-area three-generator system

The considered different power demand values given in Table 7.16 and 7.17 are used to solve the single area CEED problem using Min-Max and Max-Max penalty factors. The fuel cost and computation time are less in Min-Max penalty factor in comparison with the Max-Max one.

Case b) The whole power system is decomposed into multi-area with tie-lines – 4 areas with 3 generators in every area

The multi-area economic dispatch problem is solved using different penalty factors Min-Max, Max-Max, Min-Min and Max-Min in a task-parallel way. The group of generators in every area is assigned to individual workers using MDCE in order to optimise the generator real power values which are given in Table 7.18. The fuel cost and emission values of the MAEED problem are given in Table 7.19. The tie-line power flows and transmission loss values of the developed Lagrange's decomposition-coordinating algorithm are given in Table 7.20 and Table 7.21 respectively. The comparison of the single area CEED and the MAEED solutions with that of the sequential way Particle Swarm Optimisation (PSO) adopted from (Chen and Wang, 2010) is given in Table 7.22

Table 7. 18: Optimised real power generator values of four area three generator MAEED problem

Name of the worker	P_{mn} [MW]	Type of Penalty factor			
		Min-Max	Max-Max	Min-Min	Max-Min
mat32-wrk1	P1,1	160.1381	131.45	158.03	119.91
	P1,2	194.7165	209.49	187.44	207.49
	P1,3	188.1241	202.25	185.21	203.07
mat32-wrk2	P2,1	150.00	150.00	150.00	150.000
	P2,2	110.00	110.00	110.00	110.000
	P2,3	208.48	191.63	180.82	180.80
mat32-wrk3	P3,1	175.00	175.00	175.00	175.000
	P3,2	215.00	215.00	215.00	215.000
	P3,3	244.77	236.38	222.42	222.27
mat32-wrk4	P4,1	175.00	164.09	175.00	175.00
	P4,2	173.62	180.20	215.00	215.00
	P4,3	302.44	306.18	242.23	242.07

Table 7.19: Tie line power values of the four areas - three generator power system solution

P_{Tmn} in [MW]	P_{T12}	P_{T13}	P_{T14}	P_{T23}	P_{T24}	P_{T34}
Min-Max	9.9964	9.9964	9.9964	9.9910	9.9910	9.9934
Max-Max	9.9944	9.9944	9.9944	9.9881	9.9881	9.9876
Min-Min	9.9994	9.9994	9.9944	9.9991	9.9991	9.9989
Max-Min	9.9863	9.9863	9.9863	9.9853	9.9853	9.9243

Table 7.20: Transmission power loss values of the four areas three generator power system solution

Area	P_L [MW]			
	Min-Max	Max-Max	Min-Min	Max-Min
1	12.98	13.22	11.20	13.40
2	28.51	11.66	10.55	12.60
3	24.76	16.41	17.45	20.32
4	21.08	20.51	19.10	17.45
Total	87.35	61.82	58.30	63.77

Table 7.21: **Fuel cost and emission values for the four-area three-generator MAEED problem solution**

Area	P_D in [MW]	Lambda				F_C in [$/hr]				E_T in [ton/hr]				F_T in [$/hr]			
		Penalty factor's				Penalty factor's				Penalty factor's				Penalty factor's			
		Min-Max	Max-Max	Min-Min	Max-Min	Min-Max	Max-Max	Min-Min	Max-Min	Min-Max	Max-Max	Min-Min	Max-Min	Min-Max	Max-Max	Min-Min	Max-Min
1	500	71.82	112.04	197.43	455.87	27012.55	26920.22	26454.31	26322.43	352.62	355.92	339.88	337.40	32762.54	42755.42	64451.78	130740.04
2	410	184.72	240.95	287.79	490.61	30200.67	28554.97	27542.57	27540.73	342.82	376.72	323.04	323.00	40422.52	55923.87	50998.70	101065.72
3	580	130.74	249.14	370.93	2526.10	45409.29	44506.54	43035.10	43019.71	719.99	742.93	683.75	683.75	50353.61	84539.67	85416.68	415271.82
4	600	115.66	248.56	437.84	2614.02	44055.88	44076.47	42972.47	42960.55	443.39	448.09	406.93	406.72	49539.13	83181.94	109469.05	512381.87
Total	2090					146678.40	144058.20	140004.64	139843.44	1858.84	1923.70	1753.60	1750.50	173077.81	266400.92	310336.25	1159459.46

Table 7. 22: **Comparison of the single area CEED and the MAEED solutions with the PSO algorithm adopted from (Chen and Wang, 2010)**

Method	Lagrange's method (Developed)				PSO adopted from (Chen and Wang, 2010)
Criteria	Single area CEED problem solution for P_D 2090 [MW]		MAEED problem solution for the area P_D of [500 410 580 600] [MW]		PSO MAEED problem solution in a sequential way for area P_D of [500 410 580 600] [MW]
	Max-Max	Min-Max	Max-Max	Min-Max	Max-Max
P_L in [MW]	68.5286	60.2359	61.82	87.35	49.54
F_C in [$/hr]	125883.5243	127784.9874	144058.20	146678.40	132416.90
E_T in [kg/hr]	1684.6603	1649.6639	1923.70	1858.84	1645.20
F_T in [$/hr]	225665.1721	150069.5359	266400.92	173077.81	269747.40

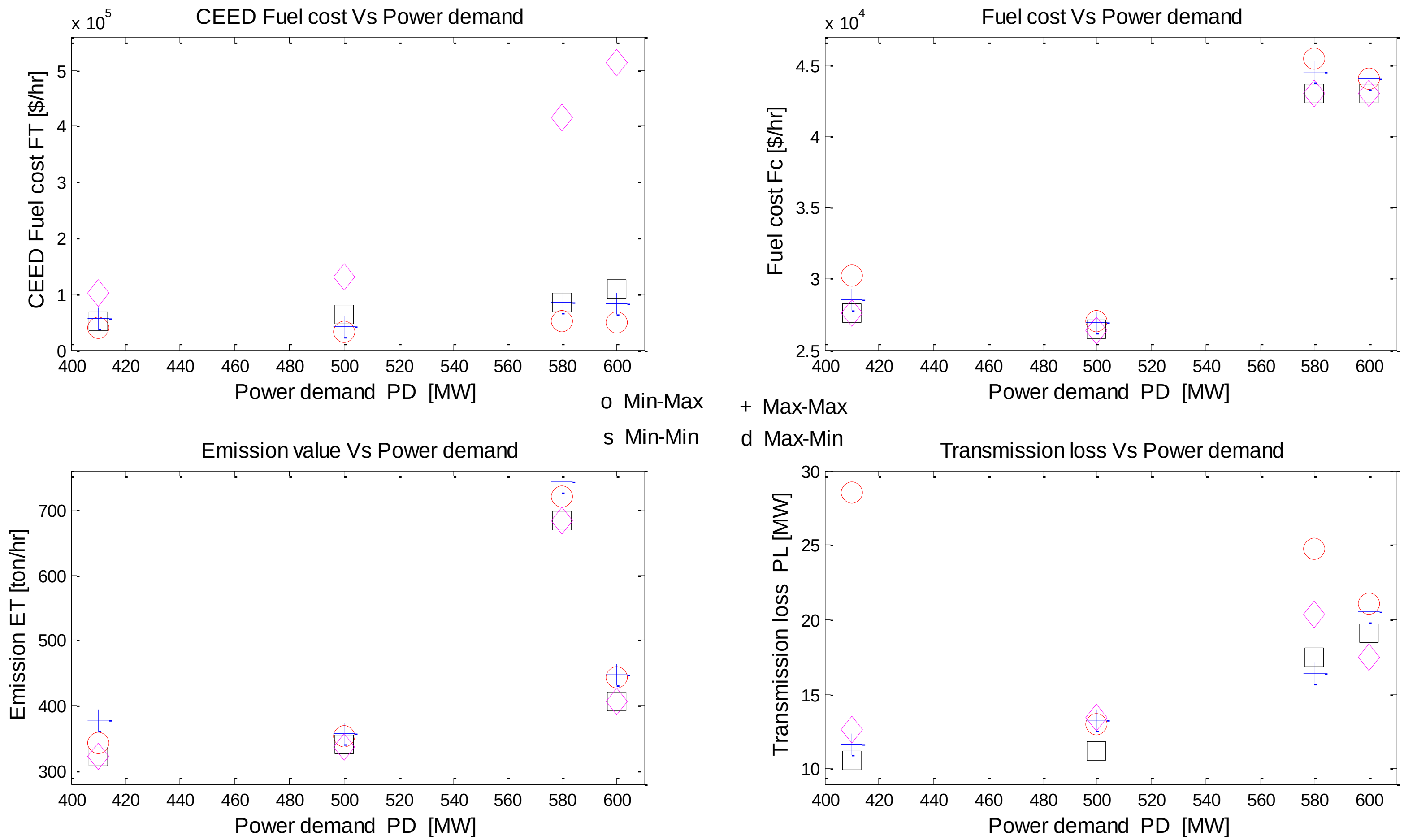
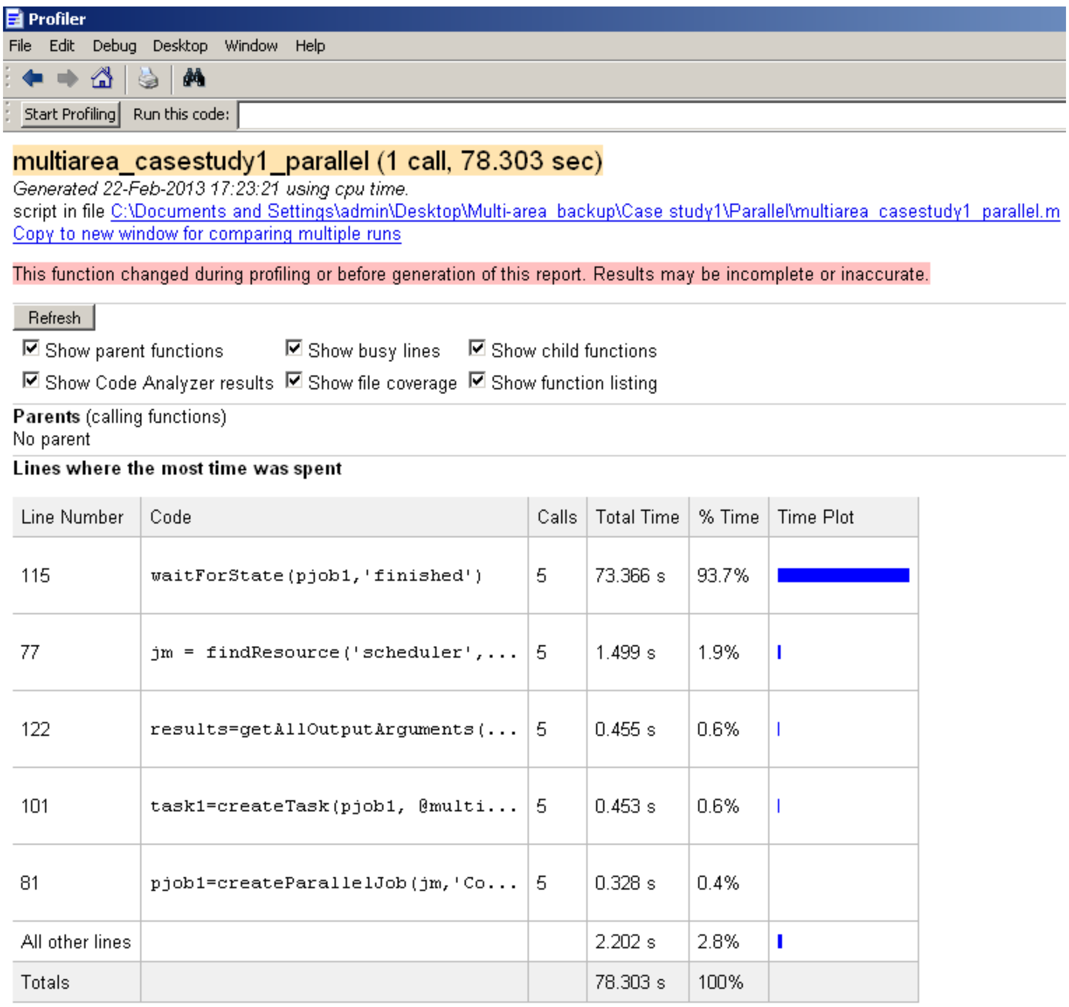


Figure 7.8: **Criteria functions fuel cost, CEED fuel cost and transmission losses of the four-area three-generator MAEED problem solution**

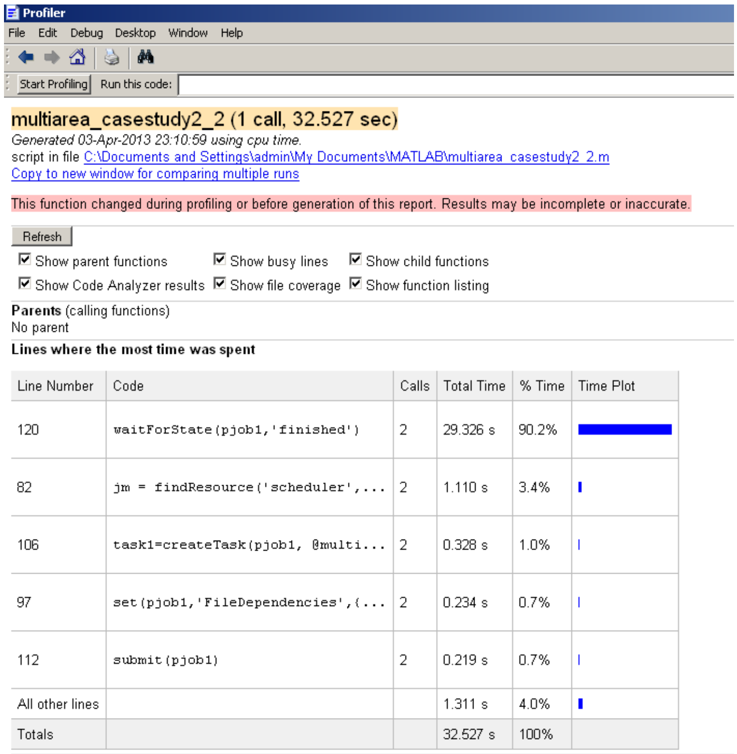
7.7 Profile viewer of the Matlab distributed computing engine

The profile viewer is a specific software in Matlab parallel toolbox that shows the amounts of time spent on the particular line. Figure 7.9 and Figure 7.10 show the profile view of the m-file (multi_area_casestudy1.m and multi_area_casestudy2.m) which are given in *Appendix J*. It is noted that the Matlab parallel toolbox function *waitForState* consumes 93.7% and 90.20% of the total computational time used respectively for the both considered cases of the total computational time used. This is due to the time spent by the Job Manager in order to perform communication between the head node and workers. Hence it is advisable to use the Matlab PCT and MDCE, if only the task is big and the computation time used to solve the task is more than the communication time.



Line Number	Code	Calls	Total Time	% Time	Time Plot
115	`waitForState(pjob1,'finished')`	5	73.366 s	93.7%	
77	`jm = findResource('scheduler',...`	5	1.499 s	1.9%	
122	`results=getAllOutputArguments(...`	5	0.455 s	0.6%	
101	`task1=createTask(pjob1, @multi...`	5	0.453 s	0.6%	
81	`pjob1=createParallelJob(jm,'Co...`	5	0.328 s	0.4%	
All other lines			2.202 s	2.8%	
Totals			78.303 s	100%	

Figure 7.9: **Profile view of the m-file (multi_area_casestudy1.m)**


Profiler
File Edit Debug Desktop Window Help
Start Profiling Run this code:

multiarea_casestudy2_2 (1 call, 32.527 sec)
Generated 03-Apr-2013 23:10:59 using cpu time.
script in file C:\Documents and Settings\admin\My Documents\MATLAB\multiarea_casestudy2_2.m
Copy to new window for comparing multiple runs

This function changed during profiling or before generation of this report. Results may be incomplete or inaccurate.

Refresh
☑ Show parent functions ☑ Show busy lines ☑ Show child functions
☑ Show Code Analyzer results ☑ Show file coverage ☑ Show function listing

Parents (calling functions)
No parent

Lines where the most time was spent

Line Number	Code	Calls	Total Time	% Time	Time Plot
120	`waitForState(pjob1,'finished')`	2	29.326 s	90.2%	
82	`jm = findResource('scheduler',...`	2	1.110 s	3.4%	
106	`task1=createTask(pjob1, @multi...`	2	0.328 s	1.0%	
97	`set(pjob1,'FileDependencies',(...`	2	0.234 s	0.7%	
112	`submit(pjob1)`	2	0.219 s	0.7%	
All other lines			1.311 s	4.0%	
Totals			32.527 s	100%	

Figure 7.10: Profile view of the m-file (multi_area_casestudy2.m)

7.8 Discussion on the results for the multi-area economic emission dispatch problem solution

Comparison of single and multi-area dispatch problem solutions is given in Table 7.11 and 7.22 for the two considered power systems of four-area forty-generator and four-area three-generator power systems. It is proved that fuel cost of a MAEED problem solution is bigger in comparison with the single area CEED problem solution. The fuel cost of both single and multi-area are not same. It is accepted that amount of tie-line power flows in multi-area system tends to increase the initial cost and operation cost in comparison with the single area one.The decomposition-coordinating method of solution allows the multi-criteria dispatch optimisation problem to be solved as a group of bi-criteria optimisation sub-problem. This solution is in parallel and is coordinated to obtain the solution of the multi-criteria problem.

7.9 Conclusion

The Multi Area Economic Emission Dispatch (MAEED) problem is solved using the Lagrange's decomposition method in this chapter. Large interconnected power systems (Multi Area) are usually decomposed into areas or zones based on criteria, such as the size of the electric power system, network topology and geographical location. MAEED problem is an optimisation task in power system operation for allocating amount of generation to the committed units within these areas. Its objective is to minimize the fuel cost subject to the power balance, generator limit, and transmission line and tie-line constraints. The solution of the MAEED problem determines the amount of power that can be economically generated in the areas and transferred to other areas if it is needed without violating tie-line capacity constraints and the whole power network constraints.

A cluster of computers working in Matlab software environment is used to implement the optimisation algorithms. Parallelization of the solution is done through decomposition of the MAEED problem according to the power system interconnected areas and coordination of the obtained solutions for every area by a coordinator. Classical (Lagrange's) decomposition-coordinating method is developed and implemented for two case studies using four-area forty-generator and four-area three-generator systems respectively. The developed Lagrange's algorithm implementations for both case studies using various price penalty factors are shown in Figure 7.7 and 7.8 respectively. The comparison of the developed Lagrange's single and multi-area solutions and multi-area PSO (Chen and Wang, 2010) solutions is given in Table 7.11 and 7.22 . It concludes that Lagrange's algorithm using Min-Max Price penalty factor method provides best solution for both single and multi-area dispatch problem in comparison with the PSO solution and other types of price penalty factors.

It is proved that the demonstration of the benefits of using Cluster of Computers for the power system optimisation problems is to obtain the reduced computation time for the calculation of the MAEED problem solution. The problem formulation allows every area to determine its own cost for the power production. The impact of the cost can be seen immediately through solution of the MAEED problem. The approach to the solution of the MAEED problem and the experience with it support the process of deregulation of the power system. The developed new methods, algorithms and software are the part of the set of energy management optimisation problems and are necessary for development and building of the Smart grid in South Africa.

Chapter 8 presents the aim, objectives, deliverables of the thesis, future work and publications in connection with the thesis.

CHAPTER EIGHT

CONCLUSION AND FUTURE RECOMMENDATIONS

8.1 Introduction

The purpose of this thesis is to develop methods, algorithms and software for single area and multi-area economic dispatch problem solution in order to reduce the fuel cost and emission of the coal based power plants.

The review paper (Krishnamurthy and Tzoneva, 2012) investigates the various methods and algorithms for the both single and multi-area economic dispatch problems. Many ongoing researches are focusing on development of new algorithms for the economic dispatch problem. Most of the existing papers compared their solutions with any one of the following types of algorithms (i) classical or (ii) heuristic or (iii) hybrid types. This thesis developed both classical (Lagrange's based) and heuristic (PSO based) algorithms for the combined economic emission dispatch problem solution. In Chapter four, the solution of the economic dispatch problem using Lagrange's and PSO algorithms are compared and it is concluded that the classical method provides better solution for the CEED problem in comparison with the heuristic method.

In Chapter three, various types of problem formulation such as Combined Economic Emission Dispatch (CEED) Problem with and without valve point effect and cubic function model are solved using various types of price penalty factors.

The electrical grid is an interconnected network for delivering electricity from the power plants to the loads. In real world scenario these power plants are geographically separated and the network is very complex. The corresponding economic dispatch problem is also very complex because it is characterized with large number of interconnections, constraints, tie-lines and loads. The conditions of deregulations requires the complex network to be considered as a set of separated but interconnected areas. This fact leads to new structure of the problem for the economic dispatch which in this case looks for two type of solutions for the areas and for the whole system as a set of the area's solutions. The recent publications focus on the Multi Area Economic Emission Dispatch (MAEED) problem but the proposed algorithms for solution are based on a sequential implementation, where the economic dispatch problem is solved as a whole. This means that the proposed algorithms do not take advantage of the changed structure of the considered problem. To be capable to develop the smarter electrical grid, it is necessary to solve the MAEED problem using decomposition technique. This thesis formulated the mathematical problem and developed Lagrange's decomposition-coordinating method and algorithm to solve the MAEED problem in a parallel way using a Cluster of Computers.

This chapter describes the aim and objectives of the thesis in part 8.2, thesis delivarables in 8.3, application of the results in 8.4, future research direction and author's publications in part 8.5 and 8.6 respectively.

8.2 Aim and objectives of the thesis

The aim of the thesis is to develop methods, algorithms, and software for solution of the CEED problem as follows: Lagrange's and Particle Swarm Optimization methods for solution of a single area CEED problems in a sequential way. Lagrange's decomposition-coordinating method to find a solution of the multi-area economic dispatch problem in a parallel way. To develop a software for sequential and parallel implementation of the algorithms of the methods in a single and in a Cluster of Computers.

8.2.1 Objectives

i. To formulate the economic dispatch problem for the various scenarios:
   a) Single area dispatch problem with the following criteria:
      - Quadratic fuel cost and emission functions
      - Quadratic fuel cost and emission functions with valve point effect loading
      - Cubic fuel cost and emission functions
   b) Multi-area dispatch problem with the following criterion:
      - Quadratic fuel cost and emission functions
      - To apply various types of price penalty factors for both single and multi-area dispatch problems, as Min-Max, Max-Max, Min-Min, and Max-Min penalty factors.

ii. To develop a Lagrange's method and algorithm for solution of the single area economic dispatch problem

iii. To develop a PSO method and algorithm for solution of the single area economic dispatch problem

iv. To develop software for sequential (centralized) solution of the economic dispatch problem using Lagrange’s algorithm based on different criteria for the cost functions.

v. To develop software for sequential (centralized) solution of the economic dispatch problem using PSO method.

vi. To develop a Lagrange’s decomposition-coordinating method and algorithm for hierarchical, two level solution of the multi-area economic emission dispatch problem.

vii. To apply the developed methods and algorithms to standard IEEE benchmark single area and multi-area models.

viii. To develop software for parallel calculation of the multi-area economic dispatch problem on the basis of the Lagrange's decomposition-coordinating algorithm in a Cluster of Computers.

ix. To implement the developed decomposition-coordinating software in a Cluster of Computers in MATLAB software environment.

The aim and objectives of the thesis are achieved and described in the thesis chapters.

8.3 Thesis delivarables

8.3.1 Investigation and a literature review on the methods for solution of the single and multi-area economic emission dispatch problems

The literature review papers are grouped according to single and multi-area economic emission dispatch problems, price penalty factors used in the CEED problem, various types of fuel cost functions used in the CEED problem such as quadratic, cubic and with/without valve point loading effects, and various optimisation methods and algorithms used in the CEED problems such as classical, heuristic, meta-heuristic and hybrid ones. The review is described in Chapter 2 and is published in (Krishnamurthy and Tzoneva, 2012b).

8.3.2 Development of new penalty factors: Min-Max and Max-Min for the bi-criteria CEED problem

The price penalty factor is used to convert the bi-objective problem into a single objective function one. The thesis developed new price penalty factors Min-Max and Max-Min (Krishnamurthy and Tzoneva, 2013) in addition to the existing penalty factors such as Max-Max, Min-Min, average and common (Balamurugan and Subramanian, 2008). The role of all penalty factors is to transfer the physical meaning of emission criterion from weight of the emission to the fuel cost for the emission. The difference between these penalty factors is in the weight of the fuel cost for emission in the final optimal fuel cost for the generation and emission criterion. The new penalty factors are used in Chapters 3, 4,6,7 and in publications (Krishnamurthy and Tzoneva,2013), (Krishnamurthy and Tzoneva, 2012d).

8.3.3 Development of Lagrange's based methods and algorithms for solution of different types of economic dispatch problems

The Lagrange's methods and algorithms are developed to solve various types of Combined Economic Emission Dispatch (CEED) problems in Chapter 3. They are: formulated with i) quadratic fuel cost and emission functions with and without valve point loading effect, and ii) cubic fuel cost and emission functions.

In addition to that, the CEED problem is formulated using various types of price penalty factors such as Min-Max, Max-Max, Min-Min, Max-Min, Average and Common.

The developed Lagrange's Matlab code is tested and validated for various IEEE benchmark models.

8.3.4 Development of a PSO method and algorithm for solution of the single area economic emission dispatch problem with transmission loss constraint

The bi-criteria combined economic emission dispatch problem is formulated using PSO algorithm in Chapter 4. In PSO, it is accepted that the number of generators is equal to the number of the particles in the swarm. For the dispatch problem the positions of the particles represent the active power produced by the generators. The velocities are variables that have the meaning of the active power but are used to search in the constraints domain. The slack bus generator is considered as the dependent generator and its active power is calculated using Equation (4.7) and (4.12).

The developed PSO Matlab code is tested and validated for various IEEE bench mark models. The comparison of the obtained economic emission dispatch problem solutions using the developed Lagrange's and PSO algorithms is done.

8.3.5 Development of a Lagrange's and PSO softwares in Matlab environment for data-parallel calculation of the single area CEED problems

The algorithms developed for sequential calculation of the various types of economic dispatch problems in chapter 3 and 4 are applied in data-parallel in chapter 6. The data-parallel software programs on the basis of the developed Lagrange's and PSO methods and algorithms are implemented in Matlab. The programs are given in Appendix E – F and are presented in Table 8.2

8.3.6 Development of a Lagrange's method and algorithm for parallel solution of the multi-area CEED problem in a Cluster of Computers

Multi-area economic dispatch problem with tie-line constraints is formulated in Chapter 7. The Lagrange's decomposition-coordinating method and algorithm are developed for multi-area economic dispatch problem solution in Parallel Matlab environment using Cluster of Computers.

The function of Lagrange is decomposed in a number of sub-functions of Lagrange according to the number of areas by using the values of the Lagrange variables as coordinating ones. Then the initial problems is decomposed in areas sub-problem and a coordinating sub-problem. The optimal solution of the coordinating sub-problem determines the optimal solutions of the area sub-problems and the optimal solution for the initial multi-area problem.

A four area three generator and a four area four generator IEEE bench mark models are used to test and validate the results obtained by the developed software in the Matlab Cluster of Computers.

Table 8.1: Programs for Sequential calculation of the single area economic emission dispatch problem solution

Type of the criterion function used in the CEED problem	Algorithm	Type of network/ System used	Appendix/Matlab script file name
Quadratic fuel cost and emission function	Lagrange's	i. IEEE 30 bus ii. Six Generator Indian Network iii. IEEE 118 bus iv. Eleven Generator system	i. Appendix A1: CEED_Casestudy1.m ii. Appendix A2: CEED_Casestudy2.m iii. Appendix A3: CEED_Casestudy3.m iv. Appendix A4: CEED_Casestudy4.m Appendix A5: CEED_Casestudy4_funct.m
Quadratic fuel cost and emission criterion functions with valve point loading effect	Lagrange's	i. 6 generator System ii. 40 generator system	i. Appendix B1: CEEDVP_Casestudy1.m ii. Appendix B2: CEEDVP_Casestudy1_funct.m iii. Appendix B3: CEEDVP_Casestudy2.m iv. Appendix B4:

			CEEDVP_Casestudy2_funct.m
Cubic fuel cost and emission criterion functions	Lagrange's	IEEE 30 bus	Appendix C: CEEDCubic.m
Quadratic fuel cost and emission criterion functions	PSO	i. IEEE 30 bus ii. IEEE 11 Generator system	i. Appendix D1: PSO_Casestudy1.m ii. Appendix D2: PSO_Casestudy2.m
Consider the whole multi-area power system as a single area system	Lagrange's sequential	i. Four area four generator system ii. Four area three generator system	i. Appendix H1: SACEEDP_Casestudy1.m ii. Appendix H2: SACEEDP_Casestudy2.m
Literature review graphs It shows the number of publications and algorithms used in different literatures	----------------	------------------	i. Appendix K1: Number_of_Publications_Yearwise .m ii. Appendix K2: Number_of_Algorithms_used.m

8.3.7 Development of algorithms and programs in Matlab for distributed-parallel calculation of the multi-area economic emission dispatch problems

The multi-area combined economic emission dispatch problem is formulated by explicitly including the power system network areas and their inter-connections with other areas by tie-lines. Two softwares are developed. Sequential one in order to solve the whole problem as a single area problem and a decomposition-coordinating one to solve the sub-problems for the separate areas in a parallel way and to coordinate these solutions towards the whole system solution.

The developed ddecomposition method allows naturally application of the parallel computing in Cluster of Computers by which the corresponding to the areas sub-problems that are functions of the coordinating variable lambda values are solved in parallel on the first level and the coordinating problem is solved on the second level of the two level calculating structures. The algorithm and software developed for parallel calculation of multi-area combined economic and emission dispatch problem are implemented in Matlab Cluster of Computers. The programs are given in Appendix I-J and are presented in Table 8.2.

Table 8.2: **Programs for data-parallel calculation of single and for task-parallel solution of the multi-area economic emission dispatch problem solution**

Description	Algorithm	Type of network	Number of Tie – lines	Appendix/Matlab script file name
Program for data-parallel solution of the single area CEED problem	Lagrange's	i. IEEE 30 bus system ii. IEEE 118 bus system	-------------	i. Appendix E1: CEED_Casestudy1_DataParallel.m ii. Appendix E2: CEED_Casestudy1_DataParallel_funct.m iii. Appendix E3: CEED_Casestudy2_DataParallel.m iv. Appendix E4: CEED_Casestudy2_DataParallel_funct.m
Program for data-parallel solution of the single area CEED problem	PSO i. 10 particle swarms ii. 30 particle swarms	IEEE 30 bus system	-------------	i. Appendix F1: PSO_10swarms_CEED_DataParallel.m ii. Appendix F2: PSO_10swarms_CEED_DataParallel_funct.m iii. Appendix F3: PSO_30swarms_CEED_DataParallel.m iv. Appendix F4: PSO_30swarms_CEED_DataParallel_funct.m
Matlab code used to start Matlab workers from the head node	----------------	----------------------------	--------------	Appendix G Start_Cluster.m
Sequential Program for MAEED problem	Lagrange's sequential	i. Four area forty generator system ii. Four area three generator system	12 Tie lines	i. Appendix I1: MACEEDP_Sequential_Casestudy1.m ii. Appendix I2: MACEEDP_Sequential_Casestudy2.m
Program for	Lagrange's	i. Four area	12 Tie	i. Appendix J1:

task-parallel calculation of the MAEED problem	decomposition-coordinating	four generator system ii. Four area three generator system	lines	MACEEDP_Parallel_Casestudy1.m ii. Appendix J2: MACEEDP_Parallel_Casestudy1_funct.m iii. Appendix J3: MACEEDP_Parallel_Casestudy2.m iv. Appendix J4: MACEEDP_Parallel_Casestudy2_funct.m

8.4 Application of the thesis results

The developed methods, algorithms and software programs can be applied with small modifications to some industrial plants with similar characteristics, as:

- Single and multi-area Economic dispatch problem
- Optimisation of Coal-based power systems
- Power grid energy management system solutions in regional or national control centers.
- Algorithm for parallel and distributed optimisation of the power system network
- Solving optimisation problems in educational courses and post graduate research
- Smart grid applications

8.5 Future research

The future developments can be listed in the following way:

- Testing the developed algorithms in a closed loop optimisation of the disptch problem using a Cluster of Computers (CC) and a Real Time Digital Simulator (RTDS) using internet communication connection in a real-time laboratory environment.
- Testing of the developed application on the bigger scale IEEE power system models.

- Application of the developed methods, algorithms, and programs for hydro and wind power systems.
- Application of the developed methods, algorithms, and programs for integration of a hydro-thermal systems into the power grid.
- Development of PSO based methods, algorithms and software programs for multi-area dispatch problem solution in a Cluster of Computers
- The metaheuristic methods such as Ant Colony Optimisation (ACO), Artifical Bee Colony (ABC) optimization, Bacterial Foraging Optimisation (BFO) can also be incorporated.

8.6 Publication

Krishnamurthy, S., Tzoneva, R., Deivakkannu, G., and Kriger, C. 2014. Development of a Lab-Scale Data Acquisition System in LabVIEW and MATLAB Environment to Solve the Economic Dispatch Problem in Real-Time. Submitted to the 19th World Congress of the International Federation of Automatic Control, Cape Town, South Africa, 24-29 August 2014, pp 1-8.

Krishnamurthy, S., and Tzoneva, R. 2013a. Economic dispatch solution using different algorithms and softwares. The International conference on Green Computing, Communication and Conservation of Energy, ICGCE-2013, RMD Engineering College, 12-14th DEC, pp 1-6.

Krishnamurthy, S., Apostolov, A., and Tzoneva, R. 2013b. Method for a Parallel Solution of a Combined Economic Emission Dispatch Problem. PAC World Africa Conference 2013, Cape Peninsula University of Technology, Cape Town, South Africa, pp 1-28.

Krishnamurthy, S., and Tzoneva, R. 2013c. Investigation on the impact of the penalty factors over solution of the dispatch optimization problem. IEEE International Conference on Industrial Technology (ICIT),Cape Town, South Africa, pp. 851 – 860.

Krishnamurthy, S., and Tzoneva, R. 2012a. Investigation of the Methods for Single area and Multi area Optimization of a Power System Dispatch Problem. International Review of Electrical Engineering (IREE), Praise Worthy Prize, Vol.7, No.1, pp 3600 – 3627.

Krishnamurthy, S., and Tzoneva, R. 2012b. Multiobjective Dispatch Problem with Valve Point Effect Loading of Fuel Cost and Emission Criterion. International Journal of Computer and Electrical Engineering (IJCEE). Vol. 4, No. 5, pp 775-784.

Krishnamurthy, S., and Tzoneva, R. 2012c. Comparison of the Lagrange's and Particle Swarm Optimisation Solutions of an Economic Emission Dispatch Problem with

transmission constraints. IEEE International Conferences on Power Electronics, Drives and Energy Systems (PEDES). Central Power Research Institute (CPRI), Bengaluru, India, pp 1-8

Krishnamurthy, S., and Tzoneva, R. 2012d. Impact of Price Penalty Factors on the Solution of the Combined Economic Emission Dispatch Problem using Cubic Criterion Functions. IEEE Power and Energy Society General Meeting, Energy Horizons - Opportunities and Challenges, San Diego, California, USA, pp 1-9.

Krishnamurthy, S., and Tzoneva, R. 2012e. Application of the Particle Swarm Optimization Algorithm to a Combined Economic Emission Dispatch Problem using a new penalty factor. IEEE PES Power Africa 2012 – Conference and Exhibition, Johannesburg, South Africa, pp 1-7.

Krishnamurthy, S., and Tzoneva, R. 2011a. Comparative Analyses of Min-Max and Max-Max Price Penalty Factor Approaches for Multi Criteria Power System Dispatch Problem Using Lagrange's Method. International conference on recent advancements in Electrical, Electronics and Control Engineering, Sivakasi, India, pp 36-43.

Krishnamurthy, S., and Tzoneva, R. 2011b. Comparative Analyses of Min-Max and Max-Max Price Penalty Factor Approaches for Multi Criteria Power System Dispatch Problem with Valve Point Effect Loading Using Lagrange's Method. International conference on International Conference on Power and Energy Systems, Chennai, India, pp 1-7.

8.7 Conclusion

This chapter describes the aim and objectives of the thesis and addresses the thesis delivarables for development of methods, algorithms, and software programs for sequential and parallel calculation of the single and multi-area economic dispatch problems. It describes the possible way of applying with small modifications to the developed methods, algorithms and software programs to some industrial systems . The future research direction and the list of the author's publications are given.

BIBLIOGRAPHY

Abhijit Chakrabarti, and Sunita Halder. 2006. Power System Analysis: Operation and Control. PHI Learning Pvt. Ltd.

Abido, M.A., June 2006. Multiobjective Evolutionary Algorithms for Electric Power Dispatch Problem. IEEE Transactions on Evolutionary Computation. Vol. 10, NO. 3, pp.315-329.

Abimbola M. Jubril., Olusola A. Komolafe., and Kehinde O. Alawode. 2013. Solving Multi-Objective Economic Dispatch Problem Via Semidefinite Programming Semidefinite Programming. IEEE Transactions on Power Systems. Volume: PP, Issue: 99, pp. 1 – 9.

Adhinarayanan, T., and Sydulu, M. 2010. An effective non-iterative ‘‘k-logic based” algorithm for economic dispatch of generators with cubic fuel cost function. Electrical Power and Energy Systems. vol 32, pp. 539–542.

Adhinarayanan, T., and Sydulu, M. 2007. Fast and Effective Algorithm for Economic Dispatch of Cubic Fuel Cost Based Thermal Units. First International Conference on Industrial and Information Systems.

Adhinarayanan, T., and Sydulu, M. 2006. Particle Swarm Optimisation for Economic Dispatch with Cubic Fuel Cost Function. TENCON , IEEE Region 10 Conference, Hong Kong.

Ahmad Heidari., Mortazavi, S.S. 2011. Optimal Operation of a Distribution Company in Restructured Power Systems with Voltage Stability and Reliability Constraints. International Review on Modeling and Simulations, Vol. 4, No. 4, pp. 1709-1716.

Ali, Ghasemi. 2013. A fuzzified multi objective Interactive Honey Bee Mating Optimization for Environmental/Economic Power Dispatch with valve point effect. Electrical Power and Energy Systems, 49, pp. 308–321.

Alsumait, J.S., Qasem, M., Sykulski, J.K., and Al-Othman, A.K. 2010. An Improved pattern search based algorithm to solve the dynamic economic dispatch problem with valve-point effect. Energy Conversion and Management, pp 2062-2067.

Al-Sumait, J. S., Sykulski, J. K., and Al-Othman, A. K. 2008. Solution of Different Types of Economic Load Dispatch Problems Using a Pattern Search Method. Electric Power Components and Systems, Vol. 36, pp. 250–265.

Amdahl, Gene M. 1967. Validity of the single processor approach to achieving large scale computing capabilities. Proceeding AFIPS '67 (Spring) Proceedings of the April 18–20, spring joint computer conference.

Amita Mahor., Vishnu Prasad., and Saroj Rangnekar. 2009. Economic dispatch using particle swarm optimization: A review. Renewable and Sustainable Energy Reviews. Vol.13, pp.2134-2141.

Anurag, K Srivastava., and Noel, N Schulz. 2009. Applications of a real-time digital simulator in power-system education and research. Annual conference on American

Society for Engineering Education (ASEE), Auxtin, Texas, pp 1-10.

Arul, R., Ravi, G., and Velusami, S. 2013. A Chaotic self-adaptive differential harmony search algorithm based dynamic economic dispatch. Electrical Power and Energy Systems 50, pp. 85–96.

Available Transfer Capability definitions and determination.(June 1996). A frame work for determining available transfer capabilities of the inter connected transmission networks for a commercially viable electricity market, NERC Report.

Balamurugan, R., and Subramanian, S. 2007. A Simplified Re-cursive Approach to Combined Economic Emission Dispatch. Electric Power Components and Systems, Vol. 36, No. 1, pp. 17-27.

Barisal, A, K. 2013. Dynamic search space squeezing strategy based intelligent algorithm solutions to economic dispatch with multiple fuels. Electrical Power and Energy Systems, 45(1), pp. 50–59.

Barzegari, A., Barforoshi, T., Asgharpour, H., and Lesan, S. 2011. A New Algorithm Based on Particle Swarm Optimization for Solving Power Economic Dispatch Considering Valve-Point Effects and Emission Constraints. International Review on Modeling and Simulations. Vol. 4, No. 3, pp. 1303-1311.

Baskar, G., and Mohan, M.R. 2008. Security constrained economic load dispatch using improved particle swarm optimization suitable for utility system. Electrical Power and Energy Systems, Vol.30, pp. 609–613.

Basu M. 2013. Artificial bee colony optimization for multi-area economic dispatch. International Journal of Electrical Power and Energy Systems, 49, pp. 181-187.

Behrooz Parhami. 2002. Introduction to Parallel Processing Algorithms and Architectures. Kluwer Academic Publishers.

Billinton, R., and Chowdhury, Nurul A. 1988. Operating Reserve Assessment in Interconnected Generating Systems. IEEE Transaction on power Apparatus and Systems, Vol. PAS-3, No. 4, pp.1479-1487.

Billinton, R., and Allan, R.N. (1984). Reliability Evaluation of Power Systems. Longman (England)/Plenum Press, New York.

Billinton, R., Jain, A.V. 1972. Power System Spinning Reserve Determination in a multi System Configuration. IEEE Paper No. T72 402-6, IEEE Summer Power Meeting, San Francisco, California.

Changhun Lee. 2002. Distributed Shared Memory. Proceedings on the 15th CISL Winter Workshop, Kushu, Japan.

Chaturvedi , K. T., Srivastava, L., and Pandit, M. 2006. Levenberg Marquardt Algorithm based Economic Load Dispatch. IEEE Power India Conference, New Delhi.

Chawdhury, N., and Billinton, R. 1990. Risk constrained economic dispatch in interconnected generating systems. IEEE Transactions on Power Systems, November, Vol. 5, No. 4, pp.1239-1247.

Chayakulkeree, K., and Ongsakul, W. 2007. Multi-Objective power flow considering system emissions and fuzzy constraints. GMSARN international conference on sustainable development: challenges and opportunities for GMS.

Chen, Y.M., and Wang, W.S. 2009. Economic Dispatch with Environmental Considerations Using Marginal Rate of Substitution Decision Approach. Journal of Quality Vol. 16, No. 2, pp 109-118.

Chen, C. L., and Chen, N. 2001. Multi-area economic generation and reserve dispatch. 22nd IEEE Power Engineering Society International Conference on Power Industry Computer Applications(PICA), Innovative Computing for Power - Electric Energy Meets the Market, pp. 368 – 373.

Chiang, C. L. 2005. Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels. IEEE Transactions on Power Systems , Vol. 20, no. 4, pp. 1690–1699.

Ching-Tzong Su., Ying-Chang Hsiao., and Yuh-Chyuan Su. 1999. A network-flow approach for security constrained economic dispatch of power systems. International Conference on Electric Power Engineering, PowerTech Budapest.

Chowdhury, B.H. and Salfur Rahman. 1990. A review of recent advances in economic dispatch. IEEE Transactions on Power Systems, Vol .5, No. 4.

Christos A. Apostolopoulos., and George N. Korres. 2010. Real-time implementation of digital relay models using MATLAB/SIMULINK and RTDS. European Transactions on Electrical Power, John Wiley and Sons Ltd, Volume 20, Issue 3, pp. 290–305.

Chuco, B. 2013. Electrical software tools overview. CIEEP-GISEI, SINATEC-IEEE. Available online at http://eurostag.regimov.net/files/DOC2.pdf, [last accessed, 06th April 2013]

Ciornei, I., and Kyriakides, E. 2012. A GA-API Solution for the Economic Dispatch of Generation in Power System Operation. IEEE Transactions on Power Systems, Volume 27 , Issue 1, pp. 233 – 242.

Coelho, L.S., and Mariani, V.C. 2006. Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. IEEE Transactions on Power Systems , Vol. 21, no. 2, pp.989–996.

Danaraj, R. M. S., and Gajendran, F. 2005. Quadratic Programming Solution to Emission and Economic Dispatch Problems. Journal of Institution of Engineers (India), Vol. 86, pp. 129-132.

David Culler , and Singh, J.P. 1997. Parallel computer architecture.Morgan Kaufmann Publication.

David, A Patterson, and John L, Hennessy. 2012. Computer organization and design: the hardware/software interface, 3rd edition. Morgan Kaufmann, Elsevier, USA, ISBN 1-55860-428-6.

David, E, Culler., Jaswinder, Pal Singh, and Anoop, Gupta. 1999. Parallel computing architecture: a hardware/software approach. Morgan Kaufmann publishers, Inc. Sanfrancisco, Calforina, USA, ISBN -13, 978-1-55860-343-1.

Deb, K., Tewari, R., Dixit, M., and Dutta, J. 2007. Finding trade-off solutions close to KKT points using evolutionary multi-objective optimization. IIT Kanpur, Kangal Report Number 2007006.

Demirovic, N., and Tesnjak, S. 2011. Possibilities of Algebraic Modelling and Solving of Power System Optimization Problems. MIPRO, Proceedings of the 34th International Convention, pp. 771 – 776.

DeRose, L., Gallivan, K., Gallopoulos, E., Marsolf, B., Padua, D. 1995. FALCON: a MATLAB interactive restructuring compiler. In: Languages and Compilers for Parallel Computing, pp. 269–288. Springer-Verlag, New York.

Dhillon, J .S., Parti, S.C., and Kothari, D. P. 1994. Multi-objective optimal thermal power dispatch. Electrical Power & Energy Systems, Volume 16, Number 6, pp 383-389.

Dhillon, J.S., Parti, S.C., and Kothari, D.P. 1993. Stochastic economic emission load dispatch. Electric Power Systems Research, Vol. 26, pp. 179-186.

Dos Santos Coelho, L.., and Mariani, V. C., 2007. Economic dispatch optimization using hybrid chaotic particle swarm optimizer. International Conference on Systems, Man and Cybernetics, Montreal, Que, pp.1963 – 1968.

Duncan Glover, J., Mulukutla S, Sarma., and Thomas J, Overbye. 2008. Power System Analysis and Design. Thomson Corporation, 4th Edition.

Eberhant, R.C., and Kennedy, J. 1995. A new optimizer using particle swarm theory. Proceedings of the sixth International Symposium on micromachine and human science. Nagoya, Japan, pp.39-43.

El-Gallad, A.I., El-Hawary, M.E., and Sallam, A.A. 2001. Swarming of intelligent particles for solving the nonlinear constrained optimization problem. Engineering Intelligent Systems for Electrical Engineering and Communications, vol. 9, no. 3, pp.155-163.

Fragomeni, U. 2012. Direct method to multi-area economic dispatch. International Conference of advances in Power Conversion and Energy Technologies (APCET), pp. 1-5.

Gaing, Z.L. 2003. Particle Swarm Optimization to Solving the Economic Dispatch Considering the Generator Constraints. IEEE Transactions on Power Systems, VOL. 18, NO. 3, pp1187-1195.

Ganesan S., and Subramanian, S. 2011. Direct Search Method for Solving Dynamic Economic Dispatch Problems. International Review on Modeling and Simulations. Vol. 4 No. 1, pp. 93-103.

Ganesan, S., and Subramanian, S. 2010. A Novel Hybrid Method for Thermal Unit Commitment Problems. International Review on Modelling and Simulations, Vol. 3. No. 4, pp. 694-704.

Gerald, L Decker., and Aaron, D. Brooks. 1958. Valve point loading of turbines. Power Apparatus and Systems Part III. Transcations of the American Instutite of Electrical Engineers, Volume 77, Issuse 3, pp.481-484.

Gilat, A. 2004. MATLAB: An Introduction with Applications. John Wiley and Sons, 2nd Edition.

Gnanadass, R. NOV 2005. Optimal power dispatch and pricing for deregulated power industry. PhD Thesis, Pondicherry University, India.

Gnanadass, R., Venkatesh, P., Palanivelu, T. G., and Manivannan, K. 2004. Evolutionary Programming Solution of Economic Load Dispatch with Combined Cycle Co-generation Effect. IE (I) Journal, Vol 85, pp.124-128.

Gottlieb, Allan, and Almasi, George S. 1989. Highly parallel computing. Benjamin-Cummings Publishing Co., Inc, Redwood City, CA, USA , ISBN 0-8053-0177-1.

Guerrero, R.P. 2004. Differential Evolution based power dispatch algorithms, M.S. Thesis, Electrical Engineering Department, University of Puerto Rico, and Mayaguez Campus.

Gustafson, John L. 1988. Reevaluating Amdahl's law. Communications of the ACM 31 (5): 532–533.

Hadi Saadat. (1999). Power System Analysis. McGraw-Hill International Edition.

Happ, H.H. 1977. Optimal power dispatch – A comprehensive survey. IEEE Transactions on Power Apparatus and systems, Vol PAS -96, pp 1248 - 1259.

Helmick, S.D., and Shoults, R.R. 1985. A practical approach to an interim multi-area economic dispatch using limited computer resources. IEEE Transactions on Power Apparatus and Systems, Vol. PAS-104, No. 6, pp. 1400-1404.

Hemamalini, S. and Simon, S. P. 2010. Dynamic Economic Dispatch with Valve-Point Effect Using Maclaurin Series Based Lagrangian Method. International Journal of Computer Applications, Volume 1 – No. 17, pp 60-67.

Hemamalini, S., and Simon, S.P. 2009. Maclaurin series-based Lagrangian method for economic dispatch with valve-point effect. Generation Transmission & Distribution, IET, Volume 3 ,Issue9, pp859-871.

Hemamalini, S., and Sishaj P Simon. 2008. Emission constrained economic dispatch with valve-point effect using particle swarm optimization.TENCON 2008, pp.1-6.

Hollingsworth, J., Liu, K., Pauca, P .1996. Parallel Toolbox for Matlab PT v. 1.00: Manual and Reference Pages. Wake Forest University URL: www.mthcsc.wfu.edu/pt/pt.html

Hong-Shan, Zhao., Qiang, Li., Zeng-qiang, Mi., and Lei,Wang. 2006. Distributed Algorithm of Multi-region Active Load in Power System Economic Dispatch. International Conference on Power System Technology – PowerCon. pp 1-7.

Hosseinnezhad, V., and Ebrahim, B. 2013. Economic load dispatch using - PSO.

Electrical Power and Energy Systems, 49, pp. 160–169.

Hosseinnezhad, V., Hagh, M.T., and Babaei, E., 2011. Quantum particle swarm optimization for economic dispatch problem with valve-point effect. 19th Iranian Conference on Electrical Engineering (ICEE).

Hu, X., and Eberhart, R.C. 2002. Solving constrained nonlinear optimization problems with particle swarm optimization. Proceedings of the Sixth World Multiconference on Systemics, Cybernetics and Informatics, vol. v, pp. 203-206.

Immanuel Selvakumar,A., and Thanushkodi,K. 2007. A new particle swarm optimization solution to nonconvex economic dispatch problems. IEEE Transactions on Power Systems, VOL. 22, NO. 1, pp.42 -51.

Jabr, R.A. 2012. Solution to Economic Dispatching With Disjoint Feasible Regions Via Semi definite Programming. IEEE Transactions on Power Systems, Volume 27, Issue 1, pp. 572 - 573 .

Jamshidian, F., Liu, K.C., and Lugtu, R. 1983. Penalty factor calculations incorporating interchange constraints. IEEE Transactions on Power Apparatus and Systems, Vol. PAS-102, No. 11, pp. 3667-3671.

Jayabarathi, T., Ramesh, V., Kothari, D. P., Kusuma Pavan, and Mithun Thumbi. 2008. Hybrid Differential Evolution Technique for Economic Dispatch Problems. Journal of Electrical Engineering & Technology, Vol. 3, No. 4, pp. 476-483.

Jeyakumar, D. N., Venkatesh, P., and Lee, K. Y. 2007. Application of Multi Objective Evolutionary Programming to Combined Economic Emission Dispatch Problem. Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA.

Jeyakumar, D.N., Jayabarathi, T., and Raghunathan, T. 2006. Particle swarm optimization for various types of economic dispatch problems. Electrical Power and Energy Systems, No. 28, pp. 36–42.

John, E. Bentley. 2000. An introduction to Parallel Computing. System Architecture. Available online at http://www2.sas.com/proceedings/sugi25/25/sy/25p283.pdf, [last accessed, 06th April 2013]

Jong-Bae Park., Yun-Won Jeong., Joong-Rin Shin., and Kwang Y. Lee. Feb 2010. An Improved Particle Swarm Optimization for Nonconvex Economic Dispatch Problems. IEEE Transactions on Power Systems. vol. 25, no. 1, pp.156-166.

Jong-Bae Park., Ki-Song Lee., Joong-Rin Shin., and Kwang Y. Lee. Feb 2005. A Particle Swarm Optimization for Economic Dispatch With Non smooth Cost Functions. IEEE Transactions on Power Systems, vol. 20, no. 1, pp.34-42.

Kannan, K.N., and Veilumuthu, V.R. 2004. Component Model for Multi- Area Power Systems On-Line Economic Load Dispatch. International Journal of Emerging Electric Power Systems, Volume 1, Issue 2, Article 1011, pp. 1-11.

Karthikeyan, S.P., Palanisamy, K., Rani, C., Raglen, I. J., Kothari, D. P. 2009. Security constrained unit commitment problem with operational Power flow and environmental constraints. WSEAS Transactions on Power Systems, Vol. 4, Issue 4, pp. 53-56.

Kennedy, J., and Eberhart, R. C. 1997. A discrete binary version of the particle swarm algorithm. Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, vol. 5, pp. 4104-4108.

Kennedy, J., and Eberhart, R. 1995. Particle Swarm Optimization. Proceedings of the IEEE international conference of neural networks (ICNN'95), Vol.IV, pp.1942-1948.

Khamsawang, S., Wannakarn, P., and Jiriwibhakorn, S. 2010. Hybrid PSO-DE for Solving the Economic Dispatch Problem with Generator Constraints. 2nd International Conference on Computer and Automation Engineering (ICCAE), Singapore, Volume 5, pp. 135 –139.

Kim, J.Y., Jeong, H.M., Lee, H.S., and Park, J.H. 2007. PC Cluster based Parallel PSO Algorithm for Optimal Power Flow. International Conference on Intelligent Systems Applications to Power Systems (ISAP), Toki Messe, Niigata, pp. 1 – 6.

Koh, B., George, A. D., Haftka, R. T., and Fregly, B. J. 2006. Parallel asynchronous particle swarm optimization. International Journal for Numerical Methods in Engineering, No. 67, pp. 578–595.

Kothari, D.P., Dhillon, J.S. 2010. Power System Optimization. PHI learning Private limited, New Delhi.

Krishnamurthy, S., Apostolov, A., and Tzoneva, R. 2013a. Method for a Parallel Solution of a Combined Economic Emission Dispatch Problem. PAC World Africa Conference 2013, Cape Peninsula University of Technology, Cape Town, South Africa, pp 1-28.

Krishnamurthy, S., and Tzoneva, R. 2013b. Investigation on the impact of the penalty factors over solution of the dispatch optimization problem. IEEE International Conference on Industrial Technology (ICIT),Cape Town, South Africa, pp. 851 – 860.

Krishnamurthy, S., and Tzoneva, R. 2012a. Investigation of the Methods for Single area and Multi area Optimization of a Power System Dispatch Problem. International Review of Electrical Engineering (IREE), Praise Worthy Prize, Vol.7, No.1, pp 3600 – 3627.

Krishnamurthy, S., and Tzoneva, R. 2012b. Multiobjective Dispatch Problem with Valve Point Effect Loading of Fuel Cost and Emission Criterion. International Journal of Computer and Electrical Engineering (IJCEE). Vol. 4, No. 5, pp 775-784.

Krishnamurthy, S., and Tzoneva, R. 2012c. Comparison of the Lagrange's and Particle Swarm Optimisation Solutions of an Economic Emission Dispatch Problem with transmission constraints. IEEE International Conferences on Power Electronics, Drives and Energy Systems (PEDES). Central Power Research Institute (CPRI), Bengaluru, India, pp 1-8

Krishnamurthy, S., and Tzoneva, R. 2012d. Impact of Price Penalty Factors on the Solution of the Combined Economic Emission Dispatch Problem using Cubic Criterion Functions. IEEE Power and Energy Society General Meeting, Energy Horizons - Opportunities and Challenges, San Diego, California, USA, pp 1-9.

Krishnamurthy, S., and Tzoneva, R. 2012e. Application of the Particle Swarm

Optimization Algorithm to a Combined Economic Emission Dispatch Problem using a new penalty factor. IEEE PES Power Africa 2012 – Conference and Exhibition, Johannesburg, South Africa, pp 1-7.

Krishnamurthy, S., and Tzoneva, R. 2011a. Comparative Analyses of Min-Max and Max-Max Price Penalty Factor Approaches for Multi Criteria Power System Dispatch Problem Using Lagrange's Method. International conference on recent advancements in Electrical, Electronics and Control Engineering, Sivakasi, India, pp 36-43.

Krishnamurthy, S., and Tzoneva, R. 2011b. Comparative Analyses of Min-Max and Max-Max Price Penalty Factor Approaches for Multi Criteria Power System Dispatch Problem with Valve Point Effect Loading Using Lagrange's Method. International conference on International Conference on Power and Energy Systems, Chennai, India, pp 1-7.

Kulkarni, P.S., Kothari, A.G., and Kothari, D.P. 2000. Combined Economic and Emission Dispatch using Improved Back Propagation Neural Network. Int. Jour. of Electrical Machines and Power Systems, 28, pp. 31-44.

Kumar, A., Srivastava, S.C., and Singh, S.N., 2004. A zonal congestion management approach using real and reactive power rescheduling. IEEE Trans. on Power Systems, Vol. 18, pp. 554–562.

Lagarias, J.C., Reeds, J.A., Wright, M.H., and Wright, P.E. 1998. Convergence properties of the Nelder-Mead simplex method in low dimensions. SIAM J. Optim., 9, (1), pp. 112–147.

Lai, L. L., Nieh, T.Y., Vujatovic, D., Ma, Y.N., Lu, Y.P., Wang, Y.W., and Braun, H. 2005. Particle swarm optimization for economic dispatch of units with non-smooth input-output characteristic functions. Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems.

Lee, F. N., and Feng, Q. 1992. Multi-Area Unit Commitment. IEEE Transactions on Power Systems, Vol. 7, No. 2, pp. 591-599.

Lee, W.N., Jeong, Y.W., Park, J.B., Shin, J.R., and Lee, K.Y. 2007. Development of an Educational Simulator for Particle Swarm Optimization and Economic Dispatch Applications. Conference on Intelligent Systems Applications to Power Systems, ISAP, Toki Messe, Niigata, pp.1-6.

Leon K. Kirchmayer. 1958. Economic operation of power systems. John Wiley & Sons.

Lin, C.E., Chen, S.T., and Huang, C.L. 1992. A Direct Newton-Raphson Economic Dispatch. IEEE Transactions on Power Systems, Vol. 7, No. 3, pp.1149-1154.

Mahor, A., Prasad, V., and Rangnekar, S., 2009. Economic dispatch using particle swarm optimization: A review. Renewable and Sustainable Energy Reviews 13 , pp. 2134–2141.

Manikandan, B. V., Charles Raja, S., Venkatesh, P., and Kannan, P. S. 2008a. Available Transfer Capability Determination in the Restructured Electricity Market. Electric Power Components and Systems, 36, pp. 941–959.

Manikandan, B. V., Charles Raja, S., and Venkatesh, P. 2008b. Multi-Area Available

Transfer Capability Determination in the Restructured Electricity Market. Joint International Conference on Power System Technology and IEEE Power India (POWERCON), New Delhi, pp.1 – 7.

Manoharan, P.S., Kannan, P.S., Baskar, S., and Iruthayarajan, M.W. 2008. Penalty parameter-less constraint handling scheme based evolutionary algorithm solutions to economic dispatch. IET Gener. Transm. Distrib., Vol. 2, No. 4, pp. 478–490.

MATLAB® Distributed Computing Server™ 5 Installation Guide. 2011. Availabe at http://www.mathworks.com/support/product/DM/installation/ver_5_1/instructions/mdce_install.pdf

Meng, K., Wang, H. G., Dong, Z.Y., and Wong, K. P. 2010. Quantum-Inspired Particle Swarm Optimization for Valve-Point Economic Load Dispatch. IEEE Transactions On Power Systems, Vol. 25, NO. 1, pp 215-222.

Message Passing Interface (MPI) report. 1992. MPI: A message-passing interface standard. Message Passing Interface Forum, Technical Report UT-CS, 94 - 230.

Ming, Z., Xiaotong, L., Fan, Y., and Kuo, T. 2010. The multi-objective optimization model of energy-efficient scheduling based on PSO algorithm. Asia-Pacific, Power and Energy Engineering Conference (APPEEC), Chengdu, pp. 1 – 4.

Mishra, D., Shukla, A., and Kalra, P. K. 2006. OR-Neuron Based Hopfield Neural Network for Solving Economic Load Dispatch Problem. Neural Information Processing – Letters and Reviews, Vol. 10, No. 11, pp249-259.

Moler, C .1995. Why isn't There a Parallel MATLAB? The Math Works Newsletter, Spring.

Mondal, S., Bhattacharya, A., and Halder nee Dey, S. 2013. Multi-objective economic emission load dispatch solution using gravitational search algorithm and considering wind power penetration. International Journal of Electrical Power and Energy Systems, 44(1), pp. 282–292.

Muneender, E., and Kumar, D.M.V. 2009. Optimal Real and Reactive Power Dispatch for Zonal Congestion Management Problem for Multi Congestion Case Using Adaptive Fuzzy PSO. TENCON, Singapore, pp. 1 – 6.

Nanda, J., Hari, L., and Kothari, M.L., 1994. Economic Emission Load Dispatch with Line Flow Constraints using a Classical Technique. IEE Proc. of Gen. Trans. and Distribution, 141, (1), pp. 1-10.

Nasr Azadani, E., Hosseinian, S. H., Moradzadeh, B., and Hasanpor, P. 2008. Economic Dispatch in Multi-area Using Particle Swarm Optimization in Electricity market. 12th International Power System Conference, (MEPCON), Middle-East , Aswan, pp. 559 – 564.

Nidul Sinha, Chakrabarti, R., and Chattopadhyay, P. K. 2003. Evolutionary Programming Techniques for Economic Load Dispatch. IEEE Transactions on Evolutionary Computation, Vol. 7, NO. 1, pp. 83 – 94.

Nithiyananthan, T.K., Ramachandran, V. 2005. RMI based distributed model for multi–

area power system on–line economic load dispatch. Journal of Electrical Engineering, Vol. 56, No. 1-2, pp. 41–44.

Okada, K., and Asano, H. 1995. A multi-area EDC with spot pricing and economic power interchange. International Conference Proceedings of Energy Management and Power Delivery(EMPD), vol.1, pp. 12 – 17.

Ouyang, Z., and Shahidehpour, S.M. 1991. Heuristic multi-area unit commitment with economic dispatch. IEE Proceedings, Vol. 138, No. 3, pp. 242-252.

Ozyon, S., and Aydin, D. 2013. Incremental artificial bee colony with local search to economic dispatch problem with ramp rate limits and prohibited operating zones. Energy Conversion and Management, 65, pp. 397–407.

Ozyon, S., Yasar, C., Aslan, Y., and Temurtas, H. 2009. Solution to environmental economic power dispatch problems in hydrothermal power systems by using genetic algorithm. International Conference on Electrical and Electronics Engineering (ELECO).

Palanichamy C., and Babu, N. S. 2008. Analytical Solution for Combined Economic and Emissions Dispatch. Electric Power Systems Research, Vol. 78, No. 7, pp. 1129- 1137.

Pankaj Kumar. 2010. Analysis of memory architecture of parallel processing computer. Dissertation, Integral University, Lucknow.

Panta, S., Premrudeepreechacharn, S., Nuchprayoon, S., Dechthummarong, C., Janjommanit, S., and Yachiangkain, S. 2007. Optimal economic dispatch using power generation of artificial neural network. International Power Engineering Conference (IPEC), Singapore, pp.1343 – 1348.

Panuganti, R., Baskaran, M.M., Hudak, D. 2006. GAMMA: Global Arrays Meets MATLAB. Technical report OSU-CISRC-1/06-TR15, Department of Computer Science and Engineering, The Ohio State University Columbus, USA.

Park, J.B., Jeong, Y.W., Shin, J.R., and Lee, K. Y. 2010. An Improved Particle Swarm Optimization for nonconvex Economic Dispatch Problems. IEEE Transactions on Power Systems, Vol. 25, NO. 1, pp 156 - 166.

Park, J.B., Lee, K.S., Shin, J.R., and Lee, K. Y. 2005. A Particle Swarm Optimization for Economic Dispatch With Non smooth Cost Functions. IEEE Transactions on Power Systems, Vol. 20, pp 34 - 42.

Parsopoulos, K.E., and Vrahatis, M.N., 2002. Particle swarm optimization method for constrained optimization problems. Proceedings of the 2nd Euro-International Symposium on Computationa Intelligence, pp. 214-220.

Poli, R., Kennedy, J., and Blackwell, T. 2000. Particle swarm optimization an overview. Swarm Intel, No. 1, pp. 33–57.

Prasanna, S., and Somasundaram, P. 2009. Multi-area security constrained economic Dispatch by fuzzy- stochastic algorithms. Journal of Theoretical and Applied Information Technology, Vol. 5, No.14, pp. 88-94.

Quintana, V.H., Lapez, R., and Valadez, V. 1981. Constrained economic dispatch of multi-area systems using the Dantzig-Wolfe decomposition principle. IEEE Transactions on Power Apparatus and Systems, Vol. PAS-100, No. 4, pp. 2127-2137.

Raghuraman, Mudumbai., Soura, Dasgupta., and Brian, B. Cho. 2012. Distributed Control for Optimal Economic Dispatch of a Network of Heterogeneous Power Generators. IEEE Transactions on power systems, Vol. 27, NO. 4, pp. 1750 - 1760.

Ramesh, R., and Ramachandran, V. 2007. Grid Service Model for Real Time Economic Load Dispatch. Iranian Journal of Electrical and Computer Engineering, Vol. 6, NO. 1, pp. 26-30.

Rani, C., and Afshin Yazdani. 2011. Emission constrained economic dispatch using various PSO algorithm. International Journal of Power Operation and Energy Management. Vol.1, Issue 1, pp.57-63.

Rani, C., Rajesh Kumar, M., and Pavan, K. 2006. Multi-Objective Generation Dispatch using Particle Swarm Optimization. Proceedings of India International Conference on Power Electronics, pp421-424.

Ratniyomchai, T., Oonsivilai, A., Pao-La-Or, P., and Kulworawanichpong, T. 2010. Economic Load Dispatch Using Improved Harmony Search. WSEAS Transactions On Systems and Control, Issue 4, Volume 5, pp 248-257.

Ravikumar Pandi, V., Panigrahi, B. K., and Manas Kumar Mallick. 2009. Improved Harmony Search for Economic Power Dispatch. Ninth International Conference on Hybrid Intelligent Systems, pp.403-408.

Ray, T., and Liew, K.M. 2001. A swarm with an effective information sharing mechanism for unconstrained and constrained single objective optimization problems. Proceedings of the Congress on Evolutionary Computation, vol. 1, pp. 75-80.

Roman, Trobec, and Peter, Zinterhof. 2009. Parallel computing: numerics, applications and trends. Springer dordrecht heidelberg, New york, London, ISBN 978-1-84882-408-9.

Rong-Mow Jan and Nanming Chen. 1995. Application of the fast Newton-Raphson economic dispatch and reactive power/voltage dispatch by sensitivity factors to optimal power flow. IEEE Transactions on Energy Conversion, Vol. 10, no.2, pp.293- 301.

Ruchir Shah., Bhardwaj Veeravalli., and Manoj Misra. 2007. On the design of adaptive and decentralized load-balancing algorithms with load estimation for computational grid environments. IEEE Transactions on Parallel and Distributed Systems, vol. 18, no. 12, pp1675-1686.

Saber, A. Y., Yona, A., and Higashi, M. 2007. Economic Load Dispatch of Higher Order Cost Polynomials Using Modified Particle Swarm Optimization. International Conference on Industrial Informatics, Vienna, Volume 2, pp. 1203 – 1208.

Senjyu, T., Chakraborty, S., Saber, A.Y., Toyama, H., Yona, A., and Funabashi, T. 2008. Thermal Generation Scheduling Strategy Using Binary Clustered Particle Swarm Optimization Algorithm. 2nd IEEE International Conference on Power and Energy (PECON), Johor Baharu, Malaysia, pp. 872-877.

Sharma, M., Pandit, M., Srivastava, L. 2010. Multi-area economic dispatch with tie-line constraints employing evolutionary approach. International Journal of Engineering, Science and Technology, Vol. 2, No. 3, pp. 132-149.

Shi, Y., and Eberhart, R.C. 2001. Fuzzy Adaptive Paricle Swarm Optimization. IEEE International conference on Evolutionary Computation. pp.101-106.

Shubham Agrawal., Panigrahi, B.K., and Manoj kumar Tiwari. OCT 2008. Multiobjective particle swarm algorithm with fuzzy clustering for electrical power dispatch. IEEE Transactions on Evolutionary Computation. Vol.12, No.5, pp.529-541.

Somasundaram, P., and Jothi Swaroopan, N. M. 2011. Fuzzified Particle Swarm Optimization Algorithm for Multi-area Security Constrained Economic Dispatch. Electric Power Components and Systems, Vol. 24, pp. 979-990.

Srikrishna, K., and Palanichamy, C. 1991. Economic Thermal Power Dispatch with Emission Constraint. Journal of Institution of Engineers (India), Vol. 72, pp. 11-18.

Storn, R., and Price, K. 1996. Minimizing the real functions of the ICEC'96 contest by Differential Evolution. IEEE International Conference on Evolutionary Computation, pp. 842-844.

Streiffert, D. 1995. Multi-Area Economic Dispatch with Tie Line Constraints. IEEE Transactions on Power Systems, Vol. 10, No. 4, pp. 1946-1951.

Subramanian, S., and Ganesan, S. 2010. Sequential Approach with Matrix Framework for Various Types of Economic Thermal Power Dispatch Problems. Energy and Power Engineering, 2, pp. 111-121.

Sugsakarn, W., and Damrongkulkamjorn, P., 2008. Economic Dispatch with Nonsmooth Cost function using Hybrid Method. Proceedings of ECTI-CON, pp.889-892.

Tao, Z., and Jin-ding, C. 2009. A new chaotic PSO with dynamic inertia weight for economic dispatch problem. International Conference on Sustainable Power Generation and Supply, SUPERGEN '09, Nanjing, pp. 1 – 6.

Thakur, T., Sem, K., Saini, S., and Sharma, S. 2006. A Particle Swarm Optimization Solution to NO2 and SO2 Emissions for Environmentally Constrained Economic Dispatch Problem. IEEE/PES, Transmission & Distribution Conference and Exposition: Latin America.

Tiacharoen, S., Potiya, S., and Polratanasuk, P. 2010. Solving various types of economic dispatch problem using Bees algorithm. International Conference on Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), pp. 617 – 620.

Tsai, M.T., and Yen, C.W. 2010. An Improved Particle Swarm Optimization for Economic Dispatch with Carbon Tax Considerations. International Conference on Power System Technology, pp 1-6.

Tseng, C.-L., Guan, X., and Svoboda, A. J. 1998. Multi-area unit commitment for large-scale power systems. IEE Proceedings of Generation,Transmission and Distribution, Vol. 145, No. 4, pp. 415-421.

Venayagamoorthy, GK. 2005. Comparison of Power System Simulation Studies on Different Platforms – RSCAD, PSCAD/EMTDC, and Simulink SimPowerSystems. International Conference on Power Systems, Operation and Planning, Praia, Cape Verde, pp. 38-41.

Venkatesh, P., and Lee, K. Y. 2008. Multi-Objective Evolutionary Programming for Economic Emission Dispatch Problem. Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy. pp 1-8.

Venkatesh, R., Ganadass, R., and Narayana Prasad Padhy. May 2003. Comparison and Application of evolutionary programming techniques to combined economic emission dispatch with line flow constraints. IEEE Transactions on Power Systems, vol.18, No.2, pp.688-697.

Wang, C., Shahidehpour, S. M. 1992. A decomposition approach to non-linear multi-area generation scheduling with tie-line constraints using expert systems. IEEE Transactions on Power Systems, Vol. 7, No. 4, pp.1409-1418.

Wang, L., and Li, L. 2013. An effective differential harmony search algorithm for the solving non-convex economic load dispatch problems. Electrical Power and Energy Systems, 44 (1), pp. 832–843.

Wang, L., and Singh, C. 2009. Reserve-constrained multiarea environmental/economic dispatch based on particle swarm optimization with local search. Engineering Applications of Artificial Intelligence, No. 22, pp. 298–307.

Wang, L., and Singh, C. 2006. Reserve-Constrained Multiarea Environmental/Economic Dispatch Using Enhanced Particle Swarm Optimization. Systems and Information Engineering Design Symposium, Charlottesville, VA, pp. 96 – 100.

Wong, K. P., Fan, B., Chang, C.S., and Liew, A.C. 1993. Multi-objective generation dispatch using bi-criterion global optimization. IEE 2nd International Conference on Advances in Power System Control, Operation and Management, Hong Kong, pp 827-832.

Wood, A.J., and Wollenberg, B.F. 1996. Power generation operation and control. 2nd Edition , Wiley Interscience, USA.

Xingwen, Jiang., Jianzhong Zhou., Hao Wang., and Yongchuan Zhang. 2013. Dynamic environmental economic dispatch using multiobjective differential evolution algorithm with expanded double selection and adaptive random restart, International Journal of Electrical Power & Energy Systems, Volume 49, Pages 399–407

Yalcionoz, T., Altun, H., and Uzam, M. 2001. Economic dispatch solution using a genetic algorithm based on arithmetic crossover. In Proc. IEEE Proto Power Tech. Conf., Portugal.

Yang, B., Chen, Y., and Zhao, Z. 2007. Survey on Applications of Particle Swarm Optimization in Electric Power Systems. IEEE International Conference on Control and Automation, Guangzhou, China, pp.481-486.

Yee Ming Chen., and Wen-Shiang Wang. 2010. A particle swarm approach to solve environmental/economic dispatch problem. International Journal of Industrial Engineering

Computations. Vol.1, pp.157-172.

Yingvivatanapong, C., Lee, W.J., and Liu, E. 2008. Multi-Area Power Generation Dispatch in Competitive Markets. IEEE Transactions on Power Systems. Vol. 23, NO.1, pp. 196-203.

Yu, Jie., and Hang, XueLiang ., 2010. Environmental Economic Dispatch Based on Sub-Area Coordinated Optimization. International Conference on Electrical and Control Engineering, pp. 3710 – 3713.

YU, Jie., LI, Yang, and XIA, An-bang. 2009. Distributed Optimization of Generation Dispatch Schedule Considering Environmental Protection and Economic Profits. Proceedings of the China Society of Electrical Engineers, Vol.29 No.16, pp. 63-68.

Yu, S.N., and Chung, W.H. 2011. Optimal Economic Dispatch with Environmental Consideration by MPSO Method. International Conference on Consumer Electronics, Communications and Networks (CECNet), pp 4235 – 4238.

Zarei, M., Roozegar, A., Kazemzadeh, R., and Kauffmann, J.M. 2007. Two Area Power Systems Economic Dispatch Problem Solving Considering Transmission Capacity Constraints. World Academy of Science, Engineering and Technology, No. 33, pp. 147-152.

Zhu, J., and Momoh, J. A. 2001. Multi-area power systems economic dispatch using nonlinear convex network flow programming. Electric Power Systems Research, 59, pp. 13–20.

Zwe-Lee Gaing., and Rung-Fang Chang. 2006. Security-Constrained economic scheduling of generation considering generator constraints. International Conference on Power System Technology. pp.1-6.

APPENDICES

APPENDICES

APPENDIX A: MATLAB PROGRAM FOR SEQUENTIAL CALCULATION OF THE CEED PROBLEM BASED ON VARIOUS PRICE PENALTY FACTORS USING LAGRANGE'S ALGORITHM

APPENDIX A1: MATLAB script file – CEED_Casestudy1.m

```
% M-File : CEED_Casestudy1.m
%IEEE 30 BUS SIX GENERATOR SYSTEM
% ==================M-File Description==================================
% The M-File is used to solve CEED problem with various price penalty
factors using Lagrange's Algorithm
clear all
clc;
tic;
% ==Economic dispatch problem including losses and generator limits==
% ============Initial Lagrange's variable============================
lambda=10                % Initial lambda
epsilon=0.001;           % Tolerance value
alfa=0.001;              % Incremental deltalambda
n=6;                         % No of generators
m=2000;                  % Total number of iterations
PD=250                   % Power demand
% ===========Fuel cost coefficients==================================
a=[0.00375 0.01750 0.06250 0.00834 0.02500 0.02500];
b=[2.00 1.75 1.00 3.25 3.00 3.00];
c=[0 0 0 0 0 0];
% ======================== Emission Coefficients=====================
d=[0.0126 0.0200 0.0270 0.0291 0.0290 0.0271];
e=[-1.1000 -0.1000 -0.1000 -0.0050 -0.0400 -0.0055];
f=[22.983 22.313 25.505 24.900 24.700 25.300];
% ==================Generator Limits ================================
Pmin=[50 20 15 10 10 12];
Pmax=[200 80 50 35 30 40];
% ==============Transmission loss coefficients=======================
B00=[0.000014];
B01=[-0.000003 0.000021 -0.000056 0.000034 0.000015 0.000078];
B=[0.000218 0.000103 0.000009 -0.000010 0.000002 0.000027;
    0.000103 0.000181 0.000004 -0.000015 0.000002 0.000030;
    0.000009 0.000004 0.000417 -0.000131 -0.000153 -0.000107;
    0.000010 -0.000015 -0.000131 0.000221 0.000094 0.000050;
    0.000002 0.000002 -0.000153 0.000094 0.000243 -0.000000;
    0.000027 0.000030 -0.000107 0.000050 -0.000000 0.000358];
%================Lagrange's algorithm starts here ===================
for iter=1:m
    iter        % To Print the each iteration number
    % ===========Calculation of Price penalty factor=================
    h=[(a.*Pmin.^2+b.*Pmin+c)]./[(d.*Pmax.^2+e.*Pmax+f)];   %
Equation (3.7 or 3.8 or 3.9 or 3.10)
    % ==================Calculation of generator real power ==========
    E=[(diag(a+(h.*d))./lambda)+B];          % Equation (3.19)
    D=0.5*((1-(b+h.*e)./lambda-B01))';       % Equation (3.19)
    P=E\D;                             % Equation (3.21)
    %======================Generator limit checking =================
    for i=1:n
        if P(i)<=Pmin(i)
            P(i)=Pmin(i)
```

```
        elseif P(i)>=Pmax(i)
            P(i)=Pmax(i)
        else P(i)=P(i)
        end
    end
    %========== Calculation of Transmission loss====================
    PL1=B00;
    PL2=zeros(1,n);
    for i=1:n
        PL2(i)=(B01(i)*P(i));        % Equation (3.3)
    end
    PL2=sum(PL2(:,:));
    PL3=zeros(n,n);
    for i=1:n
        for j=1:n
            PL3(i,j)=P(i)*B(i,j)*P(j);  % Equation (3.3)
        end
    end
    PL3=sum(PL3(:,:));
    PL3=sum(PL3(:,:));
    PL=PL1+PL2+PL3;
    % ======================Calculation of deltalambda ==============
    deltalambda=0.0;
    deltalambda=PD+PL;
    for i=1:n
        deltalambda=deltalambda-P(i);        % Equation (3.22)
    end
    if abs(deltalambda)<=epsilon | iter>=m
        break
    else lambda=lambda+deltalambda*alfa  % Equation (3.23)
    end
end
%====== Calculation of fuel cost, emission and CEED values ==========
fuelcost=0;
emissioncost=0;
CEED=0;
for i=1:n
    fuelcost=fuelcost+a(i)*P(i)^2+b(i)*P(i)+c(i); % Equation (3.1)
    emissioncost=emissioncost+d(i)*P(i)^2+e(i)*P(i)+f(i); % Equation
(3.5)
    CEED=
CEED+a(i)*P(i)^2+b(i)*P(i)+c(i)+h(i)*(d(i)*P(i)^2+e(i)*P(i)+f(i)); %
Equation (3.6)
end
%==== Printing of Real power, Transmission loss, Fuel cost, Emission
and CEED===
lambda
P
PL
totalfuelcost=sum(fuelcost(:,:))
totalemissioncost=sum(emissioncost(:,:))
CEED=sum(CEED(:,:))
toc
%=====================================================================
```

Appendix A2: MATLAB script file – CEED_Casestudy2.m

```
% M-File CEED_Casestudy2.m
% Six Generator Indian Network
% ==================M-File Description================================
% The M-File is used to solve CEED problem with various price penalty
factors using Lagrange's Algorithm
clear all
clc;
tic;
% Economic dispatch problem including losses and generator limits
% ============Initial Lagrange's variable=============================
lambda=100;           % Initial lambda
epsilon=0.001;        % Tolerance value
alfa=0.01;            % Incremental deltalambda
n=6;                  % No of generators
m=200;                % Total number of iterations
PD=500;               % Power demand
data=[0.1525  38.5397  756.7989  0.0042  0.3277   13.8593   10  125
      0.1059  46.1592  451.3251  0.0042   0.3277  13.8593   10  150
      0.0280  40.3966  1049.9977 0.0068  -0.5455  40.2669   35  225
      0.0355  38.3055  1243.5311 0.0068   -0.5455 40.2669   35  210
      0.0211  36.3278  1685.5696 0.0046   -0.5112 42.8955   130 325
      0.0180  38.2704 1356.6592  0.0046   -0.5112 42.8955   125 315];
% ===========Fuel cost coefficients===================================
a=data(:,1);
b=data(:,2);
c=data(:,3);
% ======================== Emission Coefficients=====================
d=data(:,4);
e=data(:,5);
f=data(:,6);
% ==================Generator Limits ================================
Pmin=data(:,7);
Pmax=data(:,8);
% ==============Transmission loss coefficients=======================
B00=0;
B01=[0 0 0 0 0 0]';
B=[ 0.002022    -0.000286   -0.000534   -0.000565   -0.000454   -0.000103
    -0.000286   0.003243    0.000016    -0.000307   -0.000422   -0.000147
    -0.000533   0.000016    0.002085    0.000831    0.000023    -0.000270
    -0.000565   -0.000307   0.000831    0.001129    0.000113    -0.000295
    -0.000454   -0.000422   0.000023    0.000113    0.000460    -0.000153
    0.000108    -0.000147   -0.000270   -0.000295   -0.000153   0.000898];
for iter=1:m
    iter  % Print the itertaion number
    % ===========Calculation of Price penalty factor=================
    h=[(a.*Pmin.^2+b.*Pmin+c)]./[(d.*Pmax.^2+e.*Pmax+f)];
    % =================Calculation of generator real power ==========
    E=[(diag(a+(h.*d))./lambda)+B];
    D=0.5*((1-(b+h.*e)./lambda-B01));
    %D=0.5*((1-(b+h.*e)./lambda));
    P=E\D
    TotalP=sum(P(:,:))
    %=====================Generator limit checking =================
    for i=1:n
        if P(i)<=Pmin(i)
            P(i)=Pmin(i);
        elseif P(i)>=Pmax(i);
```

```
            P(i)=Pmax(i);
        else P(i)=P(i);
        end
    end
    %========== Calculation of Transmission loss=====================
    PL1=B00;
    PL2=zeros(1,n);
    for i=1:n
        PL2(i)=(B01(i)*P(i));
    end
    PL2=sum(PL2(:,:));
    PL3=zeros(n,n);
    for i=1:n
        for j=1:n
            PL3(i,j)=P(i)*B(i,j)*P(j);
        end
    end
    PL3=sum(PL3(:,:));
    PL3=sum(PL3(:,:));
    PL=PL1+PL2+PL3
    % ======================Calculation of deltalambda ==============
    deltalambda=0.0;
    deltalambda=PD+PL;
    for i=1:n
        deltalambda=deltalambda-P(i);
    end
    deltalambda
    if abs(deltalambda)<=epsilon | iter>=m
        break
    else lambda=lambda+deltalambda*alfa
    end
    plot (iter,lambda,'rd',iter,deltalambda,'ks')
    hold on
    legend('lambda','deltalambda')
end
% Calculation of fuel cost, emission and CEED values ================
fuelcost=0;
emissioncost=0;
ceed=0;
for i=1:n
    fuelcost=fuelcost+a(i)*P(i)^2+b(i)*P(i)+c(i);
    emissioncost=emissioncost+d(i)*P(i)^2+e(i)*P(i)+f(i);
    ceed=
ceed+a(i)*P(i)^2+b(i)*P(i)+c(i)+h(i)*(d(i)*P(i)^2+e(i)*P(i)+f(i));
end
%==Printing of Real power, Transmission loss, Fuel cost, Emission and
CEED=
lambda
P
PL
totalfuelcost=sum(fuelcost(:,:))
totalemissioncost=sum(emissioncost(:,:))
ceed=sum(ceed(:,:))
toc
%=====================================================================
```

Appendix A3: **MATLAB script file - CEED_Casestudy3.m**

```
% M-File : CEED_ CEED_Casestudy3.m
% IEEE 118 bus system
%==================M-FileDescription================================
% The M-File is used to solve CEED problem with various price penalty
factors using Lagrange's Algorithm
clear all
clc;
tic;
% ==Economic dispatch problem including losses and generator limits
% ========Initial Lagrange's variable================================
lambda=4                    % Initial lambda
epsilon=0.01;               % Tolerance value
alfa=0.001;                 % Incremental deltalambda
n=14;                       % No of generators
m=10;                       % Total number of iterations
PD=3668;                 % Power demand
%===========Fuel cost and emission coefficients=====================
Data=[0.0050   1.89    150.000   0.016 -1.500    23.333    50    1000
    0.0055   2.00    115.000   0.031   -1.820    21.022    50    1000
    0.0060   3.50    40.000    0.013   -1.249    22.050    50    1000
    0.0050   3.15    122.000   0.012   -1.355    22.983    50    1000
    0.0050   3.05    125.000   0.020   -1.900    21.313    50    1000
    0.0070   2.75    120.000   0.007   0.805     21.900    50    1000
    0.0070   3.45    70.000    0.015   -1.401    23.001    50    1000
    0.0070   3.45    70.000    0.018   -1.800    24.003    50    1000
    0.0050   2.45    130.000   0.019   -2.000    25.121    50    1000
    0.0050   2.45    130.000   0.012   -1.360    22.990    50    1000
    0.0055   2.35    135.000   0.033   -2.100    27.010    50    1000
    0.0045   1.60    200.000   0.018   -1.800    25.101    50    1000
    0.0070   3.45    70.000    0.018   -1.810    24.313    50    1000
    0.0060   3.89    45.000    0.030   -1.921    27.119    50    1000
];
% ===========Fuel cost coefficients=================================
a=Data(:,1);
b=Data(:,2);
c=Data(:,3);
% ======================== Emission Coefficients====================
d=Data(:,4);
e=Data(:,5);
f=Data(:,6);
% ==================Generator Limits ===============================
Pmin=Data(:,7);
Pmax=Data(:,8);
% ==============Transmission loss coefficients=======================
B00=[0.028738];
B01=[-0.5385      -0.2832     -0.1929     -0.2642     0.0178....
      0.0219      0.0405      0.0122      0.0140      0.0044....
      0.0327      0.2178      0.0326      0.1556];
B1_half=[
    0.004274   0.003011   0.0019242   0.002151   -0.00029   -0.0004    -0.000447
    0.003011   0.003795    0.002071   0.002091   -0.00036   -0.00053   -0.000448
    0.001924   0.002071    0.002678    0.00247   -0.00025   -0.00038   -0.000298
    0.002151   0.002091   0.0024696   0.002439   -0.00023   -0.00035   -0.000309
    -0.00029   -0.00036   -0.000247   -0.00023   0.000954   0.000366   0.0002951
     -0.0004   -0.00053   -0.000378   -0.00035   0.000366   0.001068   0.0005763
```

```
    -0.00045   -0.00045   -0.000298   -0.00031   0.000295   0.000576    0.0008092
    -0.00027   -0.00037   -0.000239   -0.00022   0.000312   0.000374     0.000337
    -0.00032   -0.00036   -0.000231   -0.00023   0.000421   0.000334    0.0003566
    -0.00069    -0.0007   -0.000467   -0.00048   0.000207   0.000249    0.0003054
    -0.00075   -0.00102   -0.000786   -0.00072   3.66E-05   0.000119    0.0001293
    -0.00195     -0.002   -0.001583    -0.0016   -0.00037   -0.00028    -0.000252
    -0.00122   -0.00184   -0.001529   -0.00135   -0.00038   -0.00029    -0.000192
    -0.00172   -0.00206   -0.001668   -0.00159   -0.00042   -0.00033   -0.000272];

B2_half=[
    -0.000272   -0.0003  -0.000694  -0.000745  -0.00195  -0.001217    -0.001718
    -0.000366   -0.0004  -0.000695  -0.001018    -0.002  -0.001844    -0.002057
    -0.000239   -0.0002  -0.000467  -0.000786  -0.00158  -0.001529    -0.001668
    -0.000223   -0.0002  -0.000475  -0.000715   -0.0016   -0.01346    -0.001588
     0.000374   0.00033  0.0002486  0.0001192  -0.00028  -0.000288    -0.000331
     0.000374   0.00033  0.0002486  0.0001192  -0.00028  -0.000288    -0.000331
     0.000337   0.00036  0.0003054  0.0001252  -0.00025  -0.000192    -0.000272
    0.0003876   0.00037  0.0002934  0.0002063  -0.00015  -0.000142    -0.000188
    0.0003746   0.00054  0.0002869  0.0001477  -0.00023  -0.000189    -0.000254
    0.0002934   0.00029  0.0006738  0.0003054  0.000121  0.0001331     9.55E-05
    0.0002063   0.00015  0.0003054  0.0008576  0.000617  0.0008179     0.000726
    -0.000152   -0.0002  0.0001212  0.0006171  0.003615   0.001839    0.0020017
    -0.000142   -0.0002  0.0001331  0.0008179  0.001839  0.0033117    0.0029414
    -0.000188   -0.0003   9.55E-05   0.000726  0.002002  0.0029414   0.0041297];
B=10^-1*[B1_half B2_half];
for iter=1:m
    iter  % Print the iteration number
    % ===========Calculation of Price penalty factor=================
    h=(a.*Pmax.^2+b.*Pmax+c)./(d.*Pmax.^2+e.*Pmax+f);
    % =================Calculation of generator real power ==========
    E=(diag(a+(h.*d))./lambda)+B;
    D=0.5*((1-(b+h.*e)./lambda-B01));
    P=E\D;
    %=====================Generator limit checking =================
    for i=1:n
        if P(i)<=Pmin(i)
            P(i)=Pmin(i);
        elseif P(i)>=Pmax(i)
            P(i)=Pmax(i);
        else P(i)=P(i);
        end
    end
    TotalP=sum(P(:,:))
    %========== Calculation of Transmission loss=====================
    PL1=B00;
    PL2=zeros(1,n);
    for i=1:n
        PL2(i)=(B01(i)*P(i));
    end
    PL2=sum(PL2(:,:));
    PL3=zeros(n,n);
    for i=1:n
        for j=1:n
            PL3(i,j)=P(i)*B(i,j)*P(j);
```

```
        end
    end
    PL3=sum(PL3(:,:));
    PL3=sum(PL3(:,:));
    PL=PL1+PL2+PL3;
    % ======================Calculation of deltalambda ==============
    deltalambda=PD+PL;
    for i=1:n
        deltalambda=deltalambda-P(i);
    end
    if abs(deltalambda)<=epsilon | iter>=m
        break
    else lambda=lambda+deltalambda*alfa
    end
end
% ========Calculation of fuel cost, emission and CEED values ========
fuelcost=0;
emissioncost=0;
CEED=0;
for i=1:n
    fuelcost=fuelcost+a(i)*P(i)^2+b(i)*P(i)+c(i);
    emissioncost=emissioncost+d(i)*P(i)^2+e(i)*P(i)+f(i);
    CEED=
CEED+a(i)*P(i)^2+b(i)*P(i)+c(i)+h(i)*(d(i)*P(i)^2+e(i)*P(i)+f(i));
end
%==Printing of Real power, Transmission loss, Fuel cost, Emission and
CEED=
TotalP=sum(P(:,:))
lambda
P
PL
totalfuelcost=sum(fuelcost(:,:))
totalemissioncost=sum(emissioncost(:,:))
CEED=sum(CEED(:,:))
toc
%======================================================================
```

Appendix A4: **MATLAB script file – CEED_Casestudy4.m**

```
% M-File: CEED_Casestudy4.m
% Eleven bus system
%=================M-FileDescription=================================
% The M-File is used to solve CEED problem with various price penalty
factors using Lagrange’s Algorithm
clear all
clc;
tic;
%optimal dispatch including losses and generator limits
lambda=4 % Intial Lagrange's variable
% ==========Fuel cost coefficients==================================
a=[0.00762 0.00838 0.00523 0.00140 0.00154 0.00177 0.00195 0.00106
0.00117 0.00089 0.00098];
b=[1.92699 2.11969 2.19196 2.01983 2.22181 1.91528 2.10681 1.99138
1.99802 2.12352 2.10487];
c=[387.85 441.62 422.57 552.5 557.75 562.18 568.39 682.93 741.22
617.83 674.61];
d=[0.00419 0.00461 0.00419 0.00683 0.00751 0.00683 0.00751 0.00355
0.00417 0.00355 0.00417];
```

```
e=[-0.67767 -0.69044 -0.67767 -0.54551 -0.4006 -0.54551 -0.40006 -
0.51116 -0.56228 -0.41116 -0.56228];
f=[33.93 24.62 33.93 27.14 24.15 27.14 24.15 30.45 25.59 30.45
25.59];
Pmin=[20 20 20 60 20 60 20 100 100 110 110];
Pmax=[250 210 250 300 210 300 215 455 455 460 465];
PD=2500;
[PD totalP lambda P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 FC ET FT iter]
=CEEDwithoutloss_Sequential_funct_11bus(Pmax,Pmin,f,e,d,c,b,a,lambda,
PD)
output=[PD totalP lambda P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 FC ET FT
iter toc ]
answ=mat2cell(output)
```

Appendix A5: MATLAB Function file – CEED_Casestudy4_funct.m

```
% M-File: CEED_Casestudy4_funct.m
% Eleven bus system
%=================M-FileDescription===============================
% The M-File is used to solve CEED problem with various price penalty
factors using Lagrange’s Algorithm
function [ PD totalP lambda P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 FC ET
FT iter] =
CEEDwithoutloss_Sequential_funct_11bus(Pmax,Pmin,f,e,d,c,b,a,lambda,P
D)
epsilon=0.001; % tolerance value
alfa=0.001; %increment value in order to increase the updated lambda
value
n=11;% No of generators
m=500;% No of iterations
for iter=1:m % Lagrange's algorithm starts here
    iter
    h=(a.*Pmax.^2+b.*Pmax+c)./(d.*Pmax.^2+e.*Pmax+f); % Price penalty
factor
    %Calculation of real power value using Lagrange's algorithm for
economic dispatch problem
    E=(diag(a+(h.*d))./lambda)
    D=0.5*((1-(b+h.*e)./lambda))'
    P=E\D
    % Check the limits of real power of the generator
    for i=1:n
        if P(i)<=Pmin(i)
            P(i)=Pmin(i);
        elseif P(i)>=Pmax(i)
            P(i)=Pmax(i);
        else P(i)=P(i);
        end
    end
    % The real power output of individual generators are assigned to
the output of the function file
    P=P';
    P1=P(1);
    P2=P(2);
    P3=P(3);
    P4=P(4);
    P5=P(5);
    P6=P(6);
    P7=P(7);
```

```
    P8=P(8);
    P9=P(9);
    P10=P(10);
    P11=P(11);
    totalP=(P1+P2+P3+P4+P5+P6+P7+P8+P9+P10+P11);
    deltalambda=PD-totalP;% Calculate the change in lambda value
i.e.,deltalambda
    if abs(deltalambda)<=epsilon | iter>=m % Check the deltalambda is
less than the tolerance value or maximum Number of iteration reached
        break
    else lambda=lambda+deltalambda*alfa % Update the new lambda
    end
end % Lagrange's algorithm ends here
FC=0;
ET=0;
FT=0;
for i=1:n
    FC=FC+a(i)*P(i)^2+b(i)*P(i)+c(i);% Calculate the Fuel Cost value
(excluding emission)
    ET=ET+d(i)*P(i)^2+e(i)*P(i)+f(i);% Calculate the Emission vaue of
thermal power plant
    FT=
FT+a(i)*P(i)^2+b(i)*P(i)+c(i)+h(i)*(d(i)*P(i)^2+e(i)*P(i)+f(i));%
Calculate the Total Fuel Cost (including emission)
end
end  % function ends here
```

APPENDIX B: MATLAB PROGRAM FOR SEQUENTIAL CALCULATION OF THE CEED PROBLEM WITH VALVE POINT LOADING EFFECT

Appendix B1: MATLAB script file – CEEDVP_Casestudy1.m

```
% M-File : CEEDVP_Casestudy1.m
% IEEE 30 bus SIX GENERATOR SYSTEM
% ==================M-File Description===============================
% The M-File is used to solve CEED problem with valve point effect
loading using Lagrange’s Algorithm
clear all
clc
tic;
global lambda Pmin Pmax n a b c alpha beta d e f gamma delta B B01
B00 h;
PD=2.86;         % Power Demand
%============Fuel cost with valve point effect coefficients==========
a=[10 10 20 10 20 10];
b=[200 150 180 100 180 150];
c=[100 120 40 60 40 100];
alpha=[15 10 10 5 5 5];
beta=[6.283 8.976 14.784 20.944 25.133 18.48];
%% Emission with valve point effect coefficients
d=[4.091 2.543 4.258 5.326 4.258 6.131];
e=[-5.554 -6.047 -5.094 -3.55 -5.094 -5.555];
f=[6.49 5.638 4.586 3.38 4.586 5.151];
gamma=[0.0002 0.0005 0.000001 0.002 0.000001 0.00001];
delta=[2.857 3.333 8.0 2.0 8.0 6.667];
%================== Transmission loss coefficients===================
B00=[0.0014];
B01=[0.010731 1.7704 -4.0645 3.8453 1.3832 5.5503]*10.^(-3);
B=[ 0.0218  0.0107  -0.00036 -0.0011   0.00055   0.0033
    0.0107  0.01704 -0.0001  -0.00179  0.00026   0.0028
    -0.0004 -0.0002   0.02459 -0.01328 -0.0118   -0.0079
    -0.0011 -0.00179 -0.01328  0.0265   0.0098    0.0045
    0.00055 0.00026 -0.0118   0.0098   0.02616  -0.0001
    0.0033  0.0028  -0.00792  0.0045  -0.00012   0.0297];
%================ Generator limits====================================
Pmin=[0.05 0.05 0.05 0.05 0.05 0.05];
Pmax=[0.5 0.6 1.0 1.2 1.0 0.6];
%% Initial values of  Lagrange's algorithm
n=6; % Number of generators
lambda=20; % Initial Lambda
P=[0.05 0.05 0.05 0.05 0.05 0.05];% Initial Real Powers
m=2000; % Total number of iterations
epsilon=0.01; % Tolerance Value
varepsilon=0.1; % Tolerance value of deltalambda
%==================Lagrange's algorithm starts here==================
for iter=1:m
    iter
    % ===========To solve the CEED Problem with valve point effect
using Lagrange's algorithm===========================================
    fsolve(@CEEDVP_Casestudy1_funct,P)       % Equation (3.41)
    P=ans;
    %==============Checking of generator limits======================
    for i=1:n
        if P(i)<=Pmin(i);
            P(i)=Pmin(i);
```

```
        elseif P(i)>=Pmax(i);
            P(i)=Pmax(i);
        else P(i)=P(i);
        end
    end
    %================== Calculation of Transmission loss=============
    PL1=B00;
    PL2=zeros(1,n);
    for i=1:n
        PL2(i)=(B01(i)*P(i));       % Equation (3.3)
    end
    PL2=sum(PL2(:,:));
    PL3=zeros(n,n);
    for i=1:n
        for j=1:n
            PL3(i,j)=P(i)*B(i,j)*P(j);      % Equation (3.3)
        end
    end
    PL3=sum(PL3(:,:));
    PL3=sum(PL3(:,:));
    PL=PL1+PL2+PL3;
    %=============== Calculation of deltalambda=====================
    deltalambda=PD+PL-sum(P(:,:))   % Equation (3.42)
    if abs(deltalambda)<=epsilon | iter>=m
        break
    else lambda=lambda+(deltalambda)
    end
end
% ============Lagrange's algorithm ends here========================
%============ Pre-allocation of fuel cost, emission and CEED========
fuelcost=zeros(1,n);
emissioncost=zeros(1,n);
CEED=zeros(1,n);
%=============== Calculation of fuel cost, emission and CEED========
for i=1:n

fuelcost(i)=c(i)*P(i)^2+b(i)*P(i)+a(i)+abs(alpha(i)*sin(beta(i)*(Pmin
(i)-P(i))));
    % Equation (3.26)
    emissioncost(i)=(10.^(-
2))*(f(i)*P(i)^2+e(i)*P(i)+d(i)+(gamma(i)*(exp(delta(i)*P(i)))));
    % Equation (3.27)

CEED(i)=(c(i)*P(i)^2+b(i)*P(i)+a(i)+abs(alpha(i)*sin(beta(i)*(Pmin(i)
-P(i)))))+(h(i)*(10.^(-
2))*(f(i)*P(i)^2+e(i)*P(i)+d(i))+gamma(i)*(exp(delta(i)*P(i))));
    % Equation (3.28)
end
%=========== Printing of Real power, Transmission loss, Fuel cost,
Emission and CEED=====================================================
P
PL
totalfuelcost=sum(fuelcost(:,:))
totalemissioncost=sum(emissioncost(:,:))
CEED=sum(CEED(:,:))
toc
%=====================================================================
```

Appendix B2: **MATLAB script file - CEEDVP_Casestudy1_funct.m**

```
% M-File: CEEDVP_Casestudy1_funct.m
% IEEE 30 bus system
% ==================M-File Description===============================
% The function is used to calculate the real power of the generators
of CEED problem with valve point loading effect using Lagrange's
Algorithm
function F = CEEDVP_Casestudy1_funct(P)
global lambda Pmin Pmax n a b c alpha beta d e f gamma delta B B01 h
;
%================ Calculation of price penalty factors (h)===========
h=zeros(1,n); % Pre-allocation of price penalty factor (h)
for i=1:n
h(i)=(c(i)*Pmin(i)^2+b(i)*Pmin(i)+a(i)+abs(alpha(i)*sin(beta(i)*(Pmin
(i)-
Pmin(i)))))/((f(i)*Pmax(i)^2+e(i)*Pmax(i)+d(i))+gamma(i)*(exp(delta(i
)*Pmax(i))));     % Equation (3.29)

%h(i)=(c(i)*Pmax(i)^2+b(i)*Pmax(i)+a(i)+abs(alpha(i)*sin(beta(i)*(Pmi
n(i)-
Pmax(i)))))/((f(i)*Pmax(i)^2+e(i)*Pmax(i)+d(i))+gamma(i)*(exp(delta(i
)*Pmax(i))));     % Equation (3.30)

%h(i)=(c(i)*Pmin(i)^2+b(i)*Pmin(i)+a(i)+abs(alpha(i)*sin(beta(i)*(Pmi
n(i)-
Pmin(i)))))/((f(i)*Pmin(i)^2+e(i)*Pmin(i)+d(i))+gamma(i)*(exp(delta(i
)*Pmin(i))));     % Equation (3.31)

%h(i)=(c(i)*Pmax(i)^2+b(i)*Pmax(i)+a(i)+abs(alpha(i)*sin(beta(i)*(Pmi
n(i)-
Pmax(i)))))/((f(i)*Pmin(i)^2+e(i)*Pmin(i)+d(i))+gamma(i)*(exp(delta(i
)*Pmin(i))));     % Equation (3.32)
    %h=(h1+h2+h3+h4)/(4) % Average price penalty factor , Equation
(3.33)
    %h=(h1+h2+h3+h4)/(4*n) %Common price penalty factor, % Equation
(3.34)
end
%============ Calculation of real power of the generators============
for i=1:n
    for j=1:n
        F(i)=((2*c(i)*P(i)+b(i))-(alpha(i)*cos(beta(i)*(Pmin(i)-
(i))))+(2*h(i)*f(i)*P(i))+(h(i)*e(i))+(h(i)*gamma(i)*delta(i))*(exp(d
elta(i))))+lambda*(2*P(j)*B(i,j)+B01(i))-lambda;    % Equation (3.46)

    end
end
end
%====================================================================
```

Appendix B3: **MATLAB script file – CEEDVP_Casestudy2.m**

```
% M-File: CEEDVP_Casestudy2.m
% ==================M-File Description===============================
% The m-file is used to solve the economic dispatch problem of 40
generator system with valve point loading effect using Lagrange's
Algorithm by neglecting transmission loss and emission function
clear all
```

```
clc
tic;
global lambda Pmin n a b  alpha beta;
PD=10500        % Power Demand
%======= Fuel cost with valve point loading effect coefficients======
a=[0.0069   0.0069  0.02028 0.00942 0.0114  0.01142 0.00357 0.00492
0.00573 0.00605 0.00515 0.00569 0.00421 0.00752 0.00708 0.00708
0.00313 0.00313 0.00313 0.00313 0.00298 0.00298 0.00284 0.00284
0.00277 0.00277 0.52124 0.52124 0.52124 0.0114  0.0016  0.0016
0.0016  0.0001  0.0001  0.0001  0.00161 0.00161 0.00161 0.00313];

b=[6.73 6.73    7.07    8.18    5.35    8.05    8.03    6.99    6.6
12.9    12.9    12.8    12.5    8.84    9.15    9.15    7.97    7.95
7.97    7.97    6.63    6.63    6.66    6.66    7.1 7.1 3.33    3.33
3.33    5.35    6.43    6.43    6.43    8.95    8.62    8.62    5.88
5.88    5.88    7.97];

c=[94.705   94.705  309.54  369.03  148.89  222.33  287.71  391.98
455.76  722.82  635.2   654.69  913.4   1760.4  1728.3  1728.3
647.85  649.69  647.83  647.81  785.96  785.96  794.53  794.53
801.32  801.32  1055.1  1055.1  1055.1  148.89  222.92  222.92
222.92  107.87  116.58  116.58  307.45  307.45  307.45  647.83];

alpha=[100  100 100 150 120 100 200 200 200 200 200 200 300 300 300
300 300 300 300 300 300 300 300 300 300 300 120 120 120 120 150 150
150 200 200 200 80  80  80  300];

beta=[0.084 0.084   0.084   0.063   0.077   0.084   0.042   0.042
0.042   0.042   0.042   0.042   0.035   0.035   0.035   0.035   0.035
0.035   0.035   0.035   0.035   0.035   0.035   0.035   0.035   0.035
0.077   0.077   0.077   0.077   0.063   0.063   0.063   0.042   0.042
0.042   0.098   0.098   0.098   0.035];

%======================== Generator limits=========================
Pmin=[36    36  60  80  47  68  110 135 135 130 94  94  125 125 125
125 220 220 242 242 254 254 254 254 254 254 10  10  10  47  60  60
60  90  90  90  25  25  25  242];

Pmax=[114   114 120 190 97  140 300 300 300 300 375 375 500 500 500
500 500 500 550 550 550 550 550 550 550 550 150 150 150 97  190 190
190 200 200 200 110 110 110 550];

%==========Initial values of  Lagrange's algorithm===================
n=40;           % Number of generators
lambda=500      % Initial Lambda
% ================ Initial Real Powers===============================
P=[110  110 97  179 70  140 260 285 130 168 94  215 380 309 304 489
470 511 480 523 500 500 500 500 500 10  10  10  90  190 190 190 165
170 170 170 110 110 110 500];

m=500;      % Total number of iterations
epsilon=0.001;  % Tolerance Value
varepsilon=0.1;     % Tolerance value of delta lambda
%================= Lagrange's algorithm starts here==================
for iter=1:m
    iter
```

```
    % To solve the CEED Problem with valve point effect using
Lagrange's algorithm
    fsolve(@CEED_VP_funct_40,P)  % To call the function file
"CEED_VP_funt_40.m"    % Equation (3.52)
    P=ans;
    % ========== Checking of generator limits========================
    for i=1:n
        if P(i)<=Pmin(i);
            P(i)=Pmin(i);
        elseif P(i)>=Pmax(i);
            P(i)=Pmax(i);
        else P(i)=P(i);
        end
    end
    P
    % =====================Calculation of deltalambda================
    deltalambda=PD-sum(P(:,:))       % Equation (3.53)
    if abs(deltalambda)<=epsilon | iter>=m
        break
    else lambda=lambda+(deltalambda)
    end
end
% ==========Lagrange's algorithm ends here===========================
% ==========Pre-allocation of fuelcost, emission and CEED============
fuelcost=zeros(1,n);
CEED=zeros(1,n);
%======== Calculation of fuelcost, emission and CEED=================
for i=1:n
fuelcost(i)=a(i)*P(i)^2+b(i)*P(i)+c(i)+abs(alpha(i)*sin(beta(i)*(Pmin
(i)-P(i))));    % Equation (3.46)
end
%===Printing of Real power, Transmission loss, Fuel cost, Emission
and CEED==
P
totalfuelcost=sum(fuelcost(:,:))
toc;
ans=[lambda P totalfuelcost iter toc]
%=====================================================================
```

Appendix B4: **MATLAB script file - CEEDVP_Casestudy2_funct.m**

```
% M-File : CEEDVP_Casestudy2_funct.m
% Forty Generator System
% ==================M-File Description==============================
% The function is used to calculate the real power of the 40
generator economic dispatch problem with valve point loading effect
using Lagrange's Algorithm by neglecting transmission loss and
emission

function F= CEEDVP_Casestudy2_funct (P)
global lambda Pmin n a b alpha beta;
%% Calculation of real power of the generators
for i=1:n
    F(i)=(2*a(i)*P(i)+b(i))-(alpha(i)*cos(beta(i)*(Pmin(i)-P(i))))-
lambda;  % Equation (3.52)
end
end
%%===================================================================
```

APPENDIX C: MATLAB PROGRAM FOR SEQUENTIAL CALCULATION OF THE CEED PROBLEM WITH A CUBIC CRITERION FUNCTION

Appendix C1: MATLAB script file - CEEDCubic.m

```
% M File: CEEDCubic.m
% IEEE 30 bus system
% For all three pollutant case
% This script file solves CEED problem with cubic fuel cost and
emission function and all three emissions NOx, SO2 and CO2 are
considered.
clear all
clc;
%======TIC and TOC functions work together to measure elapsed time===
tic; % Start a stopwatch timer
%======================Fuel cost coefficients========================
a=[0.0010;0.0004;0.0006;0.0002;0.0013;0.00004];
b=[0.092;0.025;0.075;0.1;0.12;0.084];
c=[14.5;22;23;13.5;11.5;12.5];
d=[-136;-3.5;-81;-14.5;-9.75;75.6];
%======================SO2 Emission coefficients=====================
a1=[0.0005;0.0014;0.0010;0.0020;0.0013;0.0021];
b1=[0.150;0.055;0.035;0.070;0.120;0.080];
c1=[17.0;12.0;10.0;23.5;21.5;22.5];
d1=[-90.0;-30.5;-80.0;-34.5;-19.75;25.6];
%======================NOx Emission coefficients=====================
a2=[0.0012;0.0004;0.0016;0.0012;0.0003;0.0014];
b2=[0.052;0.045;0.050;0.070;0.040;0.024];
c2=[18.5;12.0;13.0;17.5;8.5;15.5];
d2=[-26.0;-35.0;-15.0;-74.0;-89.0;-75.0];
%======================CO2 Emission coefficients=====================
a3=[0.0015;0.0014;0.0016;0.0012;0.0023;0.0014];
b3=[0.092;0.025;0.055;0.010;0.040;0.080];
c3=[14.0;12.5;13.5;13.5;21.0;22.0];
d3=[-16.0;-93.5;-85.0;-24.5;-59.0;-70.0];
%======================Generator Limits==============================
Pmin=[50;20; 15 ;10 ;10 ;12];
Pmax=[200; 80; 50; 50; 50; 40];
% ============Initial Lagrange’s variable============================
PD=286.74;           % Power demad
n=6;                 % Number of generators
lambda=40;       % Initial Lagrange's variable lambda
epsilon=0.01;        % Tolerance value for deltalambda
alfa=0.01;           % Tolerance value
m=50000;            % Total Number of iterations
%===============Calculation of Min-Max Price penalty factor==========
%
h1=(a.*Pmin.^3+b.*Pmin.^2+c.*Pmin+d)./(a1.*Pmax.^3+b1.*Pmax.^2+c1.*Pm
ax+d1); % Min-Max Price penalty factor of SO2 Emission
%
h2=(a.*Pmin.^3+b.*Pmin.^2+c.*Pmin+d)./(a2.*Pmax.^3+b2.*Pmax.^2+c2.*Pm
ax+d2); % Min-Max Price penalty factor of NOx Emission
%
h3=(a.*Pmin.^3+b.*Pmin.^2+c.*Pmin+d)./(a3.*Pmax.^3+b3.*Pmax.^2+c3.*Pm
ax+d3); % Min-Max Price penalty factor of CO2 Emission


%=============Calculation of Max-Max Price penalty factor============
```

```
%h1=(a.*Pmax.^3+b.*Pmax.^2+c.*Pmax+d)./(a1.*Pmax.^3+b1.*Pmax.^2+c1.*P
max+d1); % Max-Max Price penalty factor of SO2 Emission
%h2=(a.*Pmax.^3+b.*Pmax.^2+c.*Pmax+d)./(a2.*Pmax.^3+b2.*Pmax.^2+c2.*P
max+d2);   % Max-Max Price penalty factor of NOx Emission
%h3=(a.*Pmax.^3+b.*Pmax.^2+c.*Pmax+d)./(a3.*Pmax.^3+b3.*Pmax.^2+c3.*P
max+d3);   % Max-Max Price penalty factor of CO2 Emission

%================Calculation of Min-Min Price penalty factor=========
%h1=(a.*Pmin.^3+b.*Pmin.^2+c.*Pmin+d)./(a1.*Pmin.^3+b1.*Pmin.^2+c1.*P
min+d1);          % Min-Min Price penalty factor of SO2 Emission
%h2=(a.*Pmin.^3+b.*Pmin.^2+c.*Pmin+d)./(a2.*Pmin.^3+b2.*Pmin.^2+c2.*P
min+d2);         % Min-Min Price penalty factor of NOx Emission
%h3=(a.*Pmin.^3+b.*Pmin.^2+c.*Pmin+d)./(a3.*Pmin.^3+b3.*Pmin.^2+c3.*P
min+d3);         % Min-Min Price penalty factor of CO2 Emission

%==============Calculation of Max-Min Price penalty factor===========
h1=(a.*Pmax.^3+b.*Pmax.^2+c.*Pmax+d)./(a1.*Pmin.^3+b1.*Pmin.^2+c1.*Pm
in+d1);          % Max-Min Price penalty factor of SO2 Emission
h2=(a.*Pmax.^3+b.*Pmax.^2+c.*Pmax+d)./(a2.*Pmin.^3+b2.*Pmin.^2+c2.*Pm
in+d2);          % Max-Min Price penalty factor of NOx Emission
h3=(a.*Pmax.^3+b.*Pmax.^2+c.*Pmax+d)./(a3.*Pmin.^3+b3.*Pmin.^2+c3.*Pm
in+d3);           % Max-Min Price penalty factor of CO2 Emission

%============Calculation of Average Price penalty factor=============
% h=(h1+h2+h3+h4+h5+h6)./6;

%======================Calculation of Common Price penalty factor====
% h7=(sum (h8))/6;

%======================Calculation of generator real power===========
for iter=1:m
    iter
    %% ===== Calculation of real power of generators for all three
Pollutants case=====
    x=3*(a+a1.*h1+a2.*h2+a3.*h3); % Calculation of x, equation(3.99)
    y=3*(b+b1.*h1+b2.*h2+b3.*h3); % Calculation of y, equation (3.99)
    z=(c+h1.*c1+h2.*c2+h3.*c3)-lambda;% Calculation of z,
equation(3.99)
    P =real(-y+sqrt(y.^2-4.*x.*z))./(2.*x); % Generator real power,
equation( )
    %% ===== Calculation of real power of generators for single
Pollutant case===
    %   x=3*(a+a3.*h3);          % Calculation of x, equation (3.97)
    %   y=3*(b+b3.*h3);          % Calculation of y, equation (3.97)
    %   z=(c+h3.*c3)-lambda;     % Calculation of z, equation (3.97)
    %   P =real(-y+sqrt(y.^2-4.*x.*z))./(2.*x); % Generator real
power, equation( )
    %=====================Generator limit checking =================
    for i=1:n
        if P(i)<=Pmin(i)
            P(i)=Pmin(i);
        elseif P(i)>=Pmax(i)
            P(i)=Pmax(i);
        else P(i)=P(i);
        end
    end
```

```
    %===========Calculation of deltalambda and new
lambda=============
    deltalambda=PD-sum(P)% Calculation of deltalambda using equation
()
    if abs(deltalambda)<epsilon | iter>=m       % Tolerance checking
        break
    else lambda=lambda+deltalambda*alfa % new lambda using
equation(3.102)
    end
end
TotalP=sum(P)    % Total sum of real power of all the generators

%=============Calculation of Fuel cost values======================
FC =(sum(a.*P.^3+b.*P.^2+c.*P+d)); % Fuel cost
FT1=
(sum(a.*P.^3+b.*P.^2+c.*P+d+(h1.*(a1.*P.^3+b1.*P.^2+c1.*P+d1))));
%Combined  fuel cost and SO2 emission
FT2=
(sum(a.*P.^3+b.*P.^2+c.*P+d+(h2.*(a2.*P.^3+b2.*P.^2+c2.*P+d2))));
%Combined  fuel cost and NOx emission
FT3=
(sum(a.*P.^3+b.*P.^2+c.*P+d+(h3.*(a3.*P.^3+b3.*P.^2+c3.*P+d3))));
%Combined  fuel cost and CO2 emission
FT=(sum((a.*P.^3+b.*P.^2+c.*P+d)+(h1.*(a1.*P.^3+b1.*P.^2+c1.*P+d1))+(
h2.*(a2.*P.^3+b2.*P.^2+c2.*P+d2))+(h3.*(a3.*P.^3+b3.*P.^2+c3.*P+d3)))
);

%%Combined  fuel cost and emission (CEED) of all the three pollutants
(SO2, NOx and CO2)
%=============Calculation of Emission values=========================
ET1 =(sum(a1.*P.^3+b1.*P.^2+c1.*P+d1));          % SO2 Emission
ET2 =(sum(a2.*P.^3+b2.*P.^2+c2.*P+d2));          % NOx Emission
ET3=(sum(a3.*P.^3+b3.*P.^2+c3.*P+d3));           % CO2 Emission
ET=((sum(a1.*P.^3+b1.*P.^2+c1.*P+d1))+(sum(a2.*P.^3+b2.*P.^2+c2.*P+d2
))+(sum(a3.*P.^3+b3.*P.^2+c3.*P+d3)));
% Combined emission of all the three pollutants (SO2,NOx,CO2)
%To Print [lambda, generator Real power, Fuel cost, Emission, CEED]
values
ans=[lambda P' FC ET1 ET2 ET3 ET FT1 FT2 FT3 FT iter toc]
toc;    %Read the stopwatch timer
%%===================================================================
```

APPENDIX D: MATLAB PROGRAM FOR SEQUENTIAL CALCULATION OF THE CEED PROBLEM USING PSO ALGORITHM

Appendix D1: MATLAB script file – PSO_Casestudy1.m

```
% M File: PSO_Casestudy1.m
% IEEE 30 bus system
% The M-File is used to solve CEED problem with various price penalty
factors using Particle Swarm Optimisation (PSO) Algorithm
clear all                %Removes all variables, global, functions
clc                      % Clear the command window
tic                      % Start  overall stopwatch timer
%% ======================Economic Dispatch Parameters================
%==========Load Demand and Generator limits==========================
PD=250;                     % Load demand
Pmin1=[50 20 15 10 10 12]; % Minimum generator limits including slack
bus
Pmax1=[200 80 50 35 30 40]; % Maximum generator limits including
slack bus
Pdmin=[50];                  % Minimum  generator limit of Slack bus
Pdmax=[200];                 % Maximum  generator limit of Slack bus
Pmin=[20 15 10 10 12];       % Minimum generator limits excluding
slack bus
Pmax=[80 50 35 30 40];       % Maximum generator limits excluding
slack bus

% ===================Cost and Emission coefficients=================
a=[0.00375 0.01750 0.06250 0.00834 0.02500 0.02500];
b=[2.00 1.75 1.00 3.25 3.00 3.00];
c=[0 0 0 0 0 0];
d=[0.0126 0.0200 0.0270 0.0291 0.0290 0.0271];
e=[-1.1000 -0.1000 -0.1000 -0.0050 -0.0400 -0.0055];
ff=[22.983 22.313 25.505 24.900 24.700 25.300];
n=6;                     % Number of generators

%===================Transmission loss B - coefficients==============
B=[0.000218 0.000103 0.000009 -0.000010 0.000002 0.000027;
    0.000103 0.000181 0.000004 -0.000015 0.000002 0.000030;
    0.000009 0.000004 0.000417 -0.000131 -0.000153 -0.000107;
    0.000010 -0.000015 -0.000131 0.000221 0.000094 0.000050;
    0.000002 0.000002 -0.000153 0.000094 0.000243 -0.000000;
    0.000027 0.000030 -0.000107 0.000050 -0.000000 0.000358];

%% ===============PSO Parameters==================================
wmin=0.4;                % Minimum weight
wmax=0.9;                % Maximum weight
w=0.65;                   % weight
c1=2.0;                  % Acceleration factor1
c2=2.0;                  % Acceleration factor2
Min=0;                   % Minimum Value
Max=1;                   % Maximum value
r1 = Min + (Max-Min).*rand(1,1);    %random number1 between ( 0 to 1)
r2= Min + (Max-Min).*rand(1,1);     %random number2 between ( 0 to 1)
m=25;                    % Total number of iterations

%% ================= PSO Algorithm for Economic Dispatch Problem=====
```

```
for iter=1:m                   % Number of iterations
    tic;                       % Start a stopwatch timer of PSO
    %% ===========PSO Velocity Calculation - 1st Level of solution===
    for i=1:5        % Number of velocities depends on(No of
generators of the plant)
        vmin(i)=[-0.5*Pmin(i)]       % Calculation of minimum velocity
        vmax(i)=[0.5*Pmax(i)]        % Calculation of maximum velocity
        for j=1:10  % Number of particles position (varying quantity)
            v(j,i)=[vmin(i)+rand()*(vmax(i)-vmin(i))]
                                        % calculation of velocity
        end
    end

    %% ================= Calculation of PSO Position ================
    for j=1:10                 % Number of swarm particle Positions
        for i=1:5                  % Number of velocities
            P(i)=Pmin(i)+rand()*(Pmax(i)-Pmin(i));
                                %GEN Real power- except slack bus
        end

        %% =================Generator MIN & MAX limits===============
        for i=1:5
            if P(i)<=Pmin(i)
                P(i)=Pmin(i);
            elseif P(i)>=Pmax(i)
                P(i)=Pmax(i);
            else P(i) = P(i);
            end
        end
        %% ===============Dependent Generator =======================
        P=[zeros P];                 % Pre-allocation of P
        x=B(1,1);                    % Calculation x
        y=zeros(5,1);                % Pre-allocation of y
        for k=2:6
            for r=1:1
                y(k,r)=((B(k,r)+B(r,k))*P(k));    % Calculation of y
            end
        end
        y=sum(y)-1;
        PL=zeros(5,5);               % Pre-allocation of PL
        for k=2:6
            for l=2:6
                PL(k,l)= P(k)*B(k,l)*P(l);    % Calculation of PL
            end
        end
        PL1=sum(PL);
        PL2=sum(PL1);
        PL=PL2;
        p=sum(P(2:6));
        z=PD+PL-p;                 % Calculation of z
        Pd=-(y+(y.^2- 4.*x.*z).^(1/2))/(2.*x);
                                     % Calculation of dependent GEN
        demo1=-(y-(y.^2- 4.*x.*z).^(1/2))/(2.*x);
                                     % Calculation of dependent GEN
        P=P(2:6);

        %% ====================Dependent Generator Limit ===========
        if Pd<=Pdmin          % checking the slack bus generator limit
```

```
            Pd=Pdmin;
        elseif Pd>=Pdmax
            Pd=Pdmax;
        else Pd=Pd;
        end
        p=[Pd P];            % generator power including slack bus

        %% ================= Transmission loss ======================
        for k=1:6
            for l=1:6
                PL(k,l)=p(k)*B(k,l)*p(l);
            end
        end
        Pl=sum(PL);
        PL=sum(Pl);
        eval(['PL' num2str(j) ' = (PL)'])
        eval(['p' num2str(j) ' = (p)'])

        %% ============Price Penalty Factor (PPF)====================
        h=[(a.*Pmax1.^2+b.*Pmax1+c)]./[(d.*Pmax1.^2+e.*Pmax1+ff)];

        %% ========== CEED Fuel cost, Fuel cost and Emission ========
        for i=1:6
f(i)=a(i)*p(i)^2+b(i)*p(i)+c(i)+(h(i)*(d(i)*p(i)^2+e(i)*p(i)+ff(i)));
                              % CEED Fuel cost of individual GEN
            fuel(i)=a(i)*p(i)^2+b(i)*p(i)+c(i);
                             % Fuel cost (FC)of individual GEN
            Emission(i)=d(i)*p(i)^2+e(i)*p(i)+ff(i);
                             % Emission value of individual GEN
        end

        F=sum(f);                              % CEED Fuel cost
        fuel=sum(fuel);                        % Fuel cost
        Emission=sum(Emission);                % Emission
        q=[F p PL fuel Emission ];
                    %Economic dispatch(PSO)1st level of Solution
        eval(['q' num2str(j) ' = (q)']);
        eval(['z' num2str(j) '=(q(:,1))']);
    end
    %% ========Sort the Fuel cost in ascending order=================
    p=[p1; p2; p3; p4 ;p5; p6;p7;p8;p9;p10]; %Print the real power P
    p=p(:,(2:6))                % Print the real power generator values
    z=[z1 z2 z3 z4 z5 z6 z7 z8 z9 z10];    %Total fuel cost value
    q=[q1; q2; q3 ;q4; q5; q6; q7; q8; q9; q10]
                       % Print the q=[FT, P1....Pn, FC,ET]
    [aa bb]=min(z)     % To find the minimum of FT
    gbest1=q(bb,:)     % To select the Gbest as best of all positions
    gbest=q(bb,(3:7))  % To select the global best positions

    %% % Economic dispatch (PSO) 2nd level of Solution
    %% ===================New Velocity===============================
    for j=1:10
        for i=1:5
            w(i)= wmax-((i*(wmax-wmin))/5);
                                      % Calculate the inertia weight
            vnew=w(i)*v(j,i)+(c1*r1*(p(:,i)-
p(:,i)))+(c2*r2*(gbest(:,i)-p(:,i)));   %New velocity
```

```
            eval(['vnew' num2str(i) ' = (vnew)']);
        end
    end
    vnew=[vnew1 vnew2 vnew3 vnew4 vnew5];

    %% =============check the limits of new velocities===============
    for i=1:10
        for j=1:5
            if vnew(i,j)<=vmin(j)
                vnew(i,j)=vmin(j)
            elseif vnew(i,j)>=vmax(j)
                vnew(i,j)=vmax(j)
            end
        end
    end
    %% =============== New position===================================
    pneww=zeros(10,5)           % Pre-allocation of new position
(i.e.,pneww)
    for i=1:10
        for j=1:5
            pneww(i,j)=p(i,j)+vnew(i,j); % New position
        end
        eval(['pneww' num2str(i) ' = (pneww)'])
    end

    %% ============Check the limits of new position==================
    for i=1:10
        for j=1:5
            if pneww(i,j)<=Pmin(j)
                pneww(i,j)=Pmin(j);
            elseif pneww(i,j)>=Pmax(j)
                pneww(i,j)=Pmax(j);
            end
        end
    end
    pnew=[pneww10];

    %% ===============New dependent Generator ======================
    for i=1:10
        pneww=pnew(i,:)
        pneww=[zeros pneww];
        x=B(1,1);        % calculation of x
        y=zeros(5,1);    % Pre-allocation of y
        for k=2:6
            for r=1:1
                y(k,r)=((B(k,r)+B(r,k))*pneww(k)); % Calculation of y
            end
        end
        y=sum(y)-1;
        PL=zeros(5,5); % Pre - allocation of PL
        for k=2:6
            for l=2:6
                PL(k,l)=pneww(k)*B(k,l)*pneww(l); % Calculation of PL
            end
        end
        PL11=sum(PL);
        PL22=sum(PL11);
        PL=PL22;
```

```
        p=sum(pneww(2:6));
        z=PD+PL-p;
        pdnew=-(y+(y.^2- 4.*x.*z).^(1/2))/(2*x);
                                        % New dependent Generator1
        demo1=-(y-(y.^2- 4.*x.*z).^(1/2))/(2*x);
                                        % New dependent Generator2

        %% ====================New Dependent Generator Limit =======
        if pdnew<=Pdmin % checking the slack bus generator limit
            pdnew=Pdmin;
        elseif pdnew>=Pdmax
            pdnew=Pdmax;
        else pdnew=pdnew;
        end

        %% =================== New Transmission loss ================
        pneww=pneww(2:6)      % New GEN real power excluding slack bus
        pneww=[pdnew pneww]   % New GEN real power including slack bus
        for k=1:6
            for l=1:6
                PL(k,l)=pneww(k)*B(k,l)*pneww(l);
                             % New Transmission loss
            end
        end
        Pl=sum(PL);
        PL=sum(Pl);
        eval(['PL' num2str(i) ' = (PL)']);
        eval(['pneww' num2str(i) ' = (pneww)']);
        eval(['pdnew' num2str(i) ' = (pdnew)']);
    end
    pnew=[pneww1; pneww2; pneww3; pneww4; pneww5; pneww6; pneww7;
pneww8; pneww9 ;pneww10];
    PL=[PL1 PL2 PL3 PL4 PL5 PL6 PL7 PL8 PL9 PL10]';

%%Calculate the New CEED Fuel cost,New Fuel cost and New Emission
    for j=1:10
        for i=1:6
Fnew(i)=a(i)*pnew(j,i)^2+b(i)*pnew(j,i)+c(i)+(h(i)*(d(i)*pnew(j,i)^2+
e(i)*pnew(j,i)+ff(i)));                  % New CEED Fuel cost FTnew
fuelnew(i)=a(i)*pnew(j,i)^2+b(i)*pnew(j,i)+c(i);
                                         % New Fuel cost (FCnew)
Emissionnew(i)=d(i)*pnew(j,i)^2+e(i)*pnew(j,i)+ff(i);
                                         % NewEmission ETnew
        end

        Fnew=sum(Fnew(:,:));
        fuelnew=sum(fuelnew(:,:));
        Emissionnew=sum(Emissionnew(:,:));
        eval(['Fnew' num2str(j) ' = (Fnew)']);
        eval(['fuelnew' num2str(j) ' = (fuelnew)']);
        eval(['Emissionnew' num2str(j) ' = (Emissionnew)']);
    end

%% PSO 2nd level solution; qnew=[FTnew, P1new..Pnnew, FCnew,ETnew]
Fnew=[Fnew1;Fnew2;Fnew3;Fnew4;Fnew5;Fnew6;Fnew7;Fnew8;Fnew9;Fnew10];
fuelnew=[fuelnew1;fuelnew2;fuelnew3;fuelnew4;fuelnew5;fuelnew6;fuelne
w7;fuelnew8;fuelnew9;fuelnew10;];
```

```
Emissionnew=[Emissionnew1;Emissionnew2;Emissionnew3;Emissionnew4;Emis
sionnew5;Emissionnew6;Emissionnew7;Emissionnew8;Emissionnew9;Emission
new10];
    qnew=[Fnew pnew PL fuelnew Emissionnew]

%%Compare the PSO solution of 1st level with 2nd level (q and qnew
    x=q(:,1);                      % Assign q to x
    y=qnew(:,1);                   % Assign qnew to y
    for i=1:10
        if x(i)<y(i);              % compare q and  qnew
            z(i)=y(i);
            qrow=i;
            x1=(q(qrow,:));
            mat(i,:)=x1                  % Store best solution in mat
        end
        if x(i)>y(i);                    % compare q and  qnew
            z(i)=x(i);
            qnewrow=i;
            y1=qnew(qnewrow,:);
            mat(i,:)=y1                  % Store best solution in mat
        end
    end

    %% ==================Store the best PSO solution =================
    eval(['mat' num2str(iter) ' = (mat)']);
    eval(['matz' num2str(iter) '=(mat(:,1))']);
    t(iter)=toc
end % PSO Iteration Ends

%% ==================PSO solution ==================================
final=[mat1; mat2 ;mat3 ;mat4; mat5; mat6; mat7; mat8 ;mat9; mat10
;mat11 ;mat12; mat13; mat14 ;mat15 ;mat16; mat17; mat18; mat19;
mat20;mat21;mat22;mat23;mat24;mat25];
                               % PSO Solution of all the iterations
[row col]=min(final(:,1))
% Find the best solution of PSO from the solution of each iteration
finalans=final(col,:)                % Print the best solution of PSO
iteration=fix(col/10)+1
% Print the iteration Number of the best solution

%% ========Graph: PSO solution to Economic Dispatch Problem==========
y_axis=final(:,1);
graph=plot(y_axis, '*');
hold on
plot(finalans(1,1),'o')
set(graph,'Color','red','LineWidth',2)
toc;
ANSWER=[finalans(:,:) iteration toc]
TOTALP= sum(finalans(2:7))
%% The PSO Program for CEED Problem is end here
%=================================================================
```

Appendix D2: **MATLAB script file – PSO_Casestudy2.m**

```
% M File: PSO_Casestudy2.m
% IEEE 11 Generator Systems
% The M-File is used to solve CEED problem with various price penalty
factors using Particle Swarm Optimisation (PSO) Algorithm
```

```
%====================== PSO Algorithm - 10 Positions=================
clear all
clc
tic % Start a stopwatch timer.
%for iteration=1:40 % Number of iterations

%================== Economic dispatch Parameters=====================
PD=2500; % Load demand
Pdmin=[20];% Minimum  generator limit of Slack bus
Pdmax=[250];% Maximum  generator limit of Slack bus
Pmin=[20 20 60 20 60 20 100 100 110 110];% Minimum generator limits
excluding slack bus


% ===================Cost coefficients================================
a=[0.00762 0.00838 0.00523 0.00140 0.00154 0.00177 0.00195 0.00106
0.00117 0.00089 0.00098];
b=[1.92699 2.11969 2.19196 2.01983 2.22181 1.91528 2.10681 1.99138
1.99802 2.12352 2.10487];
c=[387.85 441.62 422.57 552.5 557.75 562.18 568.39 682.93 741.22
617.83 674.61];
d=[0.00419 0.00461 0.00419 0.00683 0.00751 0.00683 0.00751 0.00355
0.00417 0.00355 0.00417];
e=[-0.67767 -0.69044 -0.67767 -0.54551 -0.4006 -0.54551 -0.40006 -
0.51116 -0.56228 -0.41116 -0.56228];
ff=[33.93 24.62 33.93 27.14 24.15 27.14 24.15 30.45 25.59 30.45
25.59];
%=====================================================================
Pmin1=[20 20 20 60 20 60 20 100 100 110 110];
                  % Minimum generator limits including slack bus
Pmax=[210 250 300 210 300 215 455 455 460 465];
                  % Maximum generator limits excluding slack bus
Pmax1=[250 210 250 300 210 300 215 455 455 460 465];
                  % Maximum generator limits including slack bus
n=11;             % Number of generators
m=20;             % Number of iterations
%===================================PSO Parameters===================
wmin=0.4;   % Minimum weight
wmax=0.9;   % Maximum weight
w=0.65;     % weight
c1=2.0;     % acceleration factor
c2=2.0;     % acceleration factor
Min=0;
Max=1;
r1 = Min + (Max-Min).*rand(1,1);%random number between ( 0 to 1)
r2= Min + (Max-Min).*rand(1,1); %random number between ( 0 to 1)

%========================PSO initial velocity========================
for iter=1:m
    tic;
    for i=1:10 % Number of velocities
        vmin(i)=[-0.5*Pmin(i)] % calculation of minimum velocity
        vmax(i)=[0.5*Pmax(i)] % calculation of maximum velocity
        for j=1:10 % Number of particles position
            v(j,i)=[vmin(i)+rand()*(vmax(i)-vmin(i))]
                                      % calculation of velocity
        end
```

```
    end
    % J loop starts here(where J= Number of particles position
    for j=1:10
        for i=1:10  % Number of velocities
            P(i)=Pmin(i)+rand()*(Pmax(i)-Pmin(i));
                        % calculation of real power except slack bus
        end
        Q=sum(P);
        Pd=[PD-Q]; % calculation of real power of slack bus

        % ====================Generator Limits checking=============
        for i=1:10 %Generators expect slack bus
            if P(i)<=Pmin(i)
                P(i)=Pmin(i);
            elseif P(i)>=Pmax(i)
                P(i)=Pmax(i);
                % checking the slack bus generator limits
                  elseif Pd(1)<=Pdmin(1)
                  Pd(1)=Pdmin(1);
                  elseif Pd(1)>=Pdmax(1)
                  Pd(1)=Pdmax(1);
            end
        end

%================================================================
        p=[Pd P]; % generator power including slack bus
        eval(['p' num2str(j) ' = (p)'])
      % print the generator power p inside the execution of for loop
h=[(a.*Pmax1.^2+b.*Pmax1+c)]./[(d.*Pmax1.^2+e.*Pmax1+ff)];
                                                  %Price penalty factor

%Calculate the Fuel cost equation with valve point effect
        for i=1:11
f(i)=a(i)*p(i)^2+b(i)*p(i)+c(i)+(h(i)*(d(i)*p(i)^2+e(i)*p(i)+ff(i)));
%Fuel cost (FT) including emission
fuel(i)=a(i)*p(i)^2+b(i)*p(i)+c(i);
                              % Fuel cost (FC) excluding emission
            Emission(i)=d(i)*p(i)^2+e(i)*p(i)+ff(i);% Emission value
        end
        F=sum(f);
        fuel=sum(fuel);
        Emission=sum(Emission);
        q=[F p fuel Emission];
        eval(['q' num2str(j) ' = (q)']);% print the q(i)=[FT(i),
P1(i)....Pn(i), FC(i),ET(i)]
        eval(['z' num2str(j) '=(q(:,1))']);%Select the FT(i) value
alone from q(i)
    end
    %===========================% J loop ends here %==================

    % Sort the Fuel cost in ascending order
    p=[p1; p2; p3; p4 ;p5; p6;p7;p8;p9;p10];%Print the real power P
    p=p(:,(2:11))% print the real power generator values
    z=[z1 z2 z3 z4 z5 z6 z7 z8 z9 z10]; %Total fuel cost value
    q=[q1; q2; q3 ;q4; q5; q6; q7; q8; q9; q10]
                              % print the q=[FT, P1....Pn, FC,ET]
    [aa bb]=min(z) % To find the minimum of FT
    gbest1=q(bb,:) % To select the Gbest as best of all positions
```

```
    gbest=q(bb,(3:12)) % To select the global best positions

    for j=1:10
        for i=1:10
            w(i)= wmax-((i*(wmax-wmin))/10);
                                    % Calculate the inertia weight
            vnew=w(i)*v(j,i)+(c1*r1*(p(:,i)-
p(:,i)))+(c2*r2*(gbest(:,i)-p(:,i))); % calculate the new velocity
            eval(['vnew' num2str(i) ' = (vnew)']);
        end
    end

 vnew=[vnew1 vnew2 vnew3 vnew4 vnew5 vnew6 vnew7 vnew8 vnew9 vnew10];
    for i=1:10
        %=============== check the limits of new velocities===========
        for j=1:10                  % Number of velocities
            if vnew(i,j)<=vmin(j)
                vnew(i,j)=vmin(j)
            elseif vnew(i,j)>=vmax(j)
                vnew(i,j)=vmax(j)
            end
        end
    end
    %================claculate the new postion========================
    for i=1:10
        for j=1:10
            pneww(i,j)=p(i,j)+vnew(i,j);
        end
        eval(['pneww' num2str(i) ' = (pneww)'])
    end
    %================check the limits of new position=================
    for i=1:10
        for j=1:10 %Generators expect slack bus
            if pneww(i,j)<=Pmin(j)
                pneww(i,j)=Pmin(j);
            elseif pneww(i,j)>=Pmax(j)
                pneww(i,j)=Pmax(j);
            end
        end
    end
    %==========calculate the new power of dependent generator========
    for i=1:10
        pdnew(i)=[PD-sum(pneww(i,:))];
        eval(['pdnew' num2str(i) ' = (pdnew)'])
    end
    pnew=[pdnew' pneww]
    %===================Calculate the FTnew,FCnew, ETnew=============
    for j=1:10
        for i=1:11
Fnew(i)=a(i)*pnew(j,i)^2+b(i)*pnew(j,i)+c(i)+(h(i)*(d(i)*pnew(j,i)^2+
e(i)*pnew(j,i)+ff(i)));
fuelnew(i)=a(i)*pnew(j,i)^2+b(i)*pnew(j,i)+c(i);
                              % Fuel cost new (FC) excluding emission
Emissionnew(i)=d(i)*pnew(j,i)^2+e(i)*pnew(j,i)+ff(i);
                                                  % Emission value new
        end

        Fnew=sum(Fnew(:,:));
```

```
        fuelnew=sum(fuelnew(:,:));
        Emissionnew=sum(Emissionnew(:,:));
        eval(['Fnew' num2str(j) ' = (Fnew)']);
        eval(['fuelnew' num2str(j) ' = (fuelnew)']);
        eval(['Emissionnew' num2str(j) ' = (Emissionnew)']);
                              %Fuel cost (FT) including emission
    end
    %=========qnew=[FTnew, P1new....Pnnew, FCnew,ETnew]==============
Fnew=[Fnew1;Fnew2;Fnew3;Fnew4;Fnew5;Fnew6;Fnew7;Fnew8;Fnew9;Fnew10];

fuelnew=[fuelnew1;fuelnew2;fuelnew3;fuelnew4;fuelnew5;fuelnew6;fuelne
w7;fuelnew8;fuelnew9;fuelnew10;];

Emissionnew=[Emissionnew1;Emissionnew2;Emissionnew3;Emissionnew4;Emis
sionnew5;Emissionnew6;Emissionnew7;Emissionnew8;Emissionnew9;Emission
new10];
    qnew=[Fnew pnew fuelnew Emissionnew]
    x=q(:,1)';
    y=qnew(:,1)';
    %================== Select the qbest solution from q and qnew=====
    for i=1:10
        if x(i)<y(i);
            z(i)=y(i);
            qrow=i;
            x1=(q(qrow,:));
            mat(i,:)=x1;
        end
        if x(i)>y(i);
            z(i)=x(i);
            qnewrow=i;
            y1=qnew(qnewrow,:);
            mat(i,:)=y1;
        end
    end
%Print the solution, when…
….iteration(i),i.e.,mat(i)=[FTbest,P1best....Pnbest,FCbest,ETbest]
    eval(['mat' num2str(iter) ' = (mat)']);
    eval(['matz' num2str(iter) '=(mat(:,1))']);
    t(iter)=toc
end

%================Print the solution, when iteration i to n
i.e.,mat=[FTbest,P1best....Pnbest,FCbest,ETbest]

final=[mat1; mat2 ;mat3 ;mat4; mat5; mat6; mat7; mat8 ;mat9; mat10
;mat11 ;mat12; mat13; mat14 ;mat15 ;mat16; mat17; mat18; mat19;
mat20];
[row col]=min(final(:,1))
finalans=final(col,:)
iteration=fix(col/10)+1
                  % print the iteration Number of the best solution
y_axis=final(:,1);
graph=plot(y_axis, '*')
hold on
plot(finalans(1,1),'o')
set(graph,'Color','red','LineWidth',2)
toc
%=====================================================================
```

APPENDIX E: MATLAB PROGRAM FOR DATA PARALLELSOLUTION OF THE CEED PROBLEM USING LAGRANGE'S ALGORITHM IN A CLUSTER OF COMPUTERS

Appendix E1: **MATLAB script file – CEED_Casestudy1_DataParallel.m**

```
% M-File : CEED_Casestudy1_DataParallel.m
%IEEE 30 BUS SIX GENERATOR SYSTEM
% Six different power demands are assigned to 6 workers using a MATLAB
parallel computing toolbox
% ==================M-File Description===============================
% The M-File is used to solve CEED problem with various price penalty
factors using Lagrange's Algorithm in a parallel way
clear all
clc;
tic;
% ==Economic dispatch problem including losses and generator limits==
% ============Initial Lagrange's variable============================
lambda=4;
% ===========Fuel cost coefficients=================================
a=[0.00375 0.01750 0.06250 0.00834 0.02500 0.02500];
b=[2.00 1.75 1.00 3.25 3.00 3.00];
c=[0 0 0 0 0 0];
% ======================== Emission Coefficients=====================
d=[0.0126 0.0200 0.0270 0.0291 0.0290 0.0271];
e=[-1.1000 -0.1000 -0.1000 -0.0050 -0.0400 -0.0055];
f=[22.983 22.313 25.505 24.900 24.700 25.300];
% ==================Generator Limits ================================
Pmin=[50 20 15 10 10 12];
Pmax=[200 80 50 35 30 40];
% ==============Transmission loss coefficients=======================
B00=[0.000014];
B01=[-0.000003 0.000021 -0.000056 0.000034 0.000015 0.000078];
B=[0.000218 0.000103 0.000009 -0.000010 0.000002 0.000027;
    0.000103 0.000181 0.000004 -0.000015 0.000002 0.000030;
    0.000009 0.000004 0.000417 -0.000131 -0.000153 -0.000107;
    0.000010 -0.000015 -0.000131 0.000221 0.000094 0.000050;
    0.000002 0.000002 -0.000153 0.000094 0.000243 -0.000000;
    0.000027 0.000030 -0.000107 0.000050 -0.000000 0.000358];
%Matlabpool open jobmanagerconfig1 %Opening 6 Matlab workers from the
cluster
t0 = clock; % start of the clock on second level calculation
jm = findResource('scheduler','configuration','jobmanagerconfig1')
disp(get(jm))%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% check the status of the
jobmanager
timingStart = tic;
start = tic;
pjob1=createParallelJob(jm,'Configuration','jobmanagerconfig1'); %create
parallel job from the jobmanager
get(pjob1) % Monitor the status of pjob1
%==========How long it takes to create a job=============================
times.jobCreateTime = toc(start)
description.jobCreateTime = 'Job creation time'
%=======================================================================
Tasks = findTask(pjob1)
[pending running finished] = findTask(pjob1)
set(pjob1,'Configuration','jobmanagerconfig1')
```

```
set(pjob1,'MinimumNumberOfWorkers',6);
set(pjob1,'MaximumNumberOfWorkers',6);
set(pjob1,'FileDependencies',{'CEED_Casestudy1_DataParallel_funct.m'})
get(jm)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Notice how jobmanager is
changing
disp(get(pjob1))
disp('busy workers')
disp(get(jm, 'NumberOfbusyWorkers'))
%=====================STARTING PARALLEL COMPUTING========================
%Create task- only one with 13 output arguments  and 12 input arguments,
%measure how long that takes and measure how long it takes to submit the
%job to the cluster
task1=createTask(pjob1,
@CEED_Casestudy1_DataParallel_funct,14,{Pmax,Pmin,f,e,d,c,b,a,lambda,B,B0
0,B01})
get(pjob1,'task')
%==========How long it takes to create a task===========================
times.taskCreateTime = toc(start)
description.taskCreateTime = 'Task creation time'
%======================================================================
submit(pjob1)
%===========How long it takes to submit a job===========================
times.submitTime = toc(start)
description.submitTime = 'Job submission time'
%======================================================================
%Once the job has been submitted, we hope all its tasks execute in
%parallel. Measure how long it takes for all the tasks to start and to
run
%to completion.
waitForState(pjob1,'finished')
times.jobWaitTime = toc(start)
description.pjobWaitTime = 'Job wait time'
get(jm)
%Tasks have now completed, so we are again executing a code in the Matlab
%client. Measure how long it takes to retrieve all the job results
disp('PD lambda P1 P2 P3 P4 P5 P6 PL FC ET FT iter time') % display the
output values
results=getAllOutputArguments(pjob1)
times.resultsTime = toc(start)
description.resultsTime = 'Results retrieval time'
%======================================================================
%============Verify that the job ran without errors=====================
errmsgs = get(pjob1.task, {'ErrorMessage'})
nonempty = ~cellfun(@isempty, errmsgs)
celldisp(errmsgs(nonempty))
%=====================================================================

%Measure the total time elapsed from creating the job up to the results
times.totalTime = toc(start)
operationtime = etime(clock,t0) %operation time for completion of
parallel computing
destroy(pjob1)
%matlabpool close% close matlapool of workers
```

Appendix E2: **MATLAB script file – CEED_Casestudy1_DataParallel_funct.m**

```
% M-File : CEED_Casestudy1_DataParallel_funct.m
%IEEE 30 BUS SIX GENERATOR SYSTEM
```

```
% ==================M-File Description===============================
% The M-File is used to solve CEED problem with various price penalty
% factors using Lagrange's Algorithm in a Parallel way
function [ PD lambda P1 P2 P3 P4 P5 P6 PL FC ET FT iter time] =
CEED_Casestudy1_DataParallel_funct(Pmax,Pmin,f,e,d,c,b,a,lambda,B,B00,B01
)
tic;
% The Power demand varies according to lab index
if labindex ==1
PD=125;
end
if  labindex==2
    PD=150;
end
if  labindex==3
    PD=175;
end
if  labindex==4
    PD=200;
end
if  labindex==5
    PD=225;
end
if labindex==6
    PD=250;
end
epsilon=0.001;          % Tolerance value
alfa=0.001;             % Incremental deltalambda
n=6;                        % No of generators
m=2000;                 % Total number of iterations
for iter=1:m
    %=================Lagrange's algorithm starts here ====================
    iter        % To Print the each iteration number
    % ===========Calculation of Price penalty factor=================
    h=[(a.*Pmin.^2+b.*Pmin+c)]./[(d.*Pmax.^2+e.*Pmax+f)];   % Equation
(3.7 or 3.8 or 3.9 or 3.10)
    % =================Calculation of generator real power ==========
    E=[(diag(a+(h.*d))./lambda)+B];         % Equation (3.19)
    D=0.5*((1-(b+h.*e)./lambda-B01))';      % Equation (3.19)
    P=E\D;                        % Equation (3.21)
    %======================Generator limit checking =================
    for i=1:n
        if P(i)<=Pmin(i)
            P(i)=Pmin(i)
        elseif P(i)>=Pmax(i)
            P(i)=Pmax(i)
        else P(i)=P(i)
        end
    end
    % The real power output of individual generators are assigned to the
output of the function file
    P1=P(1); P2=P(2); P3=P(3); P4=P(4); P5=P(5); P6=P(6);
    %========== Calculation of Transmission loss=====================
    PL1=B00;
    PL2=zeros(1,n);
    for i=1:n
        PL2(i)=(B01(i)*P(i));       % Equation (3.3)
    end
```

```
    PL2=sum(PL2(:,:));
    PL3=zeros(n,n);
    for i=1:n
        for j=1:n
            PL3(i,j)=P(i)*B(i,j)*P(j);  % Equation (3.3)
        end
    end
    PL3=sum(PL3(:,:));
    PL3=sum(PL3(:,:));
    PL=PL1+PL2+PL3;
    % ======================Calculation of deltalambda ==============
    deltalambda=0.0;
    deltalambda=PD+PL;
    for i=1:n
        deltalambda=deltalambda-P(i);       % Equation (3.22)
    end
    if abs(deltalambda)<=epsilon | iter>=m
        break
    else lambda=lambda+deltalambda*alfa  % Equation (3.23)
    end
end
%====== Calculation of fuel cost, emission and CEED values ==========
fuelcost=0;
emissioncost=0;
CEED=0;
for i=1:n
    fuelcost=fuelcost+a(i)*P(i)^2+b(i)*P(i)+c(i); % Equation (3.1)
    emissioncost=emissioncost+d(i)*P(i)^2+e(i)*P(i)+f(i); % Equation
(3.5)
    CEED=
CEED+a(i)*P(i)^2+b(i)*P(i)+c(i)+h(i)*(d(i)*P(i)^2+e(i)*P(i)+f(i)); %
Equation (3.6)
end
%==== Printing of Real power, Transmission loss, Fuel cost, Emission and
CEED===
lambda
P
PL
FC=sum(fuelcost(:,:))
ET=sum(emissioncost(:,:))
FT=sum(CEED(:,:))
time=toc;
end
%=====================================================================
```

Appendix E3: **MATLAB script file – CEED_Casestudy2_DataParallel.m**

```
% M-File : CEED_Casestudy2_DataParallel.m
% IEEE 118 bus system
%==================M-FileDescription================================
% The M-File is used to solve CEED problem with various price penalty
factors using Lagrange's Algorithm
clear all
clc;
tic;
% ==Economic dispatch problem including losses and generator limits
% ========Initial Lagrange's variable================================
lambda=40               % Initial lambda
```

```
%===========Fuel cost coefficients==================================
a=[0.005    0.0055  0.006   0.005   0.005   0.007   0.007   0.007   0.005
0.005   0.0055  0.0045  0.007   0.006];
b=[1.89 2   3.5 3.15    3.05    2.75    3.45    3.45    2.45    2.45
2.35    1.6 3.45    3.89];
c=[150  115 40  122 125 120 70  70  130 130 135 200 70  45];
% ======================== Emission Coefficients====================
d=[0.016    0.031   0.013   0.012   0.02    0.007   0.015   0.018   0.019
0.012   0.033   0.018   0.018   0.03];
e=[-1.5 -1.82   -1.249  -1.355  -1.9    0.805   -1.401  -1.8    -2  -1.36
-2.1    -1.8    -1.81   -1.921];
f=[23.333   21.022  22.05   22.983  21.313  21.9    23.001  24.003
25.121  22.99   27.01   25.101  24.313  27.119];
% ==================Generator Limits ===============================
Pmin=[50    50  50  50  50  50  50  50  50  50  50  50  50  50];
Pmax=[1000  1000    1000    1000    1000    1000    1000    1000    1000
1000    1000    1000    1000    1000];
% ==============Transmission loss coefficients=======================
B00=0.028738;
B01=[-0.5385    -0.28323    -0.19294    -0.26424    0.01776 0.02192
0.04051     0.01222     0.01401     0.00441     0.03273     0.21782
0.03256     0.15563];
B=10^-1*
[0.04274    0.03011     0.01924     0.02151     -0.00288    -0.004
-0.00447    -0.00272    -0.00323    -0.00694    -0.00745    -0.01952
-0.01217    -0.01718
0.0301      0.03795     0.02071     0.02091     -0.00363    -0.00525
-0.00448    -0.00366    -0.00359    -0.00695    -0.01018    -0.02004
-0.01844    -0.02057
0.01924     0.02071     0.02678     0.0247      -0.00247    -0.00378
-0.00298    -0.00239    -0.00231    -0.00467    -0.00786    -0.01583
-0.01529    -0.01668
0.02151     0.02091     0.0247      0.02439     -0.00232    -0.00352
-0.00309    -0.00223    -0.0023     -0.00475    -0.00715    -0.016
-0.01346    -0.01588
-0.00288    -0.00363    -0.00247    -0.00232    0.00954     0.00366
0.00295     0.00312     0.00421     0.00207     0.00037     -0.00365
-0.00381    -0.00424
-0.004      -0.00525    -0.00378    -0.00352    0.00366     0.01068
0.00576     0.00374     0.00334     0.00249     0.00119     -0.00279
-0.00288    -0.00331
-0.00447    -0.00448    -0.00298    -0.00309    0.00295     0.00576
0.00809     0.00337     0.00357     0.00305     0.00129     -0.00252
-0.00192    -0.00272
-0.00272    -0.00366    -0.00239    -0.00223    0.00374     0.00374
0.00337     0.00388     0.00375     0.00293     0.00206     -0.00152
-0.00142    -0.00188
-0.00323    -0.00359    -0.00231    -0.0023     0.00334     0.00334
0.00357     0.00375     0.0054      0.00287     0.00148     -0.00225
-0.00189    -0.00254
-0.00694    -0.00695    -0.00467    -0.00475    0.00249     0.00249
0.00305     0.00293     0.00287     0.00674     0.00305     0.00121
0.00133     0.00096
-0.00745    -0.01018    -0.00786    -0.00715    0.00119     0.00119
0.00125     0.00206     0.00148     0.00305     0.00858     0.00617
0.00818     0.00726
```

```
-0.00195    -0.02004    -0.01583    -0.016      -0.00279    -0.00279
-0.00252    -0.00152    -0.00225    0.00121     0.00617     0.03615
0.01839     0.02002
-0.01217    -0.01844    -0.01529    -0.1346     -0.00288    -0.00288
-0.00192    -0.00142    -0.00189    0.00133     0.00818     0.01839
0.03312     0.02941
-0.01718    -0.02057    -0.01668    -0.01588    -0.00331    -0.00331
-0.00272    -0.00188    -0.00254    0.00096     0.00726     0.02002
0.02941     0.0413];
%matlabpool open jobmanagerconfig1 %Opening 6 matlab workers from the
cluster
t0 = clock; % start of the clock on second level calculation
jm = findResource('scheduler','configuration','jobmanagerconfig1')
disp(get(jm))%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% check the status of the
jobmanager
timingStart = tic;
start = tic;
pjob1=createParallelJob(jm,'Configuration','jobmanagerconfig1'); %create
parallel job from the jobmanager
get(pjob1) % Monitor the status of pjob1
%==========How long it takes to create a job============================
times.jobCreateTime = toc(start)
description.jobCreateTime = 'Job creation time'
%======================================================================
Tasks = findTask(pjob1)
[pending running finished] = findTask(pjob1)
set(pjob1,'Configuration','jobmanagerconfig1')
set(pjob1,'MinimumNumberOfWorkers',14);
set(pjob1,'MaximumNumberOfWorkers',14);
set(pjob1,'FileDependencies',{'CEED_Casestudy2_DataParallel_funct.m'})
get(jm)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Notice how jobmanager is
changing
disp(get(pjob1))
disp('busy workers')
disp(get(jm, 'NumberOfbusyWorkers'))
%====================STARTING PARALLEL COMPUTING========================
%Create task- only one with 21 output arguments  and 12 input arguments,
%measure how long that takes and measure how long it takes to submit the
%job to the cluster
task1=createTask(pjob1,
@CEED_Casestudy2_DataParallel_funct,22,{Pmax,Pmin,f,e,d,c,b,a,lambda,B00,
B01,B})
get(pjob1,'task')
%==========How long it takes to create a task===========================
times.taskCreateTime = toc(start)
description.taskCreateTime = 'Task creation time'
%======================================================================
submit(pjob1)
%===========How long it takes to submit a job===========================
times.submitTime = toc(start)
description.submitTime = 'Job submission time'
%======================================================================
%Once the job has been submitted, we hope all its tasks execute in
%parallel. Measure how long it takes for all the tasks to start and to
run
%to completion.
```

```
waitForState(pjob1,'finished')
times.jobWaitTime = toc(start)
description.pjobWaitTime = 'Job wait time'
get(jm)
%Tasks have now completed, so we are again executing a code in the matlab
%client. Measure how long it takes to retrieve all the job results
disp('PD lambda P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 PL FC ET
FT iter') % display the output values
results=getAllOutputArguments(pjob1)
times.resultsTime = toc(start)
description.resultsTime = 'Results retrieval time'
%======================================================================
%============Verify that the job ran without errors====================
errmsgs = get(pjob1.task, {'ErrorMessage'})
nonempty = ~cellfun(@isempty, errmsgs)
celldisp(errmsgs(nonempty))
%=====================================================================

%Measure the total time elapsed from creating the job up to the results
times.totalTime = toc(start)
operationtime = etime(clock,t0) %operation time for completion of
parallel computing
destroy(pjob1)
%matlabpool close% close Matlabpool of workers
```

Appendix E4: MATLAB script file – CEED_Casestudy2_DataParallel_funct.m

```
% ================CEED_Casestudy2_DataParallel_function file=============
function [ PD lambda P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 PL FC
ET FT iter time] =
CEED_Casestudy2_Parallel_funct(Pmax,Pmin,f,e,d,c,b,a,lambda,B00,B01,B)
tic;
% The Power demand varies according to lab index
if labindex ==1
PD=3668;
end
if  labindex==2
    PD=3680;
end
if  labindex==3
    PD=3700;
end
if  labindex==4
    PD=3720;
end
if  labindex==5
    PD=3740;
end
if labindex==6
    PD=3760;
end
if labindex==7
    PD=3780;
end
if labindex==8
    PD=3800;
end
```

```
if labindex==9
    PD=3820;
end
if labindex==10
    PD=3840;
end
if labindex==11
    PD=3860;
end
if labindex==12
    PD=3880;
end
if labindex==13
    PD=3900;
end
if labindex==14
    PD=3920;
end
epsilon=0.001;           % Tolerance value
alfa=0.001;              % Incerememtal deltalambda
n=14;                     % No of generators
m=5000;                  % Total number of iterations
for iter=1:m
    %=================Lagrange's algorithm starts here ====================
    iter        % To Print the each iteration number
    % ===========Calculation of Price penalty factor=================
    %h=[(a.*Pmin.^2+b.*Pmin+c)]./[(d.*Pmax.^2+e.*Pmax+f)];   % Equation
(3.7 or 3.8 or 3.9 or 3.10)
    h=[(a.*Pmax.^2+b.*Pmax+c)]./[(d.*Pmax.^2+e.*Pmax+f)];   % Equation
(3.7 or 3.8 or 3.9 or 3.10)
    % =================Calculation of generator real power ==========
    E=[(diag(a+(h.*d))./lambda)+B];         % Equation (3.19)
    D=0.5*((1-(b+h.*e)./lambda-B01))';      % Equation (3.19)
    P=E\D;                          % Equation (3.21)
    %======================Generator limit checking =================
    for i=1:n
        if P(i)<=Pmin(i)
            P(i)=Pmin(i)
        elseif P(i)>=Pmax(i)
            P(i)=Pmax(i)
        else P(i)=P(i)
        end
    end
% The real power output of individual generators are assigned to the
output of the function file
P1=P(1); P2=P(2); P3=P(3); P4=P(4); P5=P(5); P6=P(6); P7=P(7);
P8=P(8); P9=P(9); P10=P(10); P11=P(11); P12=P(12); P13=P(13); P14=P(14);
 %========== Calculation of Transmission loss=====================
  PL1=B00;
    PL2=zeros(1,n);
    for i=1:n
        PL2(i)=(B01(i)*P(i));       % Equation (3.3)
    end
    PL2=sum(PL2(:,:));
    PL3=zeros(n,n);
    for i=1:n
        for j=1:n
            PL3(i,j)=P(i)*B(i,j)*P(j);  % Equation (3.3)
```

```
        end
    end
    PL3=sum(PL3(:,:));
    PL3=sum(PL3(:,:));
    PL=-(PL1+PL2+PL3);
    % ======================Calculation of deltalambda ==============
    deltalambda=0.0;
    deltalambda=PD+PL;
    for i=1:n
        deltalambda=deltalambda-P(i);       % Equation (3.22)
    end
    if abs(deltalambda)<=epsilon | iter>=m
        break
    else lambda=lambda+deltalambda*alfa  % Equation (3.23)
    end
end
%====== Calculation of fuel cost, emission and CEED values ==========
fuelcost=0;
emissioncost=0;
CEED=0;
for i=1:n
    fuelcost=fuelcost+a(i)*P(i)^2+b(i)*P(i)+c(i); % Equation (3.1)
    emissioncost=emissioncost+d(i)*P(i)^2+e(i)*P(i)+f(i); % Equation
(3.5)
    CEED=
CEED+a(i)*P(i)^2+b(i)*P(i)+c(i)+h(i)*(d(i)*P(i)^2+e(i)*P(i)+f(i)); %
Equation (3.6)
end
%==== Printing of Real power, Transmission loss, Fuel cost, Emission and
CEED===
lambda
P
PL
FC=sum(fuelcost(:,:))
ET=sum(emissioncost(:,:))
FT=sum(CEED(:,:))
time=toc;
end
%======================================================================
```

APPENDIX F: MATLAB PROGRAM FOR DATA PARALLEL CALCULATION OF THE CEED PROBLEM SOLUTION USING PSO ALGORITHM

Appendix F1: MATLAB script file – PSO_10swarms_CEED_DataParallel.m

```
% M-File : PSO_10swarms_CEED_DataParallel.m
%IEEE 30 BUS SIX GENERATOR SYSTEM
% ==================M-File Description================================
% The M-File is used to solve CEED problem with various price penalty
% factors using PSO Algorithm in a Parallel way
%=================PSO algorithm starts here ===========================
clear all %Removes all variables, global, functions and MEX links
clc % Clear the command window
tic % Start a stopwatch timer.
%% ======================Economic Dispatch Parameters======================
%==========Load Demand and Generator limits=================================
%PD=250; % Load demand
Pmin1=[50 20 15 10 10 12];% Minimum generator limits including slack bus
Pmax1=[200 80 50 35 30 40];% Maximum generator limits including slack bus
Pdmin=[50];% Minimum  generator limit of Slack bus
Pdmax=[200];% Maximum  generator limit of Slack bus
Pmin=[20 15 10 10 12];% Minimum generator limits excluding slack bus
Pmax=[80 50 35 30 40]; % Maximum generator limits excluding slack bus
% ===================Cost and Emission coefficients=========================
a=[0.00375 0.01750 0.06250 0.00834 0.02500 0.02500];
b=[2.00 1.75 1.00 3.25 3.00 3.00];
c=[0 0 0 0 0 0];
d=[0.0126 0.0200 0.0270 0.0291 0.0290 0.0271];
e=[-1.1000 -0.1000 -0.1000 -0.0050 -0.0400 -0.0055];
ff=[22.983 22.313 25.505 24.900 24.700 25.300];
n=6;   % Number of generators
%====================Transmission loss B - coefficients====================
B=[0.000218 0.000103 0.000009 -0.000010 0.000002 0.000027;
   0.000103 0.000181 0.000004 -0.000015 0.000002 0.000030;
   0.000009 0.000004 0.000417 -0.000131 -0.000153 -0.000107;
   0.000010 -0.000015 -0.000131 0.000221 0.000094 0.000050;
   0.000002 0.000002 -0.000153 0.000094 0.000243 -0.000000;
   0.000027 0.000030 -0.000107 0.000050 -0.000000 0.000358];
%% PSO Parameters
wmin=0.4; % Minimum weight
wmax=0.9; % Maximum weight
w=0.65;   % weight
c1=2.0;   % acceleration factor1
c2=2.0;   % acceleration factor2
Min=0;    % Minimum Value
Max=1;    % Maximum value
m=25;     % Number of iterations
% ==============Matlab Parallel PSO ED code=================================
%Matlabpool open jobmanagerconfig1 %Opening 6 Matlab workers from the
cluster
t0 = clock; % start of the clock on second level calculation
jm = findResource('scheduler','configuration','jobmanagerconfig1')
disp(get(jm))%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% check the status of the
jobmanager
timingStart = tic;
start = tic;
```

```
pjob1=createParallelJob(jm,'Configuration','jobmanagerconfig1'); %create
parallel job from the jobmanager
get(pjob1) % Monitor the status of pjob1

%==========How long it takes to create a job============================
times.jobCreateTime = toc(start)
description.jobCreateTime = 'Job creation time'
%======================================================================
Tasks = findTask(pjob1)
[pending running finished] = findTask(pjob1)
set(pjob1,'Configuration','jobmanagerconfig1')
set(pjob1,'MinimumNumberOfWorkers',6);
set(pjob1,'MaximumNumberOfWorkers',6);
set(pjob1,'FileDependencies',{'PSO_10swarms_CEED_DataParallel_funct.m'})
get(jm)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Notice how jobmanager is
changing
disp(get(pjob1))
disp('busy workers')
disp(get(jm, 'NumberOfbusyWorkers'))
%====================STARTING PARALLEL COMPUTING=======================
%Create task- only one with 4 output arguments  and not input arguments,
%measure how long that takes and measure how long it takes to submit the
%job to the cluster
task1=createTask(pjob1,
@PSO_10swarms_CEED_DataParallel_funct,12,{Pdmin,Pdmax,Pmin,B,a,b,c,d,e,ff,P
min1,Pmax,Pmax1,wmin,wmax,w,c1,c2,Min,Max})
get(pjob1,'task')
%==========How long it takes to create a task===========================
times.taskCreateTime = toc(start)
description.taskCreateTime = 'Task creation time'
%======================================================================
submit(pjob1)
%===========How long it takes to submit a job===========================
times.submitTime = toc(start)
description.submitTime = 'Job submission time'
%======================================================================

%Once the job has been submitted, we hope all its tasks execute in
%parallel. Measure how long it takes for all the tasks to start and to run
%to completion.
waitForState(pjob1,'finished')
times.jobWaitTime = toc(start)
description.pjobWaitTime = 'Job wait time'
get(jm)
%Tasks have now completed, so we are again executing a code in the matlab
%client. Measure how long it takes to retrieve all the job results
disp('P1  P2  P3  P4  P5  P6  PL  FC  ET FT iter time') % display the
output values
results=getAllOutputArguments(pjob1)
times.resultsTime = toc(start)
description.resultsTime = 'Results retrieval time'
%======================================================================
%============Verify that the job ran without errors====================
errmsgs = get(pjob1.task, {'ErrorMessage'})
nonempty = ~cellfun(@isempty, errmsgs)
celldisp(errmsgs(nonempty))
%=====================================================================
%Measure the total time elapsed from creating the job up to the results
```

```
times.totalTime = toc(start)
operationtime = etime(clock,t0) %operation time for completion of parallel
computing
destroy(pjob1)
%Matlabpool close% close Matlabpool of workers
```

Appendix F2: MATLAB script file – PSO_10swarms_CEED_DataParallel_funct.m

```
% M-File : PSO_10swarms_CEED_DataParallel_funct.m
%IEEE 30 BUS SIX GENERATOR SYSTEM
% ==================M-File Description===============================
% The M-File is used to solve CEED problem with various price penalty
% factors using PSO Algorithm in a Parallel way
%================PSO algorithm starts here ===========================
function [P1 P2 P3 P4 P5 P6 PL FC  ET FT iter time] =
PSO_10swarms_CEED_DataParallel_funct(Pdmin,Pdmax,Pmin,B,a,b,c,d,e,ff,Pmin1,
Pmax,Pmax1,wmin,wmax,w,c1,c2,Min,Max)
tic
if labindex ==1
 PD=125;
end
if  labindex==2
    PD=150;
end
if  labindex==3
    PD=175;
end
if  labindex==4
    PD=200;
end
if  labindex==5
    PD=225;
end
if labindex==6
    PD=250;
end
rng('shuffle')%RNG('shuffle') seeds the random number generator based on
the current time so that RAND, RANDI, and RANDN produce a different
sequence of numbers after each time you call RNG.
r1 = Min + (Max-Min).*rand(1,1); %random number1 between ( 0 to 1)
r2= Min + (Max-Min).*rand(1,1);  %random number2 between ( 0 to 1)
m=25;     % Number of iterations
%% ================= PSO Algorithm for Economic Dispatch Problem===========
for iter=1:m% Number of iterations
%% =============PSO Velocity Calculation - 1st Level of solution===========
for i=1:5   % Number of velocities depends on(No of generators of the
plant)
vmin(i)=[-0.5*Pmin(i)] % calculation of minimum velocity
vmax(i)=[0.5*Pmax(i)]  % calculation of maximum velocity
    for j=1:11         % Number of particles position (varying quantity)
    v(j,i)=[vmin(i)+rand()*(vmax(i)-vmin(i))] % calculation of velocity
    end
end
%% ================= PSO Position Calculation ==============================
for j=1:11 % Number of Positions
```

```
    for i=1:5  % Number of velocities
     rng('shuffle')
    P(i)=Pmin(i)+rand()*(Pmax(i)-Pmin(i));%GEN Real power- except slack bus
    end
%% =================Genertor MIN & MAX limits===============================
    for i=1:5
        if P(i)<=Pmin(i)
        P(i)=Pmin(i);
        elseif P(i)>=Pmax(i)
        P(i)=Pmax(i);
        else P(i) = P(i);
        end
    end
%% ===============Dependent Generator =====================================
    P=[zeros P]; % Pre-allocation of P
    x=B(1,1);    % calculation x
    y=zeros(5,1);% Pre-allocation of y
    for k=2:6
      for r=1:1
          y(k,r)=((B(k,r)+B(r,k))*P(k)); % calculation of y
      end
    end
    y=sum(y)-1;
    PL=zeros(5,5); % Pre-allocation of PL
    for k=2:6
        for l=2:6
        PL(k,l)= P(k)*B(k,l)*P(l); % calculation of PL
        end
    end
    PL1=sum(PL);
    PL2=sum(PL1);
    PL=PL2;
    p=sum(P(2:6));
    z=PD+PL-p; % calculation of z
    Pd=-(y+(y.^2- 4.*x.*z).^(1/2))/(2.*x);% calculation of dependent GEN
    demo1=-(y-(y.^2- 4.*x.*z).^(1/2))/(2.*x);% calculation of dependent GEN
    P=P(2:6);
%% ====================Dependent Generator Limit ==========================
    if Pd<=Pdmin % checking the slack bus generator limit
       Pd=Pdmin;
    elseif Pd>=Pdmax
           Pd=Pdmax;
    else Pd=Pd;
    end
    p=[Pd P]; % generator power including slack bus
%% ================= Transmission loss ====================================
    for k=1:6
        for l=1:6
            PL(k,l)=p(k)*B(k,l)*p(l);
        end
    end
    Pl=sum(PL);
    PL=sum(Pl);
    eval(['PL' num2str(j) ' = (PL)'])
    eval(['p' num2str(j) ' = (p)'])
%% ============Price Penalty Factor (PPF)===================================
%     h=[(a.*Pmax1.^2+b.*Pmax1+c)]./[(d.*Pmax1.^2+e.*Pmax1+ff)];%PPF
      h=[(a.*Pmin1.^2+b.*Pmin1+c)]./[(d.*Pmax1.^2+e.*Pmax1+ff)];%PPF
```

```
%% ========== CEED Fuel cost, Fuel cost and  Emission ====================
    for i=1:6
    f(i)=a(i)*p(i)^2+b(i)*p(i)+c(i)+(h(i)*(d(i)*p(i)^2+e(i)*p(i)+ff(i))); %
CEED Fuel cost of individual GEN
    fuel(i)=a(i)*p(i)^2+b(i)*p(i)+c(i);% Fuel cost (FC)of individual GEN
    Emission(i)=d(i)*p(i)^2+e(i)*p(i)+ff(i);% Emission value of individual
GEN
    end
    F=sum(f);               % CEED Fuel cost
    fuel=sum(fuel);         % Fuel cost
    Emission=sum(Emission); % Emission
    q=[F p PL fuel Emission ]; %Economic dispatch(PSO)1st level of Solution
    eval(['q' num2str(j) ' = (q)']);
    eval(['z' num2str(j) '=(q(:,1))']);
end
%% Sort the Fuel cost in ascending order
p=[p1; p2; p3; p4 ;p5; p6;p7;p8;p9;p10;p11];%Print the real power P
p=p(:,(2:6)) % print the real power generator values
z=[z1 z2 z3 z4 z5 z6 z7 z8 z9 z10 z11]; %Total fuel cost value
q=[q1; q2; q3 ;q4; q5; q6; q7; q8; q9; q10;q11;]%print the q=[FT, P1....Pn,
FC,ET]
[aa bb]=min(z) % To find the minimum of FT
gbest1=q(bb,:) % To select the Gbest as best of all positions
gbest=q(bb,(3:7)) % To select the global best positions
%% % Economic dispatch (PSO) 2nd level of Solution
%% ===================New Velocity=======================================
for j=1:11
    for i=1:5
    w(i)= wmax-((i*(wmax-wmin))/5); % Calculate the inertia weight
    vnew=w(i)*v(j,i)+(c1*r1*(p(:,i)-p(:,i)))+(c2*r2*(gbest(:,i)-
p(:,i)));%New velocity
    eval(['vnew' num2str(i) ' = (vnew)']);
    end
end
vnew=[vnew1 vnew2 vnew3 vnew4 vnew5];
 %% =============check the limits of new velocities========================
for i=1:11
    for j=1:5
        if vnew(i,j)<=vmin(j)
        vnew(i,j)=vmin(j)
        elseif vnew(i,j)>=vmax(j)
        vnew(i,j)=vmax(j)
        end
    end
end
%% =============== New position==========================================
pneww=zeros(11,5) % Pre-allocation of new position (i.e.,pneww)
for i=1:11
    for j=1:5
    pneww(i,j)=p(i,j)+vnew(i,j); % New position
    end
eval(['pneww' num2str(i) ' = (pneww)'])
end
%% ============Check the limits of new position============================
for i=1:11
    for j=1:5
        if pneww(i,j)<=Pmin(j)
           pneww(i,j)=Pmin(j);
```

```
        elseif pneww(i,j)>=Pmax(j)
               pneww(i,j)=Pmax(j);
        end
    end
end
pnew=[pneww11];
%% ===============New dependent Generator ==================================
for i=1:11
    pneww=pnew(i,:)
    pneww=[zeros pneww];
    x=B(1,1); % calculaition of x
    y=zeros(5,1); %Pre-allocation of y
    for k=2:6
        for r=1:1
        y(k,r)=((B(k,r)+B(r,k))*pneww(k)); % Calculation of y
        end
    end
    y=sum(y)-1;
    PL=zeros(5,5); % Pre - allocation of PL
    for k=2:6
        for l=2:6
        PL(k,l)=pneww(k)*B(k,l)*pneww(l); % Calculation of PL
        end
    end
    PL11=sum(PL);
    PL22=sum(PL11);
    PL=PL22;
    p=sum(pneww(2:6));
    z=PD+PL-p;
    pdnew=-(y+(y.^2- 4.*x.*z).^(1/2))/(2*x); %New dependent Generator1
    demo1=-(y-(y.^2- 4.*x.*z).^(1/2))/(2*x); %New dependent Generator2
%% =====================New Dependent Generator Limit =====================
    if pdnew<=Pdmin % checking the slack bus generator limit
        pdnew=Pdmin;
    elseif pdnew>=Pdmax
        pdnew=Pdmax;
    else pdnew=pdnew;
    end
%% =================== New Transmission loss ==============================
    pneww=pneww(2:6) % New GEN real power excluding slack bus
    pneww=[pdnew pneww] % New GEN real power including slack bus
    for k=1:6
        for l=1:6
        PL(k,l)=pneww(k)*B(k,l)*pneww(l); % New Transmission loss
        end
    end
    Pl=sum(PL);
    PL=sum(Pl);
    eval(['PL' num2str(i) ' = (PL)']);
    eval(['pneww' num2str(i) ' = (pneww)']);
    eval(['pdnew' num2str(i) ' = (pdnew)']);
end
pnew=[pneww1; pneww2; pneww3; pneww4; pneww5; pneww6; pneww7; pneww8;
pneww9 ;pneww10;pneww11];
PL=[PL1 PL2 PL3 PL4 PL5 PL6 PL7 PL8 PL9 PL10 PL11]';
%% ====Calculate the New CEED Fuel cost,New Fuel cost and New Emission=====
for j=1:11
    for i=1:6
```

```
Fnew(i)=a(i)*pnew(j,i)^2+b(i)*pnew(j,i)+c(i)+(h(i)*(d(i)*pnew(j,i)^2+e(i)*p
new(j,i)+ff(i))); % New CEED Fuel cost FTnew
    fuelnew(i)=a(i)*pnew(j,i)^2+b(i)*pnew(j,i)+c(i);% New Fuel cost (FCnew)
    Emissionnew(i)=d(i)*pnew(j,i)^2+e(i)*pnew(j,i)+ff(i);%NewEmission ETnew
    end
    Fnew=sum(Fnew(:,:));
    fuelnew=sum(fuelnew(:,:));
    Emissionnew=sum(Emissionnew(:,:));
    eval(['Fnew' num2str(j) ' = (Fnew)']);
    eval(['fuelnew' num2str(j) ' = (fuelnew)']);
    eval(['Emissionnew' num2str(j) ' = (Emissionnew)']);
end
%% =====PSO 2nd level solution; qnew=[FTnew, P1new....Pnnew, FCnew,ETnew]==
Fnew=[Fnew1;Fnew2;Fnew3;Fnew4;Fnew5;Fnew6;Fnew7;Fnew8;Fnew9;Fnew10;Fnew11];
fuelnew=[fuelnew1;fuelnew2;fuelnew3;fuelnew4;fuelnew5;fuelnew6;fuelnew7;fue
lnew8;fuelnew9;fuelnew10;fuelnew11];
Emissionnew=[Emissionnew1;Emissionnew2;Emissionnew3;Emissionnew4;Emissionne
w5;Emissionnew6;Emissionnew7;Emissionnew8;Emissionnew9;Emissionnew10;Emissi
onnew11];
qnew=[Fnew pnew PL fuelnew Emissionnew]
%% ====Compare the PSO solution of 1st level with 2nd level (q and qnew)===
x=q(:,1); % Assisgn q to x
y=qnew(:,1); % Assisgn qnew to y
for i=1:11
    if x(i)<y(i); % compare q and  qnew
    z(i)=y(i);
    qrow=i;
    x1=(q(qrow,:));
    mat(i,:)=x1  % Store best solution in mat
    end
    if x(i)>y(i);% compare q and  qnew
    z(i)=x(i);
    qnewrow=i;
    y1=qnew(qnewrow,:);
    mat(i,:)=y1  % Store best solution in mat
    end
end
%% ==================Store the best PSO solution ==========================
eval(['mat' num2str(iter) ' = (mat)']);
eval(['matz' num2str(iter) '=(mat(:,1))']);
end % PSO Iteration Ends
%% ==================PSO solution =========================================
final=[mat1; mat2 ;mat3 ;mat4; mat5; mat6; mat7; mat8 ;mat9; mat10 ;mat11
;mat12; mat13; mat14 ;mat15 ;mat16; mat17; mat18; mat19;
mat20;mat21;mat22;mat23;mat24;mat25]% PSO Solution of all the iterations
[row col]=min(final(:,1))% Find the best solution of PSO from the solution
of each iteration
finalans=final(col,:) % print the best solution of PSO
iteration=fix(col/11)+1 % print the iteration Number of the best solution
%% Graph: PSO solution to Economic Dispatch Problem
y_axis=final(:,1);
graph=plot(y_axis, '*');
hold on
plot(finalans(1,1),'o')
set(graph,'Color','red','LineWidth',2)
ANS=[finalans(:,:) iteration toc]
```

```
P1=ANS(2);P2=ANS(3);P3=ANS(4);P4=ANS(5);P5=ANS(6);P6=ANS(7);PL=ANS(8);FC=AN
S(9);ET=ANS(10);FT=ANS(1);iter=ANS(11);time=ANS(12);
end
%% The PSO Program for Economic Dispatch ----------Ends
```

Appendix F3: **MATLAB script file – PSO_30swarms_CEED_DataParallel.m**

```
% M-File : PSO_30swarms_CEED_DataParallel.m
%IEEE 30 BUS SIX GENERATOR SYSTEM
% ==================M-File Description================================
% The M-File is used to solve CEED problem with various price penalty
% factors using PSO Algorithm in a Parallel way
%================PSO algorithm starts here ==========================

clear all %Removes all variables, global, functions and MEX links
clc % Clear the command window
tic % Start a stopwatch timer.
%% ======================Economic Dispatch Parameters======================
%==========Load Demand and Generator limits==================================
%PD=250; % Load demand
Pmin1=[50 20 15 10 10 12];% Minimum generator limits including slack bus
Pmax1=[200 80 50 35 30 40];% Maximum generator limits including slack bus
Pdmin=[50];% Minimum  generator limit of Slack bus
Pdmax=[200];% Maximum  generator limit of Slack bus
Pmin=[20 15 10 10 12];% Minimum generator limits excluding slack bus
Pmax=[80 50 35 30 40]; % Maximum generator limits excluding slack bus
% ===================Cost and Emission coefficients=========================
a=[0.00375 0.01750 0.06250 0.00834 0.02500 0.02500];
b=[2.00 1.75 1.00 3.25 3.00 3.00];
c=[0 0 0 0 0 0];
d=[0.0126 0.0200 0.0270 0.0291 0.0290 0.0271];
e=[-1.1000 -0.1000 -0.1000 -0.0050 -0.0400 -0.0055];
ff=[22.983 22.313 25.505 24.900 24.700 25.300];
n=6;    % Number of generators
%====================Transmission loss B - coefficients===================
B=[0.000218 0.000103 0.000009 -0.000010 0.000002 0.000027;
   0.000103 0.000181 0.000004 -0.000015 0.000002 0.000030;
   0.000009 0.000004 0.000417 -0.000131 -0.000153 -0.000107;
   0.000010 -0.000015 -0.000131 0.000221 0.000094 0.000050;
   0.000002 0.000002 -0.000153 0.000094 0.000243 -0.000000;
   0.000027 0.000030 -0.000107 0.000050 -0.000000 0.000358];
%% PSO Parameters
wmin=0.4; % Minimum weight
wmax=0.9; % Maximum weight
w=0.65;   % weight
c1=2.0;   % acceleration factor1
c2=2.0;   % acceleration factor2
Min=0;    % Minimum Value
Max=1;    % Maximum value
m=40;     % Number of iterations
% =============Matlab Parallel PSO ED code================================
%Matlabpool open jobmanagerconfig1 %Opening 6 Matlab workers from the
cluster
t0 = clock; % start of the clock on second level calculation
jm = findResource('scheduler','configuration','jobmanagerconfig1')
disp(get(jm))%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% check the status of the
jobmanager
```

```
timingStart = tic;
start = tic;
pjob1=createParallelJob(jm,'Configuration','jobmanagerconfig1'); %create
parallel job from the jobmanager
get(pjob1) % Monitor the status of pjob1

%==========How long it takes to create a job============================
times.jobCreateTime = toc(start)
description.jobCreateTime = 'Job creation time'
%======================================================================
Tasks = findTask(pjob1)
[pending running finished] = findTask(pjob1)
set(pjob1,'Configuration','jobmanagerconfig1')
set(pjob1,'MinimumNumberOfWorkers',6);
set(pjob1,'MaximumNumberOfWorkers',6);
set(pjob1,'FileDependencies',{'PSO_30swarms_CEED_DataParallel_funct.m'})
get(jm)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Notice how jobmanager is
changing
disp(get(pjob1))
disp('busy workers')
disp(get(jm, 'NumberOfbusyWorkers'))
%====================STARTING PARALLEL COMPUTING========================
%Create task- only one with 4 output arguments  and not input arguments,
%measure how long that takes and measure how long it takes to submit the
%job to the cluster
task1=createTask(pjob1,
@PSO_30swarms_CEED_DataParallel_funct,12,{Pdmin,Pdmax,Pmin,B,a,b,c,d,e,ff,P
min1,Pmax,Pmax1,n,m,wmin,wmax,w,c1,c2,Min,Max})
get(pjob1,'task')
%==========How long it takes to create a task===========================
times.taskCreateTime = toc(start)
description.taskCreateTime = 'Task creation time'
%======================================================================
submit(pjob1)
%===========How long it takes to submit a job===========================
times.submitTime = toc(start)
description.submitTime = 'Job submission time'
%======================================================================

%Once the job has been submitted, we hope all its tasks execute in
%parallel. Measure how long it takes for all the tasks to start and to run
%to completion.
waitForState(pjob1,'finished')
times.jobWaitTime = toc(start)
description.pjobWaitTime = 'Job wait time'
get(jm)
%Tasks have now completed, so we are again executing a code in the matlab
%client. Measure how long it takes to retrieve all the job results
disp('FT P1  P2  P3  P4  P5  P6  PL  FC  ET') % display the output values
results=getAllOutputArguments(pjob1)
times.resultsTime = toc(start)
description.resultsTime = 'Results retrieval time'
%======================================================================
%============Verify that the job ran without errors=====================
errmsgs = get(pjob1.task, {'ErrorMessage'})
nonempty = ~cellfun(@isempty, errmsgs)
celldisp(errmsgs(nonempty))
%=====================================================================
```

```
%Measure the total time elapsed from creating the job up to the results
times.totalTime = toc(start)
operationtime = etime(clock,t0) %operation time for completion of parallel
computing
destroy(pjob1)
%Matlabpool close% close Matlabpool of workers
toc; % Stop a Stop watch timer
```

==

Appendix F4: **MATLAB script file – PSO_30swarms_CEED_DataParallel_funct.m**

```
% M-File : PSO_30swarms_CEED_Parallel_funct.m
%IEEE 30 BUS SIX GENERATOR SYSTEM
% ==================M-File Description================================
% The M-File is used to solve CEED problem with various price penalty
% factors using PSO Algorithm in a Parallel way
%=================PSO algorithm starts here ===========================
function [P1 P2 P3 P4 P5 P6 PL FC  ET FT iter time] =
PSO_30swarms_CEED_DataParallel_funct(Pdmin,Pdmax,Pmin,B,a,b,c,d,e,ff,Pmin1,
Pmax,Pmax1,n,m,wmin,wmax,w,c1,c2,Min,Max)
tic
if labindex ==1
PD=125;
end
if  labindex==2
    PD=150;
end
if  labindex==3
    PD=175;
end
if  labindex==4
    PD=200;
end
if  labindex==5
    PD=225;
end
if labindex==6
    PD=250;
end
rng('shuffle')
r1 = Min + (Max-Min).*rand(1,1); %random number1 between ( 0 to 1)
r2= Min + (Max-Min).*rand(1,1);  %random number2 between ( 0 to 1)
%% ================= PSO Algorithm for Economic Dispatch Problem===========
for iter=1:m% Number of iterations
%% =============PSO Velocity Calculation - 1st Level of solution===========
for i=1:5   % Number of velocities depends on(No of generators of the
system)
vmin(i)=[-0.5*Pmin(i)] % calculation of minimum velocity
vmax(i)=[0.5*Pmax(i)]  % calculation of maximum velocity
    for j=1:30         % Number of particles position (varying quantity)
    v(j,i)=[vmin(i)+rand()*(vmax(i)-vmin(i))] % calculation of velocity
    end
end
%% ================= PSO Position Calculation =============================
for j=1:30 % Number of Positions
    for i=1:5  % Number of velcoities
     rng('shuffle')
    P(i)=Pmin(i)+rand()*(Pmax(i)-Pmin(i));%GEN Real power- except slack bus
```

```
    end
%% =================Generator MIN & MAX limits=============================
    for i=1:5
        if P(i)<=Pmin(i)
        P(i)=Pmin(i);
        elseif P(i)>=Pmax(i)
        P(i)=Pmax(i);
        else P(i) = P(i);
        end
    end
%% ===============Dependent Generator =====================================
    P=[zeros P]; % Pre-allocation of P
    x=B(1,1);    % calculation x
    y=zeros(5,1);% Pre-allocation of y
    for k=2:6
      for r=1:1
          y(k,r)=((B(k,r)+B(r,k))*P(k)); % calculation of y
      end
    end
    y=sum(y)-1;
    PL=zeros(5,5); % Pre-allocation of PL
    for k=2:6
        for l=2:6
        PL(k,l)= P(k)*B(k,l)*P(l); % calculation of PL
        end
    end
    PL1=sum(PL);
    PL2=sum(PL1);
    PL=PL2;
    p=sum(P(2:6));
    z=PD+PL-p; % calculation of z
    Pd=-(y+(y.^2- 4.*x.*z).^(1/2))/(2.*x);% calculation of dependent GEN
    demo1=-(y-(y.^2- 4.*x.*z).^(1/2))/(2.*x);% calculation of dependent GEN
    P=P(2:6);
%% =====================Dependent Generator Limit =========================
    if Pd<=Pdmin % checking the slack bus generator limit
       Pd=Pdmin;
    elseif Pd>=Pdmax
           Pd=Pdmax;
    else Pd=Pd;
    end
    p=[Pd P]; % generator power including slack bus
%% ================= Transmission loss ===================================
    for k=1:6
        for l=1:6
            PL(k,l)=p(k)*B(k,l)*p(l);
        end
    end
    Pl=sum(PL);
    PL=sum(Pl);
    eval(['PL' num2str(j) ' = (PL)'])
    eval(['p' num2str(j) ' = (p)'])
%% ============Price Penalty Factor (PPF)==================================
%     h=[(a.*Pmax1.^2+b.*Pmax1+c)]./[(d.*Pmax1.^2+e.*Pmax1+ff)];%PPF
  h=[(a.*Pmin1.^2+b.*Pmin1+c)]./[(d.*Pmax1.^2+e.*Pmax1+ff)];%PPF
%% ========== CEED Fuel cost, Fuel cost and  Emission =====================
    for i=1:6
```

```
    f(i)=a(i)*p(i)^2+b(i)*p(i)+c(i)+(h(i)*(d(i)*p(i)^2+e(i)*p(i)+ff(i))); %
CEED Fuel cost of individual GEN
    fuel(i)=a(i)*p(i)^2+b(i)*p(i)+c(i);% Fuel cost (FC)of individual GEN
    Emission(i)=d(i)*p(i)^2+e(i)*p(i)+ff(i);% Emission value of individual
GEN
    end
    F=sum(f);               % CEED Fuel cost
    fuel=sum(fuel);         % Fuel cost
    Emission=sum(Emission); % Emission
    q=[F p PL fuel Emission ]; %Economic dispatch(PSO)1st level of Solution
    eval(['q' num2str(j) ' = (q)']);
    eval(['z' num2str(j) '=(q(:,1))']);
end
%% Sort the Fuel cost in ascending order
p=[p1; p2; p3; p4 ;p5;
p6;p7;p8;p9;p10;p11;p12;p13;p14;p15;p16;p17;p18;p19;p20;p21;p22;p23;p24;p25
;p26;p27;p28;p29;p30];%Print the real power P
p=p(:,(2:6)) % print the real power generator values
z=[z1 z2 z3 z4 z5 z6 z7 z8 z9 z10 z11 z12 z13 z14 z15 z16 z17 z18 z19 z20
z21 z22 z23 z24 z25 z26 z27 z28 z29 z30]; %Total fuel cost value
q=[q1; q2; q3 ;q4; q5; q6; q7; q8; q9; q10 ; q11; q12; q13 ;q14; q15; q16;
q17; q18; q19; q20 ;q21; q22; q23 ;q24; q25; q26; q27; q28; q29; q30
]%print the q=[FT, P1....Pn, FC,ET]
[aa bb]=min(z) % To find the minimum of FT
gbest1=q(bb,:) % To select the Gbest as best of all positions
gbest=q(bb,(3:7)) % To select the global best positions
%% % Economic dispatch (PSO) 2nd level of Solution
%% ===================New Velocity=========================================
for j=1:30
    for i=1:5
    w(i)= wmax-((i*(wmax-wmin))/5); % Calculate the inertia weight
    vnew=w(i)*v(j,i)+(c1*r1*(p(:,i)-p(:,i)))+(c2*r2*(gbest(:,i)-
p(:,i)));%New velocity
    eval(['vnew' num2str(i) ' = (vnew)']);
    end
end
vnew=[vnew1 vnew2 vnew3 vnew4 vnew5];
 %% =============check the limits of new velocities========================
for i=1:30
    for j=1:5
        if vnew(i,j)<=vmin(j)
        vnew(i,j)=vmin(j)
        elseif vnew(i,j)>=vmax(j)
        vnew(i,j)=vmax(j)
        end
    end
end
%% =============== New position=============================================
pneww=zeros(30,5) % Pre-allocation of new position (i.e.,pneww)
for i=1:30
    for j=1:5
    pneww(i,j)=p(i,j)+vnew(i,j); % New position
    end
eval(['pneww' num2str(i) ' = (pneww)'])
end
%% ============Check the limits of new position============================
for i=1:30
    for j=1:5
```

```
        if pneww(i,j)<=Pmin(j)
           pneww(i,j)=Pmin(j);
        elseif pneww(i,j)>=Pmax(j)
               pneww(i,j)=Pmax(j);
        end
    end
end
pnew=[pneww30]
%% ===============New dependent Generator =================================
for i=1:30
    pneww=pnew(i,:)
    pneww=[zeros pneww];
    x=B(1,1); % calculation of x
    y=zeros(5,1); %Pre-allocation of y
    for k=2:6
        for r=1:1
        y(k,r)=((B(k,r)+B(r,k))*pneww(k)); % Calculation of y
        end
    end
    y=sum(y)-1;
    PL=zeros(5,5); % Pre - allocation of PL
    for k=2:6
        for l=2:6
        PL(k,l)=pneww(k)*B(k,l)*pneww(l); % Calculation of PL
        end
    end
    PLr=sum(PL);
    PLs=sum(PLr);
    PL=PLs;
    p=sum(pneww(2:6));
    z=PD+PL-p;
    pdnew=-(y+(y.^2- 4.*x.*z).^(1/2))/(2*x); %New dependent Generator1
    demo1=-(y-(y.^2- 4.*x.*z).^(1/2))/(2*x); %New dependent Generator2
%% ====================New Dependent Generator Limit =====================
    if pdnew<=Pdmin % checking the slack bus generator limit
        pdnew=Pdmin;
    elseif pdnew>=Pdmax
        pdnew=Pdmax;
    else pdnew=pdnew;
    end
%% =================== New Transmission loss ==============================
    pneww=pneww(2:6) % New GEN real power excluding slack bus
    pneww=[pdnew pneww] % New GEN real power including slack bus
    for k=1:6
        for l=1:6
        PL(k,l)=pneww(k)*B(k,l)*pneww(l); % New Transmission loss
        end
    end
    Pl=sum(PL);
    PL=sum(Pl);
    eval(['PL' num2str(i) ' = (PL)']);
    eval(['pneww' num2str(i) ' = (pneww)']);
    eval(['pdnew' num2str(i) ' = (pdnew)']);
end
pnew=[pneww1; pneww2; pneww3; pneww4; pneww5; pneww6; pneww7; pneww8;
pneww9 ;pneww10; pneww11; pneww12; pneww13; pneww14; pneww15; pneww16;
pneww17; pneww18; pneww19 ;pneww20;pneww21; pneww22; pneww23; pneww24;
pneww25; pneww26; pneww27; pneww28; pneww29 ;pneww30;];
```

```
PL=[PL1; PL2; PL3; PL4 ;PL5; PL6; PL7; PL8 ;PL9; PL10 ; PL11;  PL12; PL13;
PL14; PL15; PL16; PL17; PL18; PL19; PL20; PL21; PL22; PL23; PL24; PL25;
PL26; PL27; PL28; PL29; PL30; ];
%% ====Calculate the New CEED Fuel cost,New Fuel cost and New Emission=====
for j=1:30
    for i=1:6

Fnew(i)=a(i)*pnew(j,i)^2+b(i)*pnew(j,i)+c(i)+(h(i)*(d(i)*pnew(j,i)^2+e(i)*p
new(j,i)+ff(i))); % New CEED Fuel cost FTnew
    fuelnew(i)=a(i)*pnew(j,i)^2+b(i)*pnew(j,i)+c(i);% New Fuel cost (FCnew)
    Emissionnew(i)=d(i)*pnew(j,i)^2+e(i)*pnew(j,i)+ff(i);%NewEmission ETnew
    end
    Fnew=sum(Fnew(:,:));
    fuelnew=sum(fuelnew(:,:));
    Emissionnew=sum(Emissionnew(:,:));
    eval(['Fnew' num2str(j) ' = (Fnew)']);
    eval(['fuelnew' num2str(j) ' = (fuelnew)']);
    eval(['Emissionnew' num2str(j) ' = (Emissionnew)']);
end
%% =====PSO 2nd level solution; qnew=[FTnew, P1new....Pnnew, FCnew,ETnew]==
Fnew=[Fnew1;Fnew2;Fnew3;Fnew4;Fnew5;Fnew6;Fnew7;Fnew8;Fnew9;Fnew10;Fnew11;F
new12;Fnew13;Fnew14;Fnew15;Fnew16;Fnew17;Fnew18;Fnew19;Fnew20;Fnew21;Fnew22
;Fnew23;Fnew24;Fnew25;Fnew26;Fnew27;Fnew28;Fnew29;Fnew30;];
fuelnew=[fuelnew1;fuelnew2;fuelnew3;fuelnew4;fuelnew5;fuelnew6;fuelnew7;fue
lnew8;fuelnew9;fuelnew10;fuelnew11;fuelnew12;fuelnew13;fuelnew14;fuelnew15;
fuelnew16;fuelnew17;fuelnew18;fuelnew19;fuelnew20;fuelnew21;fuelnew22;fueln
ew23;fuelnew24;fuelnew25;fuelnew26;fuelnew27;fuelnew28;fuelnew29;fuelnew30;
];
Emissionnew=[Emissionnew1;Emissionnew2;Emissionnew3;Emissionnew4;Emissionne
w5;Emissionnew6;Emissionnew7;Emissionnew8;Emissionnew9;Emissionnew10;Emissi
onnew11;Emissionnew12;Emissionnew13;Emissionnew14;Emissionnew15;Emissionnew
16;Emissionnew17;Emissionnew18;Emissionnew19;Emissionnew20;Emissionnew21;Em
issionnew22;Emissionnew23;Emissionnew24;Emissionnew25;Emissionnew26;Emissio
nnew27;Emissionnew28;Emissionnew29;Emissionnew30;];
qnew=[Fnew pnew PL fuelnew Emissionnew]
%% ====Compare the PSO solution of 1st level with 2nd level (q and qnew)===
x=q(:,1); % Assisgn q to x
y=qnew(:,1); % Assisgn qnew to y
for i=1:30
    if x(i)<y(i); % compare q and  qnew
    z(i)=y(i);
    qrow=i;
    x1=(q(qrow,:));
    mat(i,:)=x1  % Store best solution in mat
    end
    if x(i)>y(i);% compare q and  qnew
    z(i)=x(i);
    qnewrow=i;
    y1=qnew(qnewrow,:);
    mat(i,:)=y1  % Store best solution in mat
    end
end
%% ==================Store the best PSO solution ==========================
eval(['mat' num2str(iter) ' = (mat)']);
eval(['matz' num2str(iter) '=(mat(:,1))']);
end % PSO Iteration Ends
%% ==================PSO solution =========================================
```

```
final=[mat1; mat2 ;mat3 ;mat4; mat5; mat6; mat7; mat8 ;mat9; mat10 ;mat11
;mat12; mat13; mat14 ;mat15 ;mat16; mat17; mat18; mat19;
mat20;mat21;mat22;mat23;mat24;mat25;mat26;mat27;mat28;mat29;mat30;mat31;mat
32;mat33;mat34;mat35;mat36;mat37;mat38;mat39;mat40]% PSO Solution of all
the iterations
[row col]=min(final(:,1))% Find the best solution of PSO from the solution
of each iteration
finalans=final(col,:) % print the best solution of PSO
iteration=fix(col/30)+1 % print the iteration Number of the best solution
%% Graph: PSO solution to Economic Dispatch Problem
y_axis=final(:,1);
graph=plot(y_axis, '*');
hold on
plot(finalans(1,1),'o')
set(graph,'Color','red','LineWidth',2)
ANS=[finalans(:,:) iteration toc]
P1=ANS(2);P2=ANS(3);P3=ANS(4);P4=ANS(5);P5=ANS(6);P6=ANS(7);PL=ANS(8);FC=AN
S(9);ET=ANS(10);FT=ANS(1);iter=ANS(11);time=ANS(12);
end
%% The PSO Program for Economic Dispatch ----------Ends
%=====================================================================
```

APPENDIX G: MATLAB PROGRAM TO START THE CLUSTER

Appendix G1: MATLAB script file – Start_Cluster.m

```
% M-File: startcluster.m
% Description: This code used to start Matlab workers from head node
%This code is developed to Start workers of the Cluster Computers from head
node using Matlab Distributed Computing Engine (MDCE)
% Set the name of the jobmanager, head node and workers in the MDCE
configuration as given below
% Name of the jobmanager : myjobmanager
% Name of the head node  : mat32-head
% Name of the workers    : mat32-wrk1, mat32-wrk2,....,mat32-wrk32
%=====================================================================
directory = pwd; % Set default directory as present work directory (pwd)
cd(fullfile(matlabroot, 'toolbox', 'distcomp', 'bin')); % Set the Matlab
path
h = waitbar(0, 'Starting the jobmanager ...'); % Open Wait bar
system('startjobmanager -clean -name myjobmanager -remotehost mat32-head');
% Start jobmanager
set(get(get(h,'currentAxes'),'title'),'string', 'Starting the workers ...')
% Set the wait bar string as: 'Starting the Workers'
waitbar(1/17, h); % Intialize the waitbar
%=====================================================================
try
    for i=1:16
        system(['startworker -clean -name ' sprintf('mat32-wrk%d',i) ...
             ' -jobmanager myjobmanager -jobmanagerhost mat32-head', ...
             ' -remotehost ' sprintf('mat32-wrk%d', i) ' &']); % Start the
matlab workers from the head node
        waitbar((i+1)/17, h); % Wait bar status
    end
catch
end
close(h);% Close the wait bar
cd(directory);% Set the directory
%=====================================================================
```

APPENDIX H: MATLAB PROGRAM FOR SEQUENTIAL CALCULATION OF THE SINGLE AREA CEED PROBLEM USING LAGRANGE'S ALGORITHM

Appendix H1: **MATLAB script file: SACEEDP_Casestudy1.m**

```
% M-File : Multi-area_wholepowersystemCasestudy1.m
%Four area forty generator system
% ==================M-File Description====================================
% The M-File is used to solve CEED problem of Multi-
area_wholepowersystemCasestudy1 with various price penalty factors using
Lagrange’s Algorithm
clear all
clc;
tic;
% ==Economic dispatch problem including losses and generator limits========
% ============Initial Lagrange’s variable==================================
lambda=10;     % Initial lambda
epsilon=0.001;          % Tolerance value
alfa=0.001;             % Incremental deltalambda
n=40;                   % No of generators
m=50000;                % Total number of iterations
PD=10500;               % Power demand
%% ======Single area Generator fuel Cost coefficients =====================
%% ============Data are given in the reference paper (Basu, 2011)==========
%The data are given in the format[GenNum Pmin Pmax a b c d e f]============
Data=[.....
1   36  114  0.0069     6.73    94.705  0.048   -2.22   60
2   36  114  0.0069     6.73    94.705  0.048   -2.22   60
3   60  120  0.02028    7.07    309.54  0.0762  -2.36   100
4   80  190  0.00942    8.18    369.03  0.054   -3.14   120
5   47  97   0.0114     5.35    148.89  0.085   -1.89   50
6   68  140  0.01142    8.05    222.33  0.0854  -3.08   80
7   110 300  0.00357    8.03    287.71  0.0242  -3.06   100
8   135 300  0.00492    6.99    391.98  0.031   -2.32   130
9   135 300  0.00573    6.6     455.76  0.0335  -2.11   150
10  130 300  0.00605    12.9    722.82  0.425   -4.34   280
11  94  375  0.00515    12.9    635.2   0.0322  -4.34   220
12  94  375  0.00569    12.8    654.69  0.0338  -4.28   225
13  125 500  0.00421    12.5    913.4   0.0296  -4.18   300
14  125 500  0.00752    8.84    1760.4  0.0512  -3.34   520
15  125 500  0.00752    8.84    1760.4  0.0496  -3.55   510
16  125 500  0.00752    8.84    1760.4  0.0496  -3.55   510
17  220 500  0.00313    7.97    647.85  0.0151  -2.68   220
18  220 500  0.00313    7.95    649.69  0.0151  -2.66   222
19  242 550  0.00313    7.97    647.83  0.0151  -2.68   220
20  242 550  0.00313    7.97    647.81  0.0151  -2.68   220
21  254 550  0.00298    6.63    785.96  0.0145  -2.22   290
22  254 550  0.00298    6.63    785.96  0.0145  -2.22   285
23  254 550  0.00284    6.66    794.53  0.0138  -2.26   295
24  254 550  0.00284    6.66    794.53  0.0138  -2.26   295
25  254 550  0.00277    7.1     801.32  0.0132  -2.42   310
26  254 550  0.00277    7.1     801.32  0.0132  -2.42   310
27  10  150  0.52124    3.33    1055.1  1.842   -1.11   360
28  10  150  0.52124    3.33    1055.1  1.842   -1.11   360
29  10  150  0.52124    3.33    1055.1  1.842   -1.11   360
30  47  97   0.0114     5.35    148.89  0.085   -1.89   50
31  60  190  0.0016     6.43    222.92  0.0121  -2.08   80
32  60  190  0.0016     6.43    222.92  0.0121  -2.08   80
```

```
33  60  190  0.0016    6.43   222.92  0.0121  -2.08   80
34  90  200  0.0001    8.95   107.87  0.0012  -3.48   65
35  90  200  0.0001    8.62   116.58  0.0012  -3.24   70
36  90  200  0.0001    8.62   116.58  0.0012  -3.24   70
37  25  110  0.0161    5.88   307.45  0.095   -1.98   100
38  25  110  0.0161    5.88   307.45  0.095   -1.98   100
39  25  110  0.0161    5.88   307.45  0.095   -1.98   100
40  242 550  0.00313   7.97   647.83  0.0151  -2.68   220];
a=Data(:,4)';
b=Data(:,5)';
c=Data(:,6)';
%% Single area Emmis on coefficients
d=Data(:,7)';
e=Data(:,8)';
f=Data(:,9)';
%%  Single area Generator real power limits
Pmin=Data(:,2)';
Pmax=Data(:,3)';
 %================Lagrange's algorithm starts here ========================
for iter=1:m
    iter        % To Print the each iteration number
% ==========Calculation of Price penalty factor===========================
  h=[(a.*Pmin.^2+b.*Pmin+c)]./[(d.*Pmax.^2+e.*Pmax+f)];
% h=[(a.*Pmax.^2+b.*Pmax+c)]./[(d.*Pmax.^2+e.*Pmax+f)];
% h=[(a.*Pmin.^2+b.*Pmin+c)]./[(d.*Pmin.^2+e.*Pmin+f)];
% h=[(a.*Pmax.^2+b.*Pmax+c)]./[(d.*Pmin.^2+e.*Pmin+f)];
                                    % Equation (3.7 or 3.8 or 3.9 or 3.10)
=================Calculation of generator real power ==========
    E=[(diag(a+(h.*d))./lambda)];         % Equation (3.19)
    D=0.5*((1-(b+h.*e)./lambda))';        % Equation (3.19)
    P=E\D;                                % Equation (3.21)
%======================Generator limit checking ===========================
%======================Generator limit checking ===========================
    for i=1:n
        if P(i)<=Pmin(i);
            P(i)=Pmin(i);
        elseif P(i)>=Pmax(i);
            P(i)=Pmax(i);
        else P(i)=P(i);
        end
    end
% ======================Calculation of deltalambda ========================
    deltalambda=PD;
    for i=1:n
        deltalambda=deltalambda-P(i);           % Equation (3.22)
    end
    if abs(deltalambda)<=epsilon | iter>=m
        break
    else lambda=lambda+(deltalambda*alfa)       % Equation (3.23)
    end
end
%====== Calculation of fuel cost, emission and CEED values ==========
fuelcost=0;
emissioncost=0;
CEED=0;
for i=1:n
    fuelcost=fuelcost+a(i)*P(i)^2+b(i)*P(i)+c(i);
                                                        % Equation (3.1)
```

```
    emissioncost=emissioncost+d(i)*P(i)^2+e(i)*P(i)+f(i);
                                                          % Equation (3.5)
    CEED=
CEED+a(i)*P(i)^2+b(i)*P(i)+c(i)+h(i)*(d(i)*P(i)^2+e(i)*P(i)+f(i));
                                                           % Equation (3.6)
end
%==== Printing of Real power, Transmission loss, Fuel cost, Emission and
CEED===
lambda
deltalambda
P
FC=sum(fuelcost(:,:))
ET=sum(emissioncost(:,:))
FT=sum(CEED(:,:))
toc
ans=[lambda P' FC  ET FT iter toc]
%===========================================================================
```

Appendix H2: **MATLAB script file: SACEEDP_Casestudy2.m**

```
% M-File : SACEEDP_Casestudy2.m
%Twelve generator system with transmission line constraints
% ==================M-File Description=====================================
% The M-File is used to solve CEED problem of single area whole power
system prior to decompose into multi-area - Casestudy2 with various price
penalty factors using Lagrange's Algorithm
clear all
clc;
tic;
% ==Economic dispatch problem including losses and generator limits========
% ============Initial Lagrange's variable=============================
lambda=40;              % Initial lambda
epsilon=0.001;          % Tolerance value
alfa=0.001;             % Incremental deltalambda
n=12;                   % No of generators
m=10000;                 % Total number of iterations
PD=2090;                % Area power demand
% PD=[500 410 580 600]; % Area power demand
%% Multi area Generator fuel Cost coefficients
a=[0.03546 0.02111 0.01799...
   0.15247 0.02803 0.14834...
   0.10587 0.07505 0.11934...
   0.10587 0.13552 0.08963];
b=[38.30553 36.32782 38.27041...
   38.53973 40.39655 38.34001...
   46.15916 43.83562 50.63211...
   46.15916 41.03782 33.56211];
c=[1243.5311 1658.5696 1356.6592...
   0756.7989 0449.9977 0558.5696...
   0451.3251 0673.0267 0530.7199...
   0851.3251 1038.533  1285.907];
%% Multi area Emmison coefficients
d=[0.00683 0.00461 0.00461...
   0.00484 0.00754 0.00661...
   0.00914 0.00533 0.00674...
   0.00728 0.00479 0.00387];
e=[-0.54551 -0.51160  -0.51160...
   -0.32767 -0.54551  -0.63262...
```

```
   -0.43211 -0.61173  -0.49731...
   -0.6821  -0.50660  -0.49340];
f=[40.26690  42.89553  42.89553...
   33.85932  50.639310 45.83267...
   48.21560  52.45210  41.10420...
   30.36320  25.17650  27.75490];
%%  Multi area Generator real power limits
Pmin=[35 130 125....
      10 35 125....
      15 30 50....
      15 30 50];
Pmax=[210 325 315....
      150 110 215....
      175 215 335....
      175 215 335];
  %% Transmission loss coefficients
B=[ 0.000071  0.00003   0.000025
    0.00003   0.000069  0.000032
    0.000025  0.000032  0.00008
    0.000056  0.000045  0.000015
    0.000023  0.000042  0.000047
    0.000032  0.000023  0.000027
    0.00002   0.000028  0.000053
    0.000086  0.000034  0.000016
    0.000053  0.000016  0.000028
    0.000074  0.00003   0.000025
    0.000049  0.000069  0.000037
    0.000022  0.000032  0.000083];
B=[B';zeros(9,12)]
B01=zeros(1,12);
B00=0;
     %================Lagrange's algorithm starts here ===================
for iter=1:m
    iter         % To Print the each iteration number
    % ===========Calculation of Price penalty factor=================
%     h=[(a.*Pmin.^2+b.*Pmin+c)]./[(d.*Pmax.^2+e.*Pmax+f)];   % Equation
(3.7 or 3.8 or 3.9 or 3.10)
     h=[( a.*Pmax.^2+b.*Pmax+c)]./[(d.*Pmax.^2+e.*Pmax+f)];   % Equation
(3.7 or 3.8 or 3.9 or 3.10)
    % =================Calculation of generator real power ==========
    E=[(diag(a+(h.*d))./lambda)+B];          % Equation (3.19)
    D=0.5*((1-(b+h.*e)./lambda-B01))';        % Equation (3.19)
    P=E\D;                              % Equation (3.21)

%======================Generator limit checking =================
  %======================Generator limit checking =================
    for i=1:n
        if P(i)<=Pmin(i);
            P(i)=Pmin(i);
        elseif P(i)>=Pmax(i);
            P(i)=Pmax(i);
        else P(i)=P(i);
        end
    end
    %========== Calculation of Transmission loss=====================
    PL1=B00;
    PL2=zeros(1,n);
    for i=1:n
```

```
        PL2(i)=(B01(i)*P(i));       % Equation (3.3)
    end
    PL2=sum(PL2(:,:));
    PL3=zeros(n,n);
    for i=1:n
        for j=1:n
            PL3(i,j)=P(i)*B(i,j)*P(j);  % Equation (3.3)
        end
    end
    PL3=sum(PL3(:,:));
    PL3=sum(PL3(:,:));
    PL=PL1+PL2+PL3;
    % ======================Calculation of deltalambda ==============
%     PL=0.0;
    deltalambda=0.0;
    deltalambda=PD+PL;
    for i=1:n
        deltalambda=deltalambda-P(i);       % Equation (3.22)
    end
    if abs(deltalambda)<=epsilon | iter>=m
        break
    else lambda=lambda+deltalambda*alfa  % Equation (3.23)
    end
end
%====== Calculation of fuel cost, emission and CEED values ==========
fuelcost=0;
emissioncost=0;
CEED=0;
for i=1:n
    fuelcost=fuelcost+a(i)*P(i)^2+b(i)*P(i)+c(i); % Equation (3.1)
    emissioncost=emissioncost+d(i)*P(i)^2+e(i)*P(i)+f(i); % Equation (3.5)
    CEED=
CEED+a(i)*P(i)^2+b(i)*P(i)+c(i)+h(i)*(d(i)*P(i)^2+e(i)*P(i)+f(i)); %
Equation (3.6)
end
%==== Printing of Real power, Transmission loss, Fuel cost, Emission and
CEED===
lambda
P
PL
FC=sum(fuelcost(:,:))
ET=sum(emissioncost(:,:))
FT=sum(CEED(:,:))
toc
ans=[lambda P' PL FC  ET FT iter toc]
%===========================================================================
```

APPENDIX I: MATLAB PROGRAM FOR SEQUENTIAL CALCULATION OF THE MULTI-AREA CEED PROBLEM USING LAGRANGE'S ALGORITHM

Appendix I1: MATLAB script file: MACEEDP_Sequnetial_Casestudy1.m

```
% M-File : MACEEDP_Sequnetial_Casestudy1.m
% Four-area and  10 generator in each area
% Transmission loss is omitted in this case
% 6 tie-lines are used to interconnect the four-area power system
% ==================M-File Description=================================
% The M-File is used to solve Multiarea economic emission dispatch problem
using various price penalty factors by Lagrange’s Algorithm
% The software is developed for MAEED problem in a sequential way
clear all
clc
%% Multi area economic dispatch data taken from the reference paper
(Basu,2011)
tic
m=4;                        % No of area
n=40;                       % Total No of Generators in all the area
TL=6;                       % NO of Tie lines
N_iter=25000;               % No of iteration
lambda=[50 50 50 50];       % Initial Lagrangian variable
lambda1=lambda(1);lambda2=lambda(2);lambda3=lambda(3);lambda4=lambda(4);
lambda={lambda(1),lambda(2),lambda(3),lambda(4)}
% Total power demand of the system, PD=10500[MW]
% Area 1 has 15% of total power demand, ie PD=(15/100)*10500=1575[MW]
% Area 2 has 40% of total power demand, ie PD=(40/100)*10500=4200[MW]
% Area 3 has 30% of total power demand, ie PD=(30/100)*10500=3150[MW]
% Area 4 has 15% of total power demand, ie PD=(15/100)*10500=1575[MW]
PD=[1575    4200    3150    1400]; % Area power demand
alfa=[0.01 0.01 0.01 0.01];     %Incremental deltalambda Tolerance value of
each area
epsilon=[0.01 0.01 0.01 0.01];  % Tolerance value of each area
%% Multi area Generator fuel Cost coefficients
%% Single area Generator fuel Cost coefficients
%% ============Data are given in the reference paper (Basu, 2011)==========
%The data are given in the format[GenNum Pmin Pmax a b c d e f]============
Data=[.....
1   36  114     0.0069      6.73    94.705  0.048   -2.22   60
2   36  114     0.0069      6.73    94.705  0.048   -2.22   60
3   60  120     0.02028     7.07    309.54  0.0762  -2.36   100
4   80  190     0.00942     8.18    369.03  0.054   -3.14   120
5   47  97      0.0114      5.35    148.89  0.085   -1.89   50
6   68  140     0.01142     8.05    222.33  0.0854  -3.08   80
7   110 300     0.00357     8.03    287.71  0.0242  -3.06   100
8   135 300     0.00492     6.99    391.98  0.031   -2.32   130
9   135 300     0.00573     6.6     455.76  0.0335  -2.11   150
10  130 300     0.00605     12.9    722.82  0.425   -4.34   280
11  94  375     0.00515     12.9    635.2   0.0322  -4.34   220
12  94  375     0.00569     12.8    654.69  0.0338  -4.28   225
13  125 500     0.00421     12.5    913.4   0.0296  -4.18   300
14  125 500     0.00752     8.84    1760.4  0.0512  -3.34   520
15  125 500     0.00752     8.84    1760.4  0.0496  -3.55   510
16  125 500     0.00752     8.84    1760.4  0.0496  -3.55   510
17  220 500     0.00313     7.97    647.85  0.0151  -2.68   220
18  220 500     0.00313     7.95    649.69  0.0151  -2.66   222
```

```
19  242 550     0.00313     7.97    647.83  0.0151  -2.68   220
20  242 550     0.00313     7.97    647.81  0.0151  -2.68   220
21  254 550     0.00298     6.63    785.96  0.0145  -2.22   290
22  254 550     0.00298     6.63    785.96  0.0145  -2.22   285
23  254 550     0.00284     6.66    794.53  0.0138  -2.26   295
24  254 550     0.00284     6.66    794.53  0.0138  -2.26   295
25  254 550     0.00277     7.1     801.32  0.0132  -2.42   310
26  254 550     0.00277     7.1     801.32  0.0132  -2.42   310
27  10  150     0.52124     3.33    1055.1  1.842   -1.11   360
28  10  150     0.52124     3.33    1055.1  1.842   -1.11   360
29  10  150     0.52124     3.33    1055.1  1.842   -1.11   360
30  47  97      0.0114      5.35    148.89  0.085   -1.89   50
31  60  190     0.0016      6.43    222.92  0.0121  -2.08   80
32  60  190     0.0016      6.43    222.92  0.0121  -2.08   80
33  60  190     0.0016      6.43    222.92  0.0121  -2.08   80
34  90  200     0.0001      8.95    107.87  0.0012  -3.48   65
35  90  200     0.0001      8.62    116.58  0.0012  -3.24   70
36  90  200     0.0001      8.62    116.58  0.0012  -3.24   70
37  25  110     0.0161      5.88    307.45  0.095   -1.98   100
38  25  110     0.0161      5.88    307.45  0.095   -1.98   100
39  25  110     0.0161      5.88    307.45  0.095   -1.98   100
40  242 550     0.00313     7.97    647.83  0.0151  -2.68   220];
%% Multi area fuel cost coefficients
a=Data(:,4)';
a={a(1:10),a(11:20),a(21:30),a(31:40)}
b=Data(:,5)';
b={b(1:10),b(11:20),b(21:30),b(31:40)}
c=Data(:,6)';
c={c(1:10),c(11:20),c(21:30),c(31:40)}
%% Multi area Emmis on coefficients
d=Data(:,7)';
d={d(1:10),d(11:20),d(21:30),d(31:40)}
e=Data(:,8)';
e={e(1:10),e(11:20),e(21:30),e(31:40)}
f=Data(:,9)';
f={f(1:10),f(11:20),f(21:30),f(31:40)}
%%  Multi area Generator real power limits
Pmin=Data(:,2)';
Pmin1={Pmin(1:10),Pmin(11:20),Pmin(21:30),Pmin(31:40)}
Pmax=Data(:,3)';
Pmax1={Pmax(1:10),Pmax(11:20),Pmax(21:30),Pmax(31:40)}
%% Intial Tie line values between 100 to 200 [MW] (Assumed)
PT_mj=randi([100 200],1,6)
PT_jm=randi([100 200],1,6)
%% Tie line limits (Assumed)
PTmin_mj=[100 100 50....
         100 50 50];
PTmin_jm=[100 100 100....
          50 50 50];
PTmax_mj=[200 200 100....
          200 100 100];
PTmax_jm=[200 200 200....
          100 100 100];
%% Tie line coefficients (Assumed)
q_mj=rand(1,6);
q_jm=rand(1,6);
%% Tie line Fractional loss rate values (Assumed)
flr_jm=rand(1,6);
```

```
%% Tie line incremental value (Assumed)
alpha_mj=[0.01 0.01 0.01....
          0.01 0.01 0.01];
alpha_jm=[0.01 0.01 0.01....
          0.01 0.01 0.01];
% Generator real power calculation
for iter=1:N_iter
    iter
for i=1:m
h=(a{i}.*Pmin1{i}.^2+b{i}.*Pmin1{i}+c{i})./(d{i}.*Pmax1{i}.^2+e{i}.*Pmax1{i
}+f{i});
%
h=(a{i}.*Pmax1{i}.^2+b{i}.*Pmax1{i}+c{i})./(d{i}.*Pmax1{i}.^2+e{i}.*Pmax1{i
}+f{i});
%
h=(a{i}.*Pmin1{i}.^2+b{i}.*Pmin1{i}+c{i})./(d{i}.*Pmin1{i}.^2+e{i}.*Pmin1{i
}+f{i});
%
h=(a{i}.*Pmax1{i}.^2+b{i}.*Pmax1{i}+c{i})./(d{i}.*Pmin1{i}.^2+e{i}.*Pmin1{i
}+f{i});
     E=diag((a{i}+h.*d{i})./lambda{i});
     D=0.5*(1-(b{i}+h.*e{i})./lambda{i})';
     Power=E\D;
     Power1(i,:)=Power
     h1(i,:)=h
end
   P=reshape(Power1',1,40)
   h=reshape(h1',1,40)
   h={h(1:10),h(11:20),h(21:30),h(31:40)}
%% Generator Real powers
P=[P(1:10),P(11:20),P(21:30),P(31:40)]
%% Generator real power limits
for i=1:n         % Total Number of generators in all the four area is n=12
    if P(i)<=Pmin(i)
    P(i)=Pmin(i)
    elseif P(i)>=Pmax(i)
    P(i)=Pmax(i)
    else P(i)=P(i)
    end
end
%% Tie line power flow
 ePT_mj=q_mj+lambda1;
 PT_mj=(PT_mj-alpha_mj.*ePT_mj);
 ePT_jm=q_jm-(1-flr_jm).*lambda1  ;
 PT_jm=PT_jm-alpha_jm.*ePT_jm;
%% Tie Line limits
TL=6;
for i=1:TL
        if PT_mj(i)<=PTmin_mj(i)
           PT_mj(i)=PTmin_mj(i);
        elseif PT_mj(i)>=PTmax_mj(i);
            PT_mj(i)=PTmax_mj(i);
        else PT_mj(i)=PT_mj(i);
        end
end
for i=1:TL
        if PT_jm(i)<=PTmin_jm(i);
            PT_jm(i)=PTmin_jm(i);
```

```
        elseif PT_jm(i)>=PTmax_jm(i);
            PT_jm(i)=PTmax_jm(i);
        else PT_jm(i)=PT_jm(i);
        end
end
Tielinepower=PT_mj-(1-flr_jm).*PT_jm;
% Tie-line power in the order PT=[PT12 PT13 PT14 PT23 PT24 PT34]
Tielinepower1=Tielinepower(1)+Tielinepower(2)+Tielinepower(3);
Tielinepower2=Tielinepower(1)+Tielinepower(4)+Tielinepower(5);
Tielinepower3=Tielinepower(2)+Tielinepower(4)+Tielinepower(6);
Tielinepower4=Tielinepower(3)+Tielinepower(5)+Tielinepower(6);
%% Incremental lambda
P1=P(1:10);P2=P(11:20);P3=P(21:30);P4=P(31:40);
deltalambda1=(PD(1)+Tielinepower1)-sum(P1);
deltalambda2=(PD(2)+Tielinepower2)-sum(P2);
deltalambda3=(PD(3)+Tielinepower3)-sum(P3);
deltalambda4=(PD(4)+Tielinepower4)-sum(P4);
deltalambda=[deltalambda1 deltalambda2 deltalambda3 deltalambda4]
if abs(deltalambda1)<=epsilon(1) & iter>=N_iter
    break
else  lambda1=lambda1+(deltalambda1*alfa(1))
end
if abs(deltalambda2)<=epsilon(2) & iter>=N_iter
    break
else  lambda2=lambda2+(deltalambda2*alfa(2))
end
if abs(deltalambda3)<=epsilon(3) & iter>=N_iter
    break
else  lambda3=lambda3+(deltalambda3*alfa(3))
end
if abs(deltalambda4)<=epsilon(4) & iter>=N_iter
    break
else  lambda4=lambda4+(deltalambda4*alfa(4))
end
lambda={lambda1,lambda2,lambda3,lambda4}
end
deltalambda=[deltalambda1 deltalambda2 deltalambda3 deltalambda4]
P
Tielinepower
PT_mj
PT_jm
P={P(1:10),P(11:20),P(21:30),P(31:40)}
%=== Calculation of fuel cost, emission and CEED values of MAEED problem
solution===
for i=1:m
fuelcost_area=sum(a{i}.*P{i}.^2+b{i}.*P{i}+c{i});
fuelcost(i,:)=fuelcost_area;
emissioncost_area=sum(d{i}.*P{i}.^2+e{i}.*P{i}+f{i});
emissioncost(i,:)=emissioncost_area;
ceed_area=sum(a{i}.*P{i}.^2+b{i}.*P{i}+c{i}+h{i}.*(d{i}.*P{i}.^2+e{i}.*P{i}
+f{i}));
ceed(i,:)=ceed_area;
end
fuelcost=sum(fuelcost)
emissioncost=sum(emissioncost)
ceed=sum(ceed)
ans=[fuelcost emissioncost ceed iter toc]
toc
```

Appendix I2: **MATLAB script file: MACEEDP_Sequnetial_Casestudy2.m**

```
% M-File : MACEEDP_Sequnetial_Casestudy2.m
% ==================M-File Description====================================
% The M-File is used to solve Multi-area CEED problem using various price
penalty factors by Lagrange's Algorithm
% The software is developed for MAED problem in a sequential way
clear all
clc
%% Multi area economic dispatch data
m=4;      % No of area
n=12;     % Total No of Generators in all the area
NG=3;     % No of Generators in each area
TL=6;     % NO of Tie lines
N_iter=10000; % No of iteration
lambda=[100 100 100 100];
lambda1=lambda(1);lambda2=lambda(2);lambda3=lambda(3);lambda4=lambda(4);
lambda={lambda(1),lambda(2),lambda(3),lambda(4)}
PD=[500 410 580 600]; % Area power demand
%PD=[525 435 605 625]; % Area power demand
alfa=[0.01 0.01 0.01 0.01]; %Incremental deltalambda Tolerance value of
each area
epsilon=[0.01 0.01 0.01 0.01]; % Tolerance value of each area
%% Multi area Generator fuel Cost coefficients
a=[0.03546 0.02111 0.01799...
   0.15247 0.02803 0.14834...
   0.10587 0.07505 0.11934...
   0.10587 0.13552 0.08963];
a={a(1:3),a(4:6),a(7:9),a(10:12)}
b=[38.30553 36.32782 38.27041...
   38.53973 40.39655 38.34001...
   46.15916 43.83562 50.63211...
   46.15916 41.03782 33.56211];
b={b(1:3),b(4:6),b(7:9),b(10:12)}
c=[1243.5311 1658.5696 1356.6592...
   0756.7989 0449.9977 0558.5696...
   0451.3251 0673.0267 0530.7199...
   0851.3251 1038.533  1285.907];
c={c(1:3),c(4:6),c(7:9),c(10:12)}
%% Multi area Emission coefficients
d=[0.00683 0.00461 0.00461...
   0.00484 0.00754 0.00661...
   0.00914 0.00533 0.00674...
   0.00728 0.00479 0.00387];
d={d(1:3),d(4:6),d(7:9),d(10:12)}
e=[-0.54551 -0.51160  -0.51160...
   -0.32767 -0.54551  -0.63262...
   -0.43211 -0.61173  -0.49731...
   -0.6821  -0.50660  -0.49340];
e={e(1:3),e(4:6),e(7:9),e(10:12)}
f=[40.26690  42.89553  42.89553...
   33.85932  50.639310 45.83267...
   48.21560  52.45210  41.10420...
   30.36320  25.17650  27.75490];
f={f(1:3),f(4:6),f(7:9),f(10:12)}
%%  Multi area Generator real power limits
```

```
Pmin=[35 130 125....
      10 35 125....
      15 30 50....
      15 30 50];
Pmin1={Pmin(1:3),Pmin(4:6),Pmin(7:9),Pmin(10:12)}
Pmax=[210 325 315....
      150 110 215....
      175 215 335....
      175 215 335];
Pmax1={Pmax(1:3),Pmax(4:6),Pmax(7:9),Pmax(10:12)}
%% Intial Tie line values (Assumed)
PT_mj=[10 15 12....
       20 18 29];
PT_jm=[15 18 20....
       14 22 19];
%% Tie line limits
PTmin_mj=[5 5 5....
          5 5 5];
PTmin_jm=[5 5 5....
          5 5 5];
PTmax_mj=[60 50 60....
          60 60 50];
PTmax_jm=[50 60 60....
          60 50 60];
%% Tie line coefficients
q_mj=rand(1,6);
q_jm=rand(1,6);
%% Tie line Fractional loss rate values
flr_jm=[0.11 0.21 0.14....
        0.16 0.22 0.11];
%% Tie line incremental value
alpha_mj=[0.000001 0.000001 0.000001....
          0.000001 0.000001 0.000001];
alpha_jm=[0.00001 0.00001 0.00001....
          0.00001 0.00001 0.00001];
%% Transmission loss coefficients
B_area=[ 0.000071  0.00003   0.000025
         0.00003   0.000069  0.000032
         0.000025  0.000032  0.00008
         0.000056  0.000045  0.000015
         0.000023  0.000042  0.000047
         0.000032  0.000023  0.000027
         0.00002   0.000028  0.000053
         0.000086  0.000034  0.000016
         0.000053  0.000016  0.000028
         0.000074  0.00003   0.000025
         0.000049  0.000069  0.000037
         0.000022  0.000032  0.000083];

B_area1={B_area((1:3),:),B_area((4:6),:),B_area((7:9),:),B_area((10:12),:
)}
% Generator real power calculation
for iter=1:N_iter
    iter
for i=1:m

%h=(a{i}.*Pmin1{i}.^2+b{i}.*Pmin1{i}+c{i})./(d{i}.*Pmax1{i}.^2+e{i}.*Pmax
1{i}+f{i});
```

```
h=(a{i}.*Pmax1{i}.^2+b{i}.*Pmax1{i}+c{i})./(d{i}.*Pmax1{i}.^2+e{i}.*Pmax1
{i}+f{i});

%h=(a{i}.*Pmin1{i}.^2+b{i}.*Pmin1{i}+c{i})./(d{i}.*Pmin1{i}.^2+e{i}.*Pmin
1{i}+f{i});

%h=(a{i}.*Pmax1{i}.^2+b{i}.*Pmax1{i}+c{i})./(d{i}.*Pmin1{i}.^2+e{i}.*Pmin
1{i}+f{i});
     E=diag((a{i}+h.*d{i})./lambda{i})+B_area1{i};
     D=0.5*(1-(b{i}+h.*e{i})./lambda{i})';
     Power=E\D;
     Power1(i,:)=Power
     h1(i,:)=h
end
   P=reshape(Power1',1,12)
   h=reshape(h1',1,12)
   h={h(1:3),h(4:6),h(7:9),h(10:12)}
%% Generator Real powers
P=[P(1:3),P(4:6),P(7:9),P(10:12)]
%% Generator real power limits
for i=1:n         % Total Number of generators in all the four area is
n=12
    if P(i)<=Pmin(i)
    P(i)=Pmin(i)
    elseif P(i)>=Pmax(i)
    P(i)=Pmax(i)
    else P(i)=P(i)
    end
end
%% Tie line power flow
 ePT_mj=q_mj+lambda1;
 PT_mj=(PT_mj-alpha_mj.*ePT_mj);
 ePT_jm=q_jm-(1-flr_jm).*lambda1  ;
 PT_jm=PT_jm-alpha_jm.*ePT_jm;
%% Tie Line limits
TL=6;
for i=1:TL
        if PT_mj(i)<=PTmin_mj(i)
           PT_mj(i)=PTmin_mj(i);
        elseif PT_mj(i)>=PTmax_mj(i);
            PT_mj(i)=PTmax_mj(i);
        else PT_mj(i)=PT_mj(i);
        end
end
for i=1:TL
        if PT_jm(i)<=PTmin_jm(i);
            PT_jm(i)=PTmin_jm(i);
        elseif PT_jm(i)>=PTmax_jm(i);
            PT_jm(i)=PTmax_jm(i);
        else PT_jm(i)=PT_jm(i);
        end
end
Tielinepower=PT_mj-(1-flr_jm).*PT_jm;
%% Transmission line loss
 B_area11=permute(reshape(B_area,3,4,[]),[1 3 2])
 P=reshape(P,3,1,[])
 PL_area=bsxfun(@times,P,reshape(P,1,3,[])).*B_area11
```

```
 PL=sum(reshape(PL_area,[],4))
%% Incremental lambda
P1=P(1:3);P2=P(4:6);P3=P(7:9);P4=P(10:12);
deltalambda1=(PD(1)+PL(1)+Tielinepower(1))-sum(P1);
deltalambda2=(PD(2)+PL(2)+Tielinepower(2))-sum(P2);
deltalambda3=(PD(3)+PL(3)+Tielinepower(3))-sum(P3);
deltalambda4=(PD(4)+PL(4)+Tielinepower(4))-sum(P4);
deltalambda=[deltalambda1 deltalambda2 deltalambda3 deltalambda4]
if abs(deltalambda1)<=epsilon(1) & iter>=N_iter
    break
else  lambda1=lambda1+(deltalambda1*alfa(1))
end
if abs(deltalambda2)<=epsilon(2) & iter>=N_iter
    break
else  lambda2=lambda2+(deltalambda2*alfa(2))
end
if abs(deltalambda3)<=epsilon(3) & iter>=N_iter
    break
else  lambda3=lambda3+(deltalambda3*alfa(3))
end
if abs(deltalambda4)<=epsilon(4) & iter>=N_iter
    break
else  lambda4=lambda4+(deltalambda4*alfa(4))
end
lambda={lambda1,lambda2,lambda3,lambda4}
end
deltalambda=[deltalambda1 deltalambda2 deltalambda3 deltalambda4]
P
Tielinepower
PT_mj
PT_jm
P={P(1:3),P(4:6),P(7:9),P(10:12)}
for i=1:m
fuelcost_area=sum(a{i}.*P{i}.^2+b{i}.*P{i}+c{i});
fuelcost(i,:)=fuelcost_area;
emissioncost_area=sum(d{i}.*P{i}.^2+e{i}.*P{i}+f{i});
emissioncost(i,:)=emissioncost_area;
ceed_area=sum(a{i}.*P{i}.^2+b{i}.*P{i}+c{i}+h{i}.*(d{i}.*P{i}.^2+e{i}.*P{
i}+f{i}));
ceed(i,:)=ceed_area;
end
fuelcost=sum(fuelcost)
emissioncost=sum(emissioncost)
ceed=sum(ceed)
```

APPENDIX J: MATLAB PROGRAM FOR PARALLEL CALCULATION OF THE MULTI-AREA CEED PROBLEM USING LAGRANGE'S ALGORITHM

Appendix J1: MATLAB script file: MACEEDP_Parallel_Casestudy1.m

```
% M-File : MACEEDP_Parallel_Casestudy1.m
% ==================M-File Description================================
% The M-File is used to solve Multiarea economic emission problem using
various price penalty factors by Lagrange's Algorithm
% Four area and 10 generator in each area
% The software is developed for MACEED problem in a sequential way
clear all
clc
%% Multi area economic dispatch data
tic
m=4;              % No of area
n=40;             % Total No of Generators in all the area
TL=6;             % NO of Tie lines
iter=20000;       % No of iteration
lambda=[50 50 50 50];
% Final Lagrangian variable of each area
lambda1=lambda(1);lambda2=lambda(2);lambda3=lambda(3);lambda4=lambda(4);
lambda={lambda(1),lambda(2),lambda(3),lambda(4)}
PD=[1575 4200 3150 1575]; % Area power demand
alfa=[0.01 0.01 0.01 0.01]; %Incremental deltalambda Tolerance value of
each area
epsilon=[0.01 0.01 0.01 0.01]; % Tolerance value of each area
%% Multi area Generator fuel Cost coefficients
%% ============Data are given in the reference paper (Basu, 2011)========
%The data are given in the format[GenNum Pmin Pmax a b c d e f]==========
Data=[.....
1   36  114     0.0069      6.73    94.705  0.048   -2.22   60
2   36  114     0.0069      6.73    94.705  0.048   -2.22   60
3   60  120     0.02028     7.07    309.54  0.0762  -2.36   100
4   80  190     0.00942     8.18    369.03  0.054   -3.14   120
5   47  97      0.0114      5.35    148.89  0.085   -1.89   50
6   68  140     0.01142     8.05    222.33  0.0854  -3.08   80
7   110 300     0.00357     8.03    287.71  0.0242  -3.06   100
8   135 300     0.00492     6.99    391.98  0.031   -2.32   130
9   135 300     0.00573     6.6     455.76  0.0335  -2.11   150
10  130 300     0.00605     12.9    722.82  0.425   -4.34   280
11  94  375     0.00515     12.9    635.2   0.0322  -4.34   220
12  94  375     0.00569     12.8    654.69  0.0338  -4.28   225
13  125 500     0.00421     12.5    913.4   0.0296  -4.18   300
14  125 500     0.00752     8.84    1760.4  0.0512  -3.34   520
15  125 500     0.00752     8.84    1760.4  0.0496  -3.55   510
16  125 500     0.00752     8.84    1760.4  0.0496  -3.55   510
17  220 500     0.00313     7.97    647.85  0.0151  -2.68   220
18  220 500     0.00313     7.95    649.69  0.0151  -2.66   222
19  242 550     0.00313     7.97    647.83  0.0151  -2.68   220
20  242 550     0.00313     7.97    647.81  0.0151  -2.68   220
21  254 550     0.00298     6.63    785.96  0.0145  -2.22   290
22  254 550     0.00298     6.63    785.96  0.0145  -2.22   285
23  254 550     0.00284     6.66    794.53  0.0138  -2.26   295
24  254 550     0.00284     6.66    794.53  0.0138  -2.26   295
25  254 550     0.00277     7.1     801.32  0.0132  -2.42   310
26  254 550     0.00277     7.1     801.32  0.0132  -2.42   310
27  10  150     0.52124     3.33    1055.1  1.842   -1.11   360
```

```
28  10  150     0.52124     3.33    1055.1  1.842   -1.11   360
29  10  150     0.52124     3.33    1055.1  1.842   -1.11   360
30  47  97      0.0114      5.35    148.89  0.085   -1.89   50
31  60  190     0.0016      6.43    222.92  0.0121  -2.08   80
32  60  190     0.0016      6.43    222.92  0.0121  -2.08   80
33  60  190     0.0016      6.43    222.92  0.0121  -2.08   80
34  90  200     0.0001      8.95    107.87  0.0012  -3.48   65
35  90  200     0.0001      8.62    116.58  0.0012  -3.24   70
36  90  200     0.0001      8.62    116.58  0.0012  -3.24   70
37  25  110     0.0161      5.88    307.45  0.095   -1.98   100
38  25  110     0.0161      5.88    307.45  0.095   -1.98   100
39  25  110     0.0161      5.88    307.45  0.095   -1.98   100
40  242 550     0.00313     7.97    647.83  0.0151  -2.68   220];
%% Multi area fuel cost coefficients
a=Data(:,4)';
a={a(1:10),a(11:20),a(21:30),a(31:40)}
b=Data(:,5)';
b={b(1:10),b(11:20),b(21:30),b(31:40)}
c=Data(:,6)';
c={c(1:10),c(11:20),c(21:30),c(31:40)}
%% Multi area Emmis on coefficients
d=Data(:,7)';
d={d(1:10),d(11:20),d(21:30),d(31:40)}
e=Data(:,8)';
e={e(1:10),e(11:20),e(21:30),e(31:40)}
f=Data(:,9)';
f={f(1:10),f(11:20),f(21:30),f(31:40)}
%%  Multi area Generator real power limits
Pmin=Data(:,2)';
Pmin1={Pmin(1:10),Pmin(11:20),Pmin(21:30),Pmin(31:40)}
Pmax=Data(:,3)';
Pmax1={Pmax(1:10),Pmax(11:20),Pmax(21:30),Pmax(31:40)}
%% Intial Tie line values between 100 to 200 [MW] (Assumed)
PT_mj=randi([100 200],1,6)
PT_jm=randi([100 200],1,6)
%% Tie line limits (Assumed)
PTmin_mj=[100 100 50....
         100 50 50];
PTmin_jm=[100 100 100....
          50 50 50];
PTmax_mj=[200 200 100....
          200 100 100];
PTmax_jm=[200 200 200....
          100 100 100];
%% Intial Tie line values (Assumed)
PT_mj=rand(1,6);
PT_jm=rand(1,6);
%% Tie line limits
PTmin_mj=[0.001  0.001   0.001....
          0.001   0.001  0.001];
PTmin_jm=[0.001  0.001   0.001....
          0.001   0.001  0.001];
PTmax_mj=[0.060  0.040  0.200...
          0.035  0.055  0.009];
PTmax_jm=[0.060  0.040  0.035...
          0.200  0.055  0.009];
%% Tie line coefficients
q_mj=rand(1,6);
```

```
q_jm=rand(1,6);
%% Tie line Fractional loss rate values
flr_jm=rand(1,6);
%% Tie line incremental value
alpha_mj=[0.01 0.01 0.01....
          0.01 0.01 0.01];
alpha_jm=[0.01 0.01 0.01....
          0.01 0.01 0.01];
% Generator real power calculation
for i=1:m
h=(a{i}.*Pmax1{i}.^2+b{i}.*Pmax1{i}+c{i})./(d{i}.*Pmax1{i}.^2+e{i}.*Pmax1
{i}+f{i});
     %
h=(a{i}.*Pmin1{i}.^2+b{i}.*Pmin1{i}+c{i})./(d{i}.*Pmax1{i}.^2+e{i}.*Pmax1
{i}+f{i});

%h=(a{i}.*Pmin1{i}.^2+b{i}.*Pmin1{i}+c{i})./(d{i}.*Pmin1{i}.^2+e{i}.*Pmin
1{i}+f{i});

%h=(a{i}.*Pmax1{i}.^2+b{i}.*Pmax1{i}+c{i})./(d{i}.*Pmin1{i}.^2+e{i}.*Pmin
1{i}+f{i});
     h1(i,:)=h
end
    h={h1(1:10),h1(11:20),h1(21:30),h1(31:40)}
     % ==============Matlab Parallel multiarea Economic Dispatch
code====================================
%Matlabpool open jobmanagerconfig1 %Opening 6 Matlab workers from the
cluster
k=1;
while k<=20000
t0 = clock; % start of the clock on second level calculation
jm = findResource('scheduler','configuration','jobmanagerconfig1')
disp(get(jm))%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% check the status of the
jobmanager
timingStart = tic;
start = tic;
pjob1=createParallelJob(jm,'Configuration','jobmanagerconfig1'); %create
parallel job from the jobmanager
get(pjob1) % Monitor the status of pjob1
%==========How long it takes to create a job=============================
time.jobCreateTime = toc(start)
description.jobCreateTime = 'Job creation time'
%========================================================================
Tasks = findTask(pjob1)
[pending running finished] = findTask(pjob1)
set(pjob1,'Configuration','jobmanagerconfig1')
set(pjob1,'MinimumNumberOfWorkers',10);
set(pjob1,'MaximumNumberOfWorkers',10);
set(pjob1,'FileDependencies',{'MACEEDP_Prallel_Casestudy1_funct.m'})
%get(jm)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Notice how jobmanager
is changing
%disp(get(pjob1))
%disp('busy workers')
%disp(get(jm, 'NumberOfbusyWorkers'))
%====================STARTING PARALLEL COMPUTING========================
%Create task- with 1 output arguments  and 7t input arguments,
%measure how long that takes and measure how long it takes to submit the
%job to the cluster
```

```
task1=createTask(pjob1,
@MACEEDP_Prallel_Casestudy1_funct,1,{m,a,h,d,lambda,b,e})
get(pjob1,'task')
%==========How long it takes to create a task===========================
time.taskCreateTime = toc(start)
description.taskCreateTime = 'Task creation time'
%======================================================================
submit(pjob1)
%===========How long it takes to submit a job===========================
time.submitTime = toc(start)
description.submitTime = 'Job submission time'
%======================================================================
%Once the job has been submitted, we hope all its tasks execute in
%parallel. Measure how long it takes for all the tasks to start and to
run
%to completion.
waitForState(pjob1,'finished')
time.jobWaitTime = toc(start)
description.pjobWaitTime = 'Job wait time'
get(jm)
%Tasks have now completed, so we are again executing a code in the matlab
%client. Measure how long it takes to retrieve all the job results
results=getAllOutputArguments(pjob1)
time.resultsTime = toc(start)
description.resultsTime = 'Results retrieval time'
%======================================================================
%============Verify that the job ran without errors=====================
errmsgs = get(pjob1.task, {'ErrorMessage'})
nonempty = ~cellfun(@isempty, errmsgs)
celldisp(errmsgs(nonempty))
%=====================================================================
%Measure the total time elapsed from creating the job up to the results
time.totalTime = toc(start)
operationtime = etime(clock,t0) %operation time for completion of
parallel computing
%destroy(pjob1)
P=cell2mat(results);
%% Generator real power limits
for i=1:n         % Total Number of generators in all the four area is
n=40
    if P(i)<=Pmin(i)
    P(i)=Pmin(i)
    elseif P(i)>=Pmax(i)
    P(i)=Pmax(i)
    else P(i)=P(i)
    end
end
%% Tie line power flow
 ePT_mj=q_mj+lambda1;
 PT_mj=(PT_mj-alpha_mj.*ePT_mj);
 ePT_jm=q_jm-(1-flr_jm).*lambda1  ;
 PT_jm=PT_jm-alpha_jm.*ePT_jm;
%% Tie Line limits
TL=6;
for i=1:TL
        if PT_mj(i)<=PTmin_mj(i)
           PT_mj(i)=PTmin_mj(i);
        elseif PT_mj(i)>=PTmax_mj(i);
```

```
            PT_mj(i)=PTmax_mj(i);
        else PT_mj(i)=PT_mj(i);
        end
end
for i=1:TL
        if PT_jm(i)<=PTmin_jm(i);
            PT_jm(i)=PTmin_jm(i);
        elseif PT_jm(i)>=PTmax_jm(i);
            PT_jm(i)=PTmax_jm(i);
        else PT_jm(i)=PT_jm(i);
        end
end
Tielinepower=PT_mj-(1-flr_jm).*PT_jm;
%% Incremental lambda
P1=P(1:10);P2=P(11:20);P3=P(21:30);P4=P(31:40);
deltalambda1=(PD(1)+Tielinepower(1))-sum(P1);
deltalambda2=(PD(2)+Tielinepower(2))-sum(P2);
deltalambda3=(PD(3)+Tielinepower(3))-sum(P3);
deltalambda4=(PD(4)+Tielinepower(4))-sum(P4);
deltalambda=[deltalambda1 deltalambda2 deltalambda3 deltalambda4]
if abs(deltalambda1)<=epsilon(1) & iter>=N_iter
    break
else  lambda1=lambda1+(deltalambda1*alfa(1))
end
if abs(deltalambda2)<=epsilon(2) & iter>=N_iter
    break
else  lambda2=lambda2+(deltalambda2*alfa(2))
end
if abs(deltalambda3)<=epsilon(3) & iter>=N_iter
    break
else  lambda3=lambda3+(deltalambda3*alfa(3))
end
if abs(deltalambda4)<=epsilon(4) & iter>=N_iter
    break
else  lambda4=lambda4+(deltalambda4*alfa(4))
end
lambda={lambda1,lambda2,lambda3,lambda4}
k=k+1
end
deltalambda=[deltalambda1 deltalambda2 deltalambda3 deltalambda4]
P=P'
Tielinepower
PT_mj
PT_jm
P={P(1:10),P(11:20),P(21:30),P(31:40)}
for i=1:m
fuelcost_area=sum(a{i}.*P{i}.^2+b{i}.*P{i}+c{i});
fuelcost(i,:)=fuelcost_area;
emissioncost_area=sum(d{i}.*P{i}.^2+e{i}.*P{i}+f{i});
emissioncost(i,:)=emissioncost_area;
ceed_area=sum(a{i}.*P{i}.^2+b{i}.*P{i}+c{i}+h{i}.*(d{i}.*P{i}.^2+e{i}.*P{
i}+f{i}));
ceed(i,:)=ceed_area;
end
fuelcost=sum(fuelcost)
emissioncost=sum(emissioncost)
ceed=sum(ceed)
toc
```

Appendix J2: **MATLAB script file: MACEEDP_Parallel_Casestudy1_funct.m**

```
% M-File : MACEEDP_Prallel_Casestudy1_funct.m
% The software is developed for MACEED problem in a Parallel way
% ==================M-File Description================================
% The M-File is used to solve Multiarea CEED problem using various price
penalty factors by Lagrange’s Algorithm
%% The generator active power calculation in the Cluster of computers
%% The group of generators in each area is assigned to the individual
workers
%% case study1 : 4 area power systems , each area having 10 generators,
(4areax10generators=40 generators)
%% Therefore 10generators in each area is assigned to 4 workers in this
case
%% 4 areas each have 10 generators and its real power calculated using
4workers by varying the lab index 1 to 4
function [ P ] = MACEEDP_Prallel_Casestudy1_funct(m,a,h,d,lambda,b,e)
%%
% Generator active power calculation
for i=1:m
  if labindex==i
     E=diag((a{i}+h{i}.*d{i})./lambda{i});
     D=0.5*(1-(b{i}+h{i}.*e{i})./lambda{i})';
     P=E\D;
  end
end
%=======================================================================
```

Appendix J3: **MATLAB script file: MACEEDP_Parallel_Casestudy2.m**

```
% M-File :MACEEDP_Prallel_Casestudy2.m
% The software is developed for MACEED problem in a Parallel way
% ==================M-File Description==============================
% The M-File is used to solve Multiarea CEED problem using various price
penalty factors by Lagrange’s Algorithm
clear all
clc
%% Multi area economic dispatch data
iter=20000;     % No of iteration
m=4; % No of area
n=12; % Total No of Generators in all the area
TL=6;% NO of Tie lines
lambda=[ 100 100 100 100]
lambda1=lambda(1);lambda2=lambda(2);lambda3=lambda(3);lambda4=lambda(4);
lambda={lambda(1),lambda(2),lambda(3),lambda(4)}
PD=[500 410 580 600]; % Area power demand
%PD=[600 450 680 700]; % Area power demand
%PD=[525 435 605 625]; % Area power demand
alfa=[0.01 0.01 0.01 0.01]; %Incremental deltalambda Tolerance value of
each area
epsilon=[0.01 0.01 0.01 0.01]; % Tolerance value of each area
%% Multi area Generator fuel Cost coefficients
a=[0.03546 0.02111 0.01799...
   0.15247 0.02803 0.14834...
   0.10587 0.07505 0.11934...
   0.10587 0.13552 0.08963];
a={a(1:3),a(4:6),a(7:9),a(10:12)}
b=[38.30553 36.32782 38.27041...
   38.53973 40.39655 38.34001...
```

```
   46.15916 43.83562 50.63211...
   46.15916 41.03782 33.56211];
b={b(1:3),b(4:6),b(7:9),b(10:12)}
c=[1243.5311 1658.5696 1356.6592...
   0756.7989 0449.9977 0558.5696...
   0451.3251 0673.0267 0530.7199...
   0851.3251 1038.533  1285.907];
c={c(1:3),c(4:6),c(7:9),c(10:12)}
%% Multi area Emmis on coefficients
d=[0.00683 0.00461 0.00461...
   0.00484 0.00754 0.00661...
   0.00914 0.00533 0.00674...
   0.00728 0.00479 0.00387];
d={d(1:3),d(4:6),d(7:9),d(10:12)}
e=[-0.54551 -0.51160  -0.51160...
   -0.32767 -0.54551  -0.63262...
   -0.43211 -0.61173  -0.49731...
   -0.6821  -0.50660  -0.49340];
e={e(1:3),e(4:6),e(7:9),e(10:12)}
f=[40.26690  42.89553  42.89553...
   33.85932  50.639310 45.83267...
   48.21560  52.45210  41.10420...
   30.36320  25.17650  27.75490];
f={f(1:3),f(4:6),f(7:9),f(10:12)}
%%  Multi area Generator real power limits
Pmin=[35 130 125....
      10 35 125....
      15 30 50....
      15 30 50];
Pmin1={Pmin(1:3),Pmin(4:6),Pmin(7:9),Pmin(10:12)}
Pmax=[210 325 315....
      150 110 215....
      175 215 335....
      175 215 335];
Pmax1={Pmax(1:3),Pmax(4:6),Pmax(7:9),Pmax(10:12)}
%% Intial Tie line values (Assumed)
PT_mj=[10 15 12....
       20 18 29];
PT_jm=[15 18 20....
       14 22 19];
%% Tie line limits
PTmin_mj=[5 5 5....
          5 5 5];
PTmin_jm=[5 5 5....
          5 5 5];
PTmax_mj=[60 50 60....
          60 60 50];
PTmax_jm=[50 60 60....
          60 50 60];
%% Tie line coefficients
q_mj==[0.01     0.021       0.012....
        0.012   0.131       0.011];
q_jm==[0.211    0.012       0.214....
       0.106    0.201       0.101];
%% Tie line Fractional loss rate values
flr_jm=[0.11 0.21 0.14....
        0.16 0.22 0.11];
%% Tie line incremental value
```

```
alpha_mj=[0.000001 0.000001 0.000001....
          0.000001 0.000001 0.000001];
alpha_jm=[0.00001 0.00001 0.00001....
          0.00001 0.00001 0.00001];
%% Transmission loss coefficients
B_area=[ 0.000071  0.00003   0.000025
         0.00003   0.000069  0.000032
         0.000025  0.000032  0.00008
         0.000056  0.000045  0.000015
         0.000023  0.000042  0.000047
         0.000032  0.000023  0.000027
         0.00002   0.000028  0.000053
         0.000086  0.000034  0.000016
         0.000053  0.000016  0.000028
         0.000074  0.00003   0.000025
         0.000049  0.000069  0.000037
         0.000022  0.000032  0.000083];

B_area1={B_area((1:3),:),B_area((4:6),:),B_area((7:9),:),B_area((10:12),:
)}
     for i=1:m

h=(a{i}.*Pmax1{i}.^2+b{i}.*Pmax1{i}+c{i})./(d{i}.*Pmax1{i}.^2+e{i}.*Pmax1
{i}+f{i});
     %
h=(a{i}.*Pmin1{i}.^2+b{i}.*Pmin1{i}+c{i})./(d{i}.*Pmax1{i}.^2+e{i}.*Pmax1
{i}+f{i});

%h=(a{i}.*Pmin1{i}.^2+b{i}.*Pmin1{i}+c{i})./(d{i}.*Pmin1{i}.^2+e{i}.*Pmin
1{i}+f{i});

%h=(a{i}.*Pmax1{i}.^2+b{i}.*Pmax1{i}+c{i})./(d{i}.*Pmin1{i}.^2+e{i}.*Pmin
1{i}+f{i});
     h1(i,:)=h
     end
    h={h1(1:3),h1(4:6),h1(7:9),h1(10:12)}
     % ==============Matlab Parallel multiarea Economic Dispatch
code================================
%Matlabpool open jobmanagerconfig1 %Opening 6 Matlab workers from the
cluster
k=1;
while k<=20000
t0 = clock; % start of the clock on second level calculation
jm = findResource('scheduler','configuration','jobmanagerconfig1')
disp(get(jm))%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% check the status of the
jobmanager
timingStart = tic;
start = tic;
pjob1=createParallelJob(jm,'Configuration','jobmanagerconfig1'); %create
parallel job from the jobmanager
get(pjob1) % Monitor the status of pjob1
%==========How long it takes to create a job=============================
time.jobCreateTime = toc(start)
description.jobCreateTime = 'Job creation time'
%=====================================================================
Tasks = findTask(pjob1)
[pending running finished] = findTask(pjob1)
set(pjob1,'Configuration','jobmanagerconfig1')
```

```
set(pjob1,'MinimumNumberOfWorkers',4);
set(pjob1,'MaximumNumberOfWorkers',4);
set(pjob1,'FileDependencies',{'MACEEDP_Prallel_Casestudy2_funct.m'})
%get(jm)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Notice how jobmanager
is changing
%disp(get(pjob1))
%disp('busy workers')
%disp(get(jm, 'NumberOfbusyWorkers'))
%====================STARTING PARALLEL COMPUTING=======================
%Create task- only one with 4 output arguments  and not input arguments,
%measure how long that takes and measure how long it takes to submit the
%job to the cluster
task1=createTask(pjob1,
@MACEEDP_Prallel_Casestudy2_funct,1,{m,a,h,d,lambda,B_area1,b,e})
get(pjob1,'task')
%==========How long it takes to create a task===========================
time.taskCreateTime = toc(start)
description.taskCreateTime = 'Task creation time'
%=======================================================================
submit(pjob1)
%===========How long it takes to submit a job===========================
time.submitTime = toc(start)
description.submitTime = 'Job submission time'
%=======================================================================
%Once the job has been submitted, we hope all its tasks execute in
%parallel. Measure how long it takes for all the tasks to start and to
run
%to completion.
waitForState(pjob1,'finished')
time.jobWaitTime = toc(start)
description.pjobWaitTime = 'Job wait time'
get(jm)
%Tasks have now completed, so we are again executing a code in the matlab
%client. Measure how long it takes to retrieve all the job results
results=getAllOutputArguments(pjob1)
time.resultsTime = toc(start)
description.resultsTime = 'Results retrieval time'
%=======================================================================
%============Verify that the job ran without errors=====================
errmsgs = get(pjob1.task, {'ErrorMessage'})
nonempty = ~cellfun(@isempty, errmsgs)
celldisp(errmsgs(nonempty))
%======================================================================
%Measure the total time elapsed from creating the job up to the results
time.totalTime = toc(start)
operationtime = etime(clock,t0) %operation time for completion of
parallel computing
%destroy(pjob1)
P=cell2mat(results);
%% Generator real power limits
for i=1:n          % Total Number of generators in all the four area is
n=12
    if P(i)<=Pmin(i)
    P(i)=Pmin(i)
    elseif P(i)>=Pmax(i)
    P(i)=Pmax(i)
    else P(i)=P(i)
    end
```

```
end
%% Tie line power flow
 ePT_mj=q_mj+lambda1;
 PT_mj=(PT_mj-alpha_mj.*ePT_mj);
 ePT_jm=q_jm-(1-flr_jm).*lambda1  ;
 PT_jm=PT_jm-alpha_jm.*ePT_jm;
%% Tie Line limits
for i=1:TL
        if PT_mj(i)<=PTmin_mj(i)
           PT_mj(i)=PTmin_mj(i);
        elseif PT_mj(i)>=PTmax_mj(i);
            PT_mj(i)=PTmax_mj(i);
        else PT_mj(i)=PT_mj(i);
        end
end
for i=1:TL
        if PT_jm(i)<=PTmin_jm(i);
            PT_jm(i)=PTmin_jm(i);
        elseif PT_jm(i)>=PTmax_jm(i);
            PT_jm(i)=PTmax_jm(i);
        else PT_jm(i)=PT_jm(i);
        end
end
Tielinepower=PT_mj-(1-flr_jm).*PT_jm;
%% Transmission line loss
 B_area11=permute(reshape(B_area,3,4,[]),[1 3 2])
 P=reshape(P,3,1,[])
 PL_area=bsxfun(@times,P,reshape(P,1,3,[])).*B_area11
 PL=sum(reshape(PL_area,[],4))
 PL_area1=PL(1);PL_area2=PL(2);PL_area3=PL(3);PL_area4=PL(4);
%% Incremental lambda
P1=P(1:3);P2=P(4:6);P3=P(7:9);P4=P(10:12);
deltalambda1=(PD(1)+PL(1)+Tielinepower(1))-sum(P1);
deltalambda2=(PD(2)+PL(2)+Tielinepower(2))-sum(P2);
deltalambda3=(PD(3)+PL(3)+Tielinepower(3))-sum(P3);
deltalambda4=(PD(4)+PL(4)+Tielinepower(4))-sum(P4);
deltalambda=[deltalambda1 deltalambda2 deltalambda3 deltalambda4]
if abs(deltalambda1)<=epsilon(1) & iter>=k
    break
else  lambda1=lambda1+(deltalambda1*alfa(1))
end
if abs(deltalambda2)<=epsilon(2) & iter>=k
    break
else  lambda2=lambda2+(deltalambda2*alfa(2))
end
if abs(deltalambda3)<=epsilon(3) & iter>=k
    break
else  lambda3=lambda3+(deltalambda3*alfa(3))
end
if abs(deltalambda4)<=epsilon(4) & iter>=k
    break
else  lambda4=lambda4+(deltalambda4*alfa(4))
end
lambda={lambda1,lambda2,lambda3,lambda4}
k=k+1
end
deltalambda=[deltalambda1 deltalambda2 deltalambda3 deltalambda4]
P
```

```
Tielinepower
PT_mj
PT_jm
P={P(1:3),P(4:6),P(7:9),P(10:12)}
for i=1:m
fuelcost_area=sum(a{i}.*P{i}.^2+b{i}.*P{i}+c{i});
fuelcost(i,:)=fuelcost_area;
emissioncost_area=sum(d{i}.*P{i}.^2+e{i}.*P{i}+f{i});
emissioncost(i,:)=emissioncost_area;
ceed_area=sum(a{i}.*P{i}.^2+b{i}.*P{i}+c{i}+h{i}.*(d{i}.*P{i}.^2+e{i}.*P{
i}+f{i}));
ceed(i,:)=ceed_area;
end
fuelcost
emissioncost
ceed
%========================================================================
```

Appendix J4: **MATLAB script file: MACEEDP_Parallel_Casestudy2_funct.m**

```
% M-File : MACEEDP_Prallel_Casestudy2_funct.m
% The software is developed for MACEED problem in a Parallel way
% ==================M-File Description===============================
% The M-File is used to solve Multi-area CEED problem using various price
penalty factors by Lagrange's Algorithm
%% The generator active power calculation in the Cluster of computers
%% The group of generators in each area is assigned to the individual
workers
%% case study2 : 4 area power systems , each area having 3 generators,
(4areax3generators=12 generators)
%% Therefore 3generators in each area is assigned to 4 workers in this
case
%% 4 areas each have 3 generators and its real power calculated using
4workers by varying the lab index 1 to 4
function [ P ] =
MAEDP_Prallel_Casestudy2_funct(m,a,h,d,lambda,B_area1,b,e)
%%
% Generator active power calculation
for i=1:m
  if labindex==i
     E=diag((a{i}+h{i}.*d{i})./lambda{i})+B_area1{i};
     D=0.5*(1-(b{i}+h{i}.*e{i})./lambda{i})';
     P=E\D;
  end
end
%========================================================================
```

APPENDIX K: MATLAB SCRIPT FILE TO GENERATE BAR GRAPH FOR NUMBER OF PUBLICATIONS AND MOST USED ALGORITHMS

Appendix K1: MATLAB script file: Number_of_Publications_Yearwise.m

```
% The M-File is used to generate the bar graph to represent the
number of publications used in thesis versus year (Figure 2.2)
function Number_of_Publications_Yearwise(Number_of_Publications)
Number_of_Publications=[2  1   1   1   1   1   1   1   2   1   2  2
4   2   2   7    4   2   2   3   3   6   4   3   5   8  14  14  16
14   19    11   10   13];
%CREATEFIGURE1(YVECTOR1)
%  YVECTOR1:  bar y vector

%  Auto-generated by MATLAB on 31-May-2013 22:45:31

% Create figure
figure1 = figure;

% Create axes
axes1 = axes('Parent',figure1,'XTickLabel',{},'XTick',[1 2 3 4 5]);
% Uncomment the following line to preserve the X-limits of the axes
xlim(axes1,[0 36]);
% Uncomment the following line to preserve the Y-limits of the axes
ylim(axes1,[0 22]);
box(axes1,'on');
hold(axes1,'all');

% Create bar
bar(Number_of_Publications);

% Create title
title('Number of Publications versus Year','FontSize',14);

% Create xlabel
xlabel('Year','FontSize',12);

% Create ylabel
ylabel(' Number of Publications','FontSize',12);

% Create text
text('Parent',axes1,'String','1958','Rotation',90,'Position',[1 2.1
0]);

% Create text
text('Parent',axes1,'String','1967','Rotation',90,'Position',[2 1.1
0]);

% Create text
text('Parent',axes1,'String','1972','Rotation',90,'Position',[3 1.1
0]);

% Create text
text('Parent',axes1,'String','1977','Rotation',90,'Position',[4 1.1
0]);
```

```
% Create text
text('Parent',axes1,'String','1981','Rotation',90,'Position',[5 1.1
0]);

% Create text
text('Parent',axes1,'String','1983','Rotation',90,'Position',[6 1.1
0]);

% Create text
text('Parent',axes1,'String','1984','Rotation',90,'Position',[7 1.1
0]);

% Create text
text('Parent',axes1,'String','1985','Rotation',90,'Position',[8 1.1
0]);

% Create text
text('Parent',axes1,'String','1988','Rotation',90,'Position',[9 2.1
0]);

% Create text
text('Parent',axes1,'String','1989','Rotation',90,'Position',[10 1.1
0]);

% Create text
text('Parent',axes1,'String','1990','Rotation',90,'Position',[11 2.1
0]);

% Create text
text('Parent',axes1,'String','1991','Rotation',90,'Position',[12 2.1
0]);

% Create text
text('Parent',axes1,'String','1992','Rotation',90,'Position',[13 4.1
0]);

% Create text
text('Parent',axes1,'String','1993','Rotation',90,'Position',[14 2.1
0]);

% Create text
text('Parent',axes1,'String','1994','Rotation',90,'Position',[15 2.1
0]);

% Create text
text('Parent',axes1,'String','1995','Rotation',90,'Position',[16 7.1
0]);

% Create text
text('Parent',axes1,'String','1996','Rotation',90,'Position',[17 4.1
0]);

% Create text
text('Parent',axes1,'String','1997','Rotation',90,'Position',[18 2.1
0]);
```

```
% Create text
text('Parent',axes1,'String','1998','Rotation',90,'Position',[19 2.1
0]);

% Create text
text('Parent',axes1,'String','1999','Rotation',90,'Position',[20 3.1
0]);

% Create text
text('Parent',axes1,'String','2000','Rotation',90,'Position',[21 3.1
0]);

% Create text
text('Parent',axes1,'String','2001','Rotation',90,'Position',[22 6.1
0]);

% Create text
text('Parent',axes1,'String','2002','Rotation',90,'Position',[23 4.1
0]);

% Create text
text('Parent',axes1,'String','2003','Rotation',90,'Position',[24 3.1
0]);

% Create text
text('Parent',axes1,'String','2004','Rotation',90,'Position',[25 5.1
0]);

% Create text
text('Parent',axes1,'String','2005','Rotation',90,'Position',[26 8.1
0]);

% Create text
text('Parent',axes1,'String','2006','Rotation',90,'Position',[27 14.1
0]);

% Create text
text('Parent',axes1,'String','2007','Rotation',90,'Position',[28 14.1
0]);

% Create text
text('Parent',axes1,'String','2008','Rotation',90,'Position',[29 16.1
0]);

% Create text
text('Parent',axes1,'String','2009','Rotation',90,'Position',[30 14.1
0]);

% Create text
text('Parent',axes1,'String','2010','Rotation',90,'Position',[31 19.1
0]);

% Create text
text('Parent',axes1,'String','2011','Rotation',90,'Position',[32 11.1
0]);
```

```
% Create text
text('Parent',axes1,'String','2012','Rotation',90,'Position',[33 10.1
0]);

% Create text
text('Parent',axes1,'String','2013','Rotation',90,'Position',[34 13.1
0]);


=====================================================================
```

Appendix K2: **MATLAB script file: Number_of_Algorithms_used.m**

```
% The M-File is used to generate the bar graph to represent the
number of algorithms used in thesis versus year (Figure 2.3)

function Number_of_Algorithm(Algorithms)
Algorithms=[5 1 1 1 1 1 5 2 4 1 1 10 1 3 4 1 1 1 1 3 1 10 2 2 2 2 51
2 1 1 1 2 1 1 53];

%CREATEFIGURE1(YVECTOR1)
%  YVECTOR1:  bar y vector

%  Auto-generated by MATLAB on 01-Jun-2013 01:37:24

% Create figure
figure1 = figure;

% Create axes
axes1 = axes('Parent',figure1);
% Uncomment the following line to preserve the X-limits of the axes
 xlim(axes1,[0 37]);
% Uncomment the following line to preserve the Y-limits of the axes
ylim(axes1,[0 60]);
box(axes1,'on');
hold(axes1,'all');

% Create bar
bar(Algorithms);

% Create title
title('Number of Publications Vs Algorithm','FontSize',14);

% Create xlabel
xlabel('Algorithm','FontSize',12);

% Create ylabel
ylabel(' Number of Publications','FontSize',12);

% Create text
text('Parent',axes1,'String','ANN','Rotation',90,'Position',[1 5.5
0]);

% Create text
```

```
text('Parent',axes1,'String','BA','Rotation',90,'Position',[2 1.5
0]);

% Create text
text('Parent',axes1,'String','BGO','Rotation',90,'Position',[3 1.5
0]);

% Create text
text('Parent',axes1,'String','CM','Rotation',90,'Position',[4 1.5
0]);

% Create text
text('Parent',axes1,'String','CT','Rotation',90,'Position',[5 1.5
0]);

% Create text
text('Parent',axes1,'String','CUF','Rotation',90,'Position',[6 1.5
0]);

% Create text
text('Parent',axes1,'String','DE','Rotation',90,'Position',[7 5.5
0]);

% Create text
text('Parent',axes1,'String','DP','Rotation',90,'Position',[8 2.5
0]);

% Create text
text('Parent',axes1,'String','DS','Rotation',90,'Position',[9 4.5
0]);

% Create text
text('Parent',axes1,'String','DWDP','Rotation',90,'Position',[10 1.5
0]);

% Create text
text('Parent',axes1,'String','EM','Rotation',90,'Position',[11 1.5
0]);

% Create text
text('Parent',axes1,'String','EP','Rotation',90,'Position',[12 10.5
0]);

% Create text
text('Parent',axes1,'String','ES','Rotation',90,'Position',[13 1.5
0]);

% Create text
text('Parent',axes1,'String','FL','Rotation',90,'Position',[14 3.5
0]);

% Create text
text('Parent',axes1,'String','GA','Rotation',90,'Position',[15 4.5
0]);

% Create text
```

```
text('Parent',axes1,'String','GAMS','Rotation',90,'Position',[16 1.5
0]);

% Create text
text('Parent',axes1,'String','GSA','Rotation',90,'Position',[17 1.5
0]);

% Create text
text('Parent',axes1,'String','GSM','Rotation',90,'Position',[18 1.5
0]);

% Create text
text('Parent',axes1,'String','HA','Rotation',90,'Position',[19 1.5
0]);

% Create text
text('Parent',axes1,'String','HS','Rotation',90,'Position',[20 3.5
0]);

% Create text
text('Parent',axes1,'String','JM','Rotation',90,'Position',[21 1.5
0]);

% Create text
text('Parent',axes1,'String','LA','Rotation',90,'Position',[22 10.5
0]);

% Create text
text('Parent',axes1,'String','LI','Rotation',90,'Position',[23 2.5
0]);

% Create text
text('Parent',axes1,'String','NFM','Rotation',90,'Position',[24 2.5
0]);

% Create text
text('Parent',axes1,'String','NR','Rotation',90,'Position',[25 2.5
0]);

% Create text
text('Parent',axes1,'String','PS','Rotation',90,'Position',[26 2.5
0]);

% Create text
text('Parent',axes1,'String','PSO','Rotation',90,'Position',[27 51.5
0]);

% Create text
text('Parent',axes1,'String','PTDF','Rotation',90,'Position',[28 2.5
0]);

% Create text
text('Parent',axes1,'String','QP','Rotation',90,'Position',[29 1.5
0]);

% Create text
```

```
text('Parent',axes1,'String','RMI','Rotation',90,'Position',[30 1.5
0]);

% Create text
text('Parent',axes1,'String','SAMF','Rotation',90,'Position',[31 1.5
0]);

% Create text
text('Parent',axes1,'String','SDP','Rotation',90,'Position',[32 2.5
0]);

% Create text
text('Parent',axes1,'String','TS','Rotation',90,'Position',[33 1.5
0]);

% Create text
text('Parent',axes1,'String','UCA','Rotation',90,'Position',[34 1.5
0]);

% Create text
text('Parent',axes1,'String','others','Rotation',90,'Position',[35
53.5 0]);

%=====================================================================
```

APPENDIX L: PHOTOGRAPH OF THE CLUSTER COMPUTER LABORATORY

The Photograph shows the Cape Peninsula University of Technology Research Center "Real Time Distributed Systems" (RTDS) Cluster Computer Laboratory, which has 32 workers installed with Matlab Distributed Computing Engine (MDCE). The workers are connected in a network using Ethernet communication.