UNIVERSITY OF NAIROBI

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING

PROJECT: SECURITY CONSTRAINED ECONOMIC DISPATCH USING IMPROVED OUT-OF-KILTER ALGORITHM (IOKA)

PROJECT INDEX: 57

NAME: KENYATTA BRIAN WILLIAM

REG. NO: F17/1404/2010

SUPERVISOR: PROF. N. O. ABUNGU

EXAMINER: DR.C.W.WEKESA

A project report submitted to the Department of Electrical and Information Engineering in partial fulfilment of the requirements for the award of the Degree of Bachelor of Science in Electrical and Electronic Engineering of the University of Nairobi

Submitted on: 24th April 2015

DECLARATION OF ORIGINALITY

NAME OF STUDENT: Kenyatta Brian William

REGISTRATION NUMBER: F17/1404/2010

COLLEGE: Architecture and Engineering

FACULTY/ SCHOOL/ INSTITUTE: Engineering

DEPARTMENT: Electrical and Information Engineering

COURSE NAME: Bachelor of Science in Electrical & Electronic Engineering

TITLE OF WORK: Security Constrained Economic Dispatch Using Improved Out-of-Kilter Algorithm (IOKA)

I understand what plagiarism is and I am aware of the university policy in this regard.

I declare that this final year project report is my original work and has not been submitted elsewhere for examination, award of a degree or publication. Where other people's work or my own work has been used, this has properly been acknowledged and referenced in accordance with the University of Nairobi's requirements.

I have not sought or used the services of any professional agencies to produce this work.

I have not allowed, and shall not allow anyone to copy my work with the intention of passing it off as his/her own work.

I understand that any false claim in respect of this work shall result in disciplinary action, in accordance with University anti-plagiarism policy.

Signature:

....

Date:

....

CERTIFICATION

This report has been submitted to the Department of Electrical and Information Engineering University of Nairobi with my approval as supervisor:

Prof. Nicodemus Abungu Odero

Date:...

DEDICATION

I dedicate this project to my family for their undying and continued support and encouragement.

ACKNOWLEDGEMENTS

Firstly I want to sincerely thank my family for their support and understanding through the hard moments of this study and the continued help and encouragement they have afforded me.

In relation with University of Nairobi I am really thankful to my supervisor, Prof. Nicodemus Abungu Odero for sharing his knowledge, help in the execution and all the parts entailed in the completion of this project.

Sincere thanks to Lina Doktur, Ian Atsobwa and classmates for the encouragement and love.

I would also like to thank the almighty for the gift of life and the strength in reaching this far.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ED	Economic Dispatch
ED	classical Economic Dispatch
SCED	Security Constrained Economic Dispatch
GA	Genetic Algorithm
LP	Linear Programming
NLP	Non-Linear Programming
GRG	Generalized Reduced Gradient
QP	Quadratic Programming
NFP	Network flow programming
OPF	Optimal Power Flow
ACO	Ant Colony Optimization
PSO	Particle Swarm Optimization
TS	Tabu Search
FIFO	first-in-first-out
EA	Evolutionary Algorithm
CPU	Central Processing Unit
SA	Simulated Annealing
ONN	Optimization neural network
SCED	Security Constrained Economic Dispatch
ELD	Economic Load Dispatch
MATLAB	Matrix Laboratory
SCADA	Supervisory Control and Data Acquisition
EMS	Energy Management Systems
IOKA	Improved Out-of-Kilter Algorithm
OKA	Out-of-Kilter Algorithm
HVDC	High Voltage Direct Current
MW	Megawatts
IEEE	Institute of Electrical and Electronic Engineering

ABSTRACT

This day and age, a vast and array of equipments are electricity consuming devices, from our means of transport, pumps that enable water supply, cellular communication devices to lighting for our homes. Therefore, we require reliable and efficient power supply at reasonable cost. Generation is to be done while ensuring that production of energy is at its lowest cost in order to reliably serve consumers. In addition to ensuring reduced costs for both consumers and producers, power system operators are also responsible for ensuring that power supply match load demand and for keeping the system under safe operational limits, such as, ensuring that no single fault may cause the power system to fail. Hence security constrained Economic dispatch (SCED) problem is of prime relevance.

SCED is an optimization process that takes account of these factors such as, the varying load demands, the varying cost of different types of generation units, and the unexpected conditions of the transmission network that affect which generation units can be used to serve load reliability. These factors are bared in mind while selecting the generation units to dispatch in order to deliver a reliable supply of the electrical power at the lowest cost possible under given conditions.

SCED problem has been successfully performed with conventional methods such as linear programming [LP] & Quadratic programming, intelligent search methods such as particle swarm optimization and genetic algorithm. In this project, the improved out-of-kilter algorithm has been realized to solve the SCED problem.

The realized IOKA algorithm has been implemented on the IEEE 30-bus network. The optimal real power output for a system total load demand of 283.4MW was found to be; Total Generation = 293.04 MW. The generation for each of the six generation units; P_{G1}=178.347 MW, P_{G2}=49.01 MW, P_{G5}=20.09 MW, P_{G8}=21.99 MW, P_{G11}=11.84 MW and P_{G13}=10.92 MW. The obtained real power losses; 9.64 MW and the total optimal generation cost obtained as 802.34 $/hr. The IOKA algorithm was able to achieve similar results in cost as those obtained from LP = 802.4 $/hr and OKA = 802.51 $/hr. The cost of SCED and classical ED was also compared for various load demands; SCED had higher costs of generation which was more pronounced in higher load demands.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1. DEFINITION OF TERMS

1.1.1. Economic Load Dispatch

ELD is the short term determination of the optimal output of a number of electricity generation facilities, to meet the system demand, at the lowest possible fuel cost, while serving power to the public in a robust but reliable manner [1].

Optimal efficiency in power generation reduces the cost per kilowatt hour passed to consumers from the power producers and also the cost of operation of the power companies despite the fluctuating prices of fuel, labour, supplies and maintenance. Economic dispatch deals with the problem of minimum cost of production of energy. It coordinates the production costs of all the power plants operating in the system. Variation of load demand in per hour and per day basis makes it paramount to perform coordination control of power plant outputs in order to keep the system as secure as possible [2].

1.1.2. System security

A power system must be capable of withstanding the loss of some or several components e.g. transmission lines, transformers and generators while still staying in operation. Power system security is the ability to maintain the flow of electricity from the generators to the consumers, especially under disturbed conditions. Since the disturbance can be small or widespread, the security criteria should be able to ensure sufficient security margins. The measure of power system security is amount, duration and frequency of power outages from consumers. The need for power system security led to interconnection of power generation units to form a large transmission network that could provide alternate paths in case of transmission outages on the power grid. This leads to the disadvantage of rare disturbance affecting a large geographical area as was apparent in the widespread power blackout in the USA in 2003. However, an overlay of computers and communications on the power networks has enabled more secure operations and control [3] [4].

Therefore there are various engineering tools that are used in the energy control centre to perform system monitoring, contingency analysis, preventive analysis and corrective analysis.

1.1.3. Security constraints

1.1.3.1 Equality constraints

The equality constraint g(x) of the ELD problem is represented by the power balance constraint, where the total power generation must cover the total power demand and the power loss. This implies solving the load flow problem, which has equality constraints on an active and reactive power at each bus as follows [3]:

$$P_i = P_{gi} - P_{di} = \sum_{j=1}^{N}[V_j V_i[G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}]] \quad (1)$$

$$Q_i = Q_{gi} - Q_{di} = \sum_{j=1}^{N}[V_j V_i[G_{ij} \sin \theta_{ij} + B_{ij} cos\theta_{ij}]]$$

Where: $i = 1,2,\dots,n$ and $\theta ij = \theta i - \theta j$; Pi, Qi: injected active and reactive power at bus I; Pdi, Qdi: active and reactive power demand at bus i; $Vi, \theta i$: bus voltage magnitude and angle at bus i; Gij, Bij: conductance and susceptance of the (i,j) element in the admittance matrix.

1.1.3.2. Inequality constraints

The inequality constraints h(x) reflect the limits on physical devices in the power system as well as the limits created to ensure system security:

a) Upper and lower bounds on the active and reactive generations

$$\mathrm{P_{gi-min}} \leq \mathrm{P_{gi}} \leq \mathrm{P_{gi-max}} \text{ for } \mathrm{i} = 1, \dots, \mathrm{N} \quad (2)$$

$$\mathrm{Q_{gi-min}} \leq \mathrm{Q_{gi}} \leq \mathrm{Q_{gi-max}} \text{ for } \mathrm{i} = 1, \dots, \mathrm{N} \quad (3)$$

b) Upper and lower bounds on the tap ratio (t) and phase shifting (α) of variable transformers:

$$\mathrm{t_{ijmin}} \leq \mathrm{t_{ij}} \leq \mathrm{t_{ijmax}} \quad \text{for } \mathrm{i} = 1, \dots, \mathrm{N} \quad (4)$$

$$\alpha_{\mathrm{ijmin}} \leq \alpha_{\mathrm{ij}} \leq \alpha_{\mathrm{ijmax}} \quad \text{for } \mathrm{i} = 1, \dots, \mathrm{N} \quad (5)$$

c) Upper limit on the active power flow (P_{ij}) of line i-j:

$$|Pij| \leq Pijmax \tag{6}$$

Where [55]:

$$P_{ij} = |-G_{ij}V_i^2 + G_{ij}V_iV_j cos(\theta_i - \theta_j) + B_{ij}V_iV_j sin(\theta_i - \theta_j)| \tag{7}$$

d) Upper and lower bounds on the bus voltage magnitude:

$$Vi\,min \leq Vi \leq Vi\,max \tag{8}$$

1.1.4. Security Constrained Economic Dispatch

Security constrained Economic dispatch is the operation of generation facilities to produce energy at the lowest cost to reliably serve consumers, recognizing any operational limits of generation and transmission facilities [4]. This definition describes the basic way utilities dispatch their own and purchased resources to meet electrical load.

The various challenges faced in supplying electricity include,

a) The production must be simultaneous with demand, and demand varies greatly over cause of day, weak and season.
b) The cost of different types of generating units varies greatly.
c) Expected and unexpected conditions on the transmission network also affect which generating units can be used to serve the load reliability.

These factors are bared in mind while selecting the generating units to be dispatched in order to deliver reliable supply of power. This is done while keeping within the stipulated constraints (equality and inequality constraints) of the generation units and transmission facilities.

1.2. SURVEY OF EARLIER WORKS

There are three groups of method classification that have been used to solve the various optimization problems.

- Conventional methods
- Intelligence search methods
- Hybrid methods

1.2.1. CONVENTIONAL METHODS

1.2.1.1. GENETIC ALGORITHM (GA)

Genetic Algorithm is adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetics as observed in natural systems and populations of living beings. In nature, each species is confronted by a challenging environment and should adapt itself for the maximum likelihood of survival. As time proceeds, the species with improved characteristics survives. In fact, the so called fittest type is survived [4]. This type of phenomenon which happens in nature is the basis of the stochastic search technique of the GA. Hence, it differs from many other approaches in that it acts on a population of solutions applying competition and selection tools. Genetic algorithms can be seen as procedures that mimic a simplistic mechanism of biological evolution [5].

GA provides a solution to a problem by working with a population of individuals each representing a possible solution. Each possible solution is termed a chromosome [3]. The objective function is calculated for this chromosome as the problem fitness function. After setting an initial population, selecting a chromosome and calculating its fitness, a next population is generated. Initial chromosomes are called parents and the regenerated chromosomes are called offspring. The regeneration results in chromosomes with better fitness values. The algorithm proceeds until no further improvement is achieved in the fitness function. There are multiple ways to encode elements of solutions including binary, value, and tree encodings. Selection, crossover and mutation are the three main GA operators and are described next [6].

• Selection. Based on the chromosome structure defined, a population of chromosomes is initially generated, either, randomly or intelligently. 30–100 chromosomes may be considered. Then, we may select two chromosomes as parents for further process. The fitness value is used as the criterion for parent's selection.

• Crossover. Crossover combines elements of solutions in the current generation to create a member of the next generation. Once parents are selected, we should generate new strings; offspring, through two types of operators. The so called crossover works on the principle of interchanging the values after a specific position. This type of regeneration is done randomly at various positions. As a result, a new population of chromosomes is generated in which, again, the selection process may be restarted.

• Mutation. Mutation systematically changes elements of a solution from the current generation in order to create a member of the next generation. An inherent drawback of the crossover operator is the fact that at some particular position, the value of the gene may not change at all. To avoid this problem, the mutation operator tries to alter the value of a gene, randomly from 1 to 0 and vice versa. We should mention, however, that this is done quite infrequently.

Crossover and mutation accomplish exploration of the search space by creating diversity in the members of the next generation.

Their advantage lies in the ease of coding them and their inherent parallelism. The use of genotypes instead of phenotypes to travel in the search space makes them less likely to get stuck in local minima. The GA only needs to evaluate the objective function (fitness) to guide its search. There is no requirement for derivatives or other auxiliary knowledge. Hence, there is no need for computation of derivatives or other auxiliary functions [7].

There are, however, certain drawbacks to them. Genetic algorithms require very intensive computation time and hence they are slow. Genetic algorithms also suffer from deception. The term deception describes problems that are misleading for genetic algorithms (GAs). Well-known examples of GA-deceptive problems are *n*-bit trap functions. These functions are characterized by (I) fix-points that correspond to sub-optimal solutions and that have large basins of attraction, and (ii) fix-points with relatively small basins of attraction that correspond to optimal solutions. Therefore, for these problems a GA will—in most cases—not find an optimal solution [8]. Also, finding an appropriate crossover operator turns out to be a difficult task, while using some "general purpose" crossover operators often leads to poor performance. Another problem is the existence of *genetic drift*, that is, a loss of population diversity due to the finite population size, and, as a result, a premature convergence to sub-optimal solutions [9].

1.2.1.2. LINEAR PROGRAMMING (LP)

If the objective and constraint functions are linear and the variables are constrained to be positive, their solution can be readily achieved by using LP [10]. Nonlinear power system optimization problems may also be linearized, so that objective function and constraints of power system optimization have linear forms and solved by linear programming (LP). The general optimization problem assumes the form

Minimize $f(\boldsymbol{x}) = n_i = 1\ \alpha_i x_i$

Subject to: $aj\ (\boldsymbol{x}) = n_i = 1 \quad \beta_{\ ij} x_i - \mu_j = 0\ for\ j = 1, 2 \ldots P$

$Cj\ (\boldsymbol{x}) = n_i = 1 \quad \gamma_{\ ij} x_i - v_j \geq 0\ for\ j = 1, 2 \ldots q\ xi \geq 0\ for\ i = 1, 2 \ldots n$

Whereα_i, $\beta_{\ ij}$,$\gamma_{\ ij}$, μ_j and v_j *are* constants [11]. For example,

Minimize $f(\mathbf{x}) = -2x1 + 4x2 + 7x3 + x4 + 5x5$

Subject to: $a1(\boldsymbol{x}) = -x1 + x2 + 2x3 + x4 + 2x5 - 7 = 0$

$a2(\boldsymbol{x}) = -x1 + 2x2 + 3x3 + x4 + x5 - 6 = 0$

$a3(\boldsymbol{x}) = -x1 + x2 + x3 + 2x4 + x5 - 4 = 0$

$xi \geq 0\ for\ i = 1, 2 \ldots 5$

The LP approach has several advantages. First, it is reliable, especially regarding convergence properties. Second, it can quickly identify infeasibility. Third, it accommodates a large variety of power system operating limits, including the very important contingency constraints. The disadvantages of LP - based techniques are inaccurate evaluation of system losses and insufficient ability to find an exact solution compared with an accurate nonlinear power system model [3].

1.2.1.3. NONLINEAR PROGRAMMING (NLP)

The nonlinear programming optimization algorithm deals with problems involving nonlinear objective and constraint functions. Power system operation problems are nonlinear in nature. To solve a nonlinear programming problem, the first step in this method is to choose a search direction in the iterative procedure, which is determined by the first partial derivatives of the equations (the reduced gradient). Therefore, these methods are referred to as first - order methods, such as the generalized reduced gradient (GRG) method.
A nonlinear optimization problem assumes the form:

Minimize $f = f(\boldsymbol{x})$
Subject to: $\boldsymbol{x} \in R$
Where $f(\mathbf{x})$ is a real-valued function and $R \subset En$ is the feasible region.
There are numerous algorithms that can be used for the solution of nonlinear programming problems ranging from some simple to some highly complex algorithms.
The two most fundamental common properties of nonlinear programming algorithms are

1. They are iterative algorithms.
2. They are descent algorithms.

An algorithm is *iterative* if the solution is obtained by calculating a series of points in sequence, starting with an initial estimate of the solution. On the other hand, an algorithm is a *descent* algorithm if each new point generated by the algorithm yields a reduced value of some function, possibly the objective function [10].
NLP - based methods have higher accuracy than LP - based approaches, and also have global convergence, which means that the convergence can be guaranteed independent of the starting point, but a slow convergent rate may occur because of zigzagging in the search direction [3]. Generally, nonlinear programming based procedures have many drawbacks such as insecure convergence properties and algorithmic complexity [7].

1.2.1.4. QUADRATIC PROGRAMMING (QP)

The quadratic programming technique is a special form of nonlinear programming whose objective function is quadratic with linear constraints. If the optimization problem assumes the form

Minimize $f(\boldsymbol{x}) = \alpha 0 + \boldsymbol{\gamma} T\, \boldsymbol{x} + \boldsymbol{x} T \boldsymbol{Q}\, \boldsymbol{x}$

Subject to: $\boldsymbol{\alpha} T\, \boldsymbol{x} \geq \boldsymbol{\beta}$

Where

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha 11 & \cdots & \alpha 1q \\ \vdots & \ddots & \vdots \\ \alpha n1 & \cdots & \alpha nq \end{bmatrix}$$

$\boldsymbol{\beta} T = [\beta 1\, \beta 2\, \cdots\, \beta q\,]$

$\boldsymbol{\gamma} T = [\gamma 1\, \gamma 2 \ldots \gamma n\,]$

Q is a positive definite or semi definite symmetric square matrix, and the constraints are linear and the objective function quadratic. Such an optimization problem is said to be a quadratic programming (QP) problem [10]. A typical example of this type of problem is as follows:

Minimize $f(\mathbf{x}) = \frac{1}{2}x_1^2 + \frac{1}{2}x_2^2 + x_1 + 2x_2$

Subject to: $c1(\boldsymbol{x}) = 6 - 2x1 - 3x2 \geq 0$

$c2(\boldsymbol{x}) = 5 - x1 - 4x2 \geq 0$

$c3(\boldsymbol{x}) = x1 \geq 0$

$c4(\boldsymbol{x}) = x2 \geq 0$

Quadratic programming has higher accuracy than LP – based approaches. Especially, the most often - used objective function in power system optimization is the generator cost function, which generally is a quadratic. Thus there is no simplification for such objective function for a power system optimization problem solved by QP [3]. Quadratic programming based techniques have some disadvantages associated with the piecewise quadratic cost approximation [7].

1.2.1.5. NEWTON METHOD

The Newton method is a second-order method as it requires the computation of the second - order partial derivatives of the power flow equations and other constraints. The necessary conditions of optimality commonly referred to as the Kuhn–Tucker conditions are obtained [3]. Newton-based techniques have a drawback of convergence characteristics that are sensitive to the initial conditions and they may even fail to converge due to the inappropriate initial conditions [7]. The convergence property is slow, away from the solution, and fast, close to the solution. The main disadvantage of the Newton method is that the second derivatives of the function is required, these exact formulas may be unavailable or difficult to obtain [12]. However, Newton's method is favoured for its quadratic convergence properties [3].

1.2.1.6. NETWORK FLOW PROGRAMMING (NFP)

Network flow programming (NFP) is special linear programming. The early applications of NFP were mainly on a linear model. Recently, nonlinear convex network flow programming has been used in power systems' optimization problems [3].

The general network flow problem can be illustrated by considering figure 1.1. We have a number of *sources* of material and a number of *sinks* (or demand points) for material. Normally, we have each source having an upper bound on the amount of material it can supply and each demand point having an accompanying number indicating the amount of material it needs.

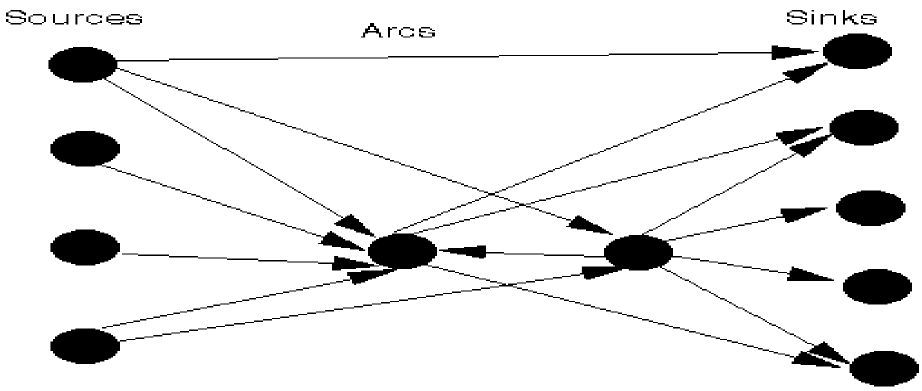

Figure 1.1: Example of a Network

Between the sources and the sinks are intermediate *nodes* through which material can be shipped (flows) to other intermediate nodes or to the sinks. There are also *arcs* (essentially directed from the sources to the sinks), which have associated with each of the arc 1) an upper limit (or *capacity*) on the amount of material which can flow down the arc; and 2) a *cost* per unit of material sent down the arc.

Therefore, the problem, termed the *minimum cost network flow problem,* is one of deciding how to supply the sinks from the sources at minimum cost. Ford and Fulkerson developed an algorithm called the out-of-kilter algorithm for this problem in the early 1960's and this original algorithm has been revised and improved since then.

NFP - based algorithms have the features of fast speed and simple calculation. However, in solving such problems it is the number of arcs which (essentially) determines the solution time. These methods are efficient for solving simplified OPF problems such as security - constrained economic dispatch. This method is reviewed in details in chapter 2 of this project.

1.2.2. INTELLIGENCE SEARCH METHODS

These algorithms are Metaheuristics and are often inspired by natural processes. Artificial Intelligence techniques, unlike strict mathematical methods, have the apparent ability to adapt to nonlinearities and discontinuities commonly found in power systems [7]. Developing solutions with these tools offers two major advantages [13]:

1. Development time is much shorter than when using more traditional approaches.
2. The systems are very robust, being relatively insensitive to noisy and/or missing data.

They include The Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Simulated Annealing, Evolutionary Computation and Tabu Search (TS) methods.

1.2.2.1. ANT COLONY OPTIMIZATION

Ant algorithms were inspired by the observation of real ant colonies hence most of the ideas of ACO stem from real ants. In particular, the use of: (a) a colony of cooperating individuals - ant algorithms are composed of a population, or colony, that cooperate to find a good “solution” to the task under consideration, (b) an (artificial) pheromone trail for local stigmergetic communication - While real ants deposit on the world’s state they visit a chemical substance, the pheromone, artificial ants change some numeric information locally stored in the problem's state they visit. This information takes into account the ant's current history or performance and can be read or written by any ant accessing the state, (c) a sequence of local moves to find shortest paths - Artificial and real ants share a common task: to find a shortest (minimum cost) path joining an origin (nest) to destination (food) sites, and (d) a stochastic decision policy using local information and no look ahead - The policy is a function of both the a priori information represented by the problem specifications (equivalent to the terrain's structure for real ants), and of the local modifications in the environment (pheromone trails) induced by past ants.

While walking from food sources to the nest and vice versa, ants deposit on the ground a substance called *pheromone*, forming in this way a pheromone trail. Ants can smell pheromone, and when choosing their way, they tend to choose, in probability, paths marked by strong pheromone concentrations. The pheromone trail allows the ants to find their way back to the food source (or to the nest). Also, it can be used by other ants to find the location of the food sources found by their nest mates. This pheromone trail following behaviour can give rise, once employed by a colony of ants, to the emergence of shortest paths. That is, when more paths are available from the nest to a food source, a colony of ants may be able to exploit the pheromone trails left by the individual ants to discover the shortest path from the nest to the food source and back [14].

It is clear that what is going on in the above-described process is a kind of distributed optimization mechanism to which each single ant gives only a very small contribution. Though a single ant is in principle capable of building a solution (i.e., of finding a path between nest and food reservoir), it is only the ensemble of ants, that is, the ant colony, that presents the "shortest path-finding" behaviour.

Correspondingly, artificial ants simulate pheromone laying by modifying appropriate "pheromone variables" associated with problem states they visit while building solutions to the optimization problem to which they have been applied.

It has the disadvantage of premature convergence (*stagnation*), that is, the situation in which some not very good individual takes over the population just because of a local optimum impeding further exploration of the search space. Pheromone trail evaporation and stochastic state transitions are the needed complements to alienate such drawbacks. Pheromone evaporation allows the ant colony slowly to forget its past history so that it can direct its search toward new directions without being over-constrained by past decisions [15]. Making pheromone update a function of the generated solution quality helps in directing future ants more strongly toward better solutions [16]. Another disadvantage is its convergence speed is slow because of poor performance on the early path. An advantage of the ACO is its ability of parallel processing and global searching.

1.2.2.2. PARTICLE SWARM OPTIMIZATION

It is a stochastic search technique that has been developed based on behaviour of social animals which live and move in groups such as fish and birds. Birds and fish usually move in groups at a certain speed and position. Their design of movement is dependent on their experience as well as experience of others in the group [17].

The PSO algorithm exploits a population of individuals to probe promising regions of search space. The population is called a swarm and the individuals are called particles or agents. Traditionally, PSO has no crossover between individuals and has no mutation, and particles are never substituted by other individuals during the run. Instead, the PSO refines its search by attracting the particles to positions with good solutions. Each particle remembers its own best position found so far in the exploration. This position is called the personal best. Additionally, among these personal best, there is only one particle that has the best fitness, called the global best. Each particle moves with an adaptable velocity within the regions of decision space and retains a memory of the best position it ever encountered. The best position ever attained by each particle of the swarm is communicated to all other particles. The conventional PSO assumes an n - dimensional search space, where n is the number of decision variables in the optimization problem, and a swarm consisting of N particles. In PSO, a number of particles form a swarm that evolves or flies throughout the problem hyperspace to search for optimal or near - optimal solution. The coordinates of each particle represent a possible solution with two vectors associated with it, the position X and velocity V vectors. During their search, particles interact with each others in a certain way to optimize their search experience [3].

It has the advantage of a superior convergence time hence is fast in solving complex problems effectively. [9]PSO can be applied to non linear and non-continuous optimization problems with continuous variables [17]. It has the flexibility to adapt and enhance both global and local exploration abilities. Therefore, when solving problems with several local optimal solutions, there is high possibility that PSO will explore more local optimal solutions with the potential of global optimal solution after convergence [3]. It is a stochastic search technique with reduced memory requirement, computationally effective and easier to implement than other artificial intelligence techniques as it is easily programmed and modified with basic mathematical and logic operations. It requires less parameter tuning [3].

1.2.2.3. TABU SEARCH

Tabu Search is an iterative search algorithm, characterized by the use of a flexible memory. Tabu means forbidden to search or to consider. Unlike other combinatorial approaches which are developed by physical phenomena, TS is not related to physical phenomena. It is inspired by the clever management of memory structures. The main component of this algorithm is memory structures, in order to have a trace of evolution of the search, and strategy for using the memory information in the best possible way [4].

The fundamental memory structure is a so called Tabu list, which stores attributes characterizing solutions that should not be considered again for a certain length of time. Usually a first-in-first-out (FIFO) strategy is applied to the list. Old attributes are deleted as new attributes are inserted. The neighbourhood, from which the next solution/move is to be selected, is modified by classifying some moves as tabu, others as desirable. For a complete iteration, a neighbourhood structure is defined and a move is then made to the best configuration. To escape from local optimum points, some transitions to the configurations with higher costs are also allowed. Two extra parameters are often used: aspiration and diversification. Aspiration is used when all the neighbouring states of the current state are also included in the tabu list [18]. In that case, the tabu obstacle is overridden by selecting a new state. . Most commonly, the aspiration criterion drops the tabu status of moves leading to a better solution than the best solution visited so far. Diversification adds randomness to this otherwise deterministic search. If the tabu search does not converge, the search is reset randomly [7]. These help avoiding trapping in local optimum points. Intensification strategies are also used, intensification strategies are intended to explore more carefully promising regions of the search space either by recovering elite solutions (i.e., the best solutions obtained so far) or attributes of these solutions [19].

The steps involved in a TS optimization algorithm may be summarized as (a) Generate an initial solution, (b) Select move, (c) Update the solution. The next solution is chosen from the list of neighbours which is either considered as desired (aspirant) or not tabu and for which the objective function is optimum. The process is repeated based on any stopping rule proposed [4].

Tabu search has the advantage of not using hill-climbing strategies. Its performance can also be enhanced by branch-and-bound techniques. However, a solution space must be generated. Hence, tabu search requires knowledge of the entire operation at a more detailed level [11, 12].

1.2.2.4. EVOLUTIONARY ALGORITHMS (EA)

Evolutionary computation has become a standard term to indicate problem-solving techniques which use design principles inspired from models of the natural evolution of species. There are three main algorithmic developments in Evolutionary computation: evolution strategies, evolutionary programming, genetic programming and genetic algorithms. All employ a population-based algorithm that use operators inspired by population genetics to explore the search space, these genetic operators include reproduction, mutation, and recombination or crossover operator. The population of individuals each represents a solution to the problem under consideration.

The reproduction operator refers to the process of selecting the individuals that will survive and be part of the next generation. This operator typically uses a bias toward good-quality individuals. The better the objective function value of an individual, the higher the probability that the individual will be selected and therefore be made part of the next generation. The recombination operator combines parts of two or more individuals and generates new individuals, also called offspring. The mutation operator is a unary operator that introduces random modifications to one individual [19]. These algorithms simulate the principle of evolution (a two-step process of variation and selection), and maintain a population of potential solutions (individuals) through repeated application of these evolutionary operators of reproduction mutation and crossover. They yield individuals with successively improved fitness, and converge; it is hoped, to the fittest individuals representing the optimum solutions.

It has the advantage of converging to the global optimum solution. Evolutionary algorithms are robust and powerful global optimization techniques for solving large-scale problems that have many local optima [7].

Disadvantages

Since EAs require all information to be included in the fitness function, it is very difficult to consider all OPF constraints. It is therefore used to solve simplified OPF problems such as the classic economic dispatch and security - constrained economic power dispatch [3]. They require high CPU times, and they are very poor in terms of convergence performance.

1.2.2.5. SIMULATED ANNEALING (SA)

Simulated annealing is a stochastic search method inspired by an analogy between the physical annealing of solids (crystals) and combinatorial optimization problems. By making an analogy between the annealing process (which is the natural process of cooling a molten material; from a high temperature [4]) and the optimization problem, a large class of combinatorial optimization problems can be solved following the same procedure of transition from an equilibrium state to another, reaching the minimum energy level of the system. This analogy can be set as follows [7].

- Solutions in the combinatorial optimization problem are equivalent to states (configurations) of the physical system.
- The cost of a solution is equivalent to the energy of a state.
- A control parameter is introduced to play the role of the temperature in the annealing process.

The process involves first melting a solid and then cooling it very slowly, letting it spend a long time at low temperatures, to obtain a perfect lattice structure corresponding to a minimum energy state. At each step the temperature is maintained constant for a period of time sufficient for the solid to reach thermal equilibrium. At equilibrium, the solid could have many configurations, each corresponding to different spins of the electrons and to a specific energy level. At equilibrium the probability of a given configuration is given by the Boltzman distribution. SA transfers this process to local search algorithms for combinatorial optimization problems. It does so by associating the set of solutions of the problem attacked with the states of the physical system, the objective function with the physical energy of the solid, and the optimal solutions with the minimum energy states [4]. In simulated annealing [20], (a) an acceptance criterion is defined and only those randomly generated moves that satisfy the criteria are executed, and (b) the search is usually performed in the space of the solutions.

The salient features of the SA method may be summarized as follows:- It could find a high-quality solution that does not strongly depend on the choice of the initial solution, it does not need a complicated mathematical model of the problem under study, it can start with any given solution and try to improve it (This feature could be utilized to improve a solution obtained from other suboptimal or heuristic methods), it has been theoretically proved to converge to the optimum solution and it does not need large computer memory [7]. However, it has its disadvantages too. Like GAs it is very slow; its efficiency is dependent on the nature of the surface it is trying to optimize and it must be adapted to specific problems. The availability of supercomputing resources, however, mitigates these drawbacks and makes simulated annealing a good candidate [7].

1.2.2.6. NEURAL NETWORK

Optimization neural network (ONN) changes the solution of an optimization problem into an equilibrium point (or equilibrium state) of nonlinear dynamic system, and changes the optimal criterion into energy functions for dynamic systems. Because of its parallel computational structure and the evolution of dynamics, the ONN approach appears superior to traditional optimization methods [3].

1.2.3. HYBRID METHODS

These are a combination of two or more methods. A big challenge in developing global optimization approaches is to compromise the contradictory requirements, including accuracy, robustness and computation time. It is difficult to meet all these requirements by concentrating on a sole meta-heuristic. In recent years, there has been an up-growing interest in hybridization of different meta-heuristics to provide more efficient algorithms [21].

Examples include:-

a) ACO and GA - combines Genetic Algorithm and Ant colony algorithms. Genetic Algorithm is added to Ant Colony Algorithm's every generation in the proposed algorithm. Making use of Genetic Algorithm's advantage of whole quick convergence, Ant Colony Algorithm's convergence speed is quickened. Genetic Algorithm's mutation mechanism improves the ability of Ant Colony Algorithm to avoid being trapped in a local optimal [16].

b) ACO and TS – combines Tabu Search and Ant Colony algorithms. The new algorithm incorporates the concepts of promising list, tabu list and tabu balls from TS into the framework of ACO. This enables the resultant algorithm to avoid bad regions and to be guided toward the areas more likely to contain the global minimum. New strategies are proposed to dynamically tune the radius of the tabu balls during the execution and also to handle the variable correlations. The promising list is also used to update the pheromone distribution over the search space. The parameters of the new method are tuned based on the results obtained for a set of standard test functions [21].

c) SA and TS – combines Simulated Annealing and Tabu Search algorithms. The proposed algorithm may be described as an SA algorithm with the TS algorithm used as a filter to reject the repeated trial solutions from being tested by the SA algorithm. The TS method is implemented as a pre-processor step in the SA algorithm to test a set of neighbours to the current solution. The trial solution that satisfies the tabu test is accepted. This accepted trial solution is then accepted or rejected according to the SA test. The main idea in the proposed algorithm is to use the TS algorithm to prevent the repeated solutions from being accepted by the SA. This saves time and improves the quality of the obtained solution [7].

d) GA, SA and TS – based on integrating the Genetic Algorithm, Simulated Annealing, and Tabu Search methods. The core of the proposed algorithm is based on the GA. The TS is used to generate new population members in the reproduction phase of the GA. Moreover, the SA method is adopted to improve the convergence of the GA by testing the population members of the GA after each generation. The SA test allows the acceptance of any solution at the beginning of the search, and only good solutions will have a higher probability of acceptance as the generation number increases. The effect of using the SA is to accelerate the convergence of the GA and also increase the fine-tuning capability of the GA when approaching a local minimum [22].

1.2.4. SUMMARY

We have introduced various methods of solving the SCED problem, this includes, conventional methods (which are mathematical methods), intelligence search methods (which are built around some basic principles taken from the observation of a particular natural phenomenon) and hybrid methods (which are a merger of two or more methods), and we have given an overview of some of the algorithms.

Solving the problem by different techniques we should arrive at the same conclusion. The difference of the approaches can be characterized by the time spent for the photo typing, the speed of convergence and ability to handle e.g. power loss reduction [23].

1.3. PROBLEM STATEMENT

To develop an improved out-of-kilter algorithm using MATLAB that could be used to realize the solution of the economic load dispatch (ELD) problem with security constraints of power systems.

1.3.1. OBJECTIVES

- To attain an optimal solution to the security constrained economic dispatch.
- To understand the improved out-of-kilter algorithm for formulation of SCED problem.
- To develop an IOKA algorithm to be used for simulating the SCED problem.
- To compare the results with those obtained from OKA for the 30 bus IEEE test Network.

CHAPTER TWO

LITERATURE REVIEW

2.1. LITERATURE REVIEW ON SECURITY CONSTRAINED ECONOMIC DISPATCH

2.1.1. SYSTEM SECURITY

Everything seems to have a propensity to fail. Power systems are no exception. Power system security practices try to control and operate power systems in a defensive posture so that the effects of these inevitable failures are minimized. Power system security is therefore the ability to maintain the flow of electricity from the generators to the consumers, especially under disturbed conditions [2].
Definition of states of power system operations:-

A normal state is the ideal operation condition wherein all the equipment are operating within their design limits and the demanded load is being met. Also, the power system can withstand a contingency without violation of any of the constraints. This means that the power system has reserve capacity (generation and transmission) available in this state. If this reserve capacity is reduced, or the possibility of major multiple contingencies increases due to adverse weather, then the system is said to be in the alert (insecure state). Although equipments are within their limits and load demand is met, the system is "weaker" and may not be able to withstand a contingency. Preventive control actions are required to get the system back to normal state.

If preventive control actions do not succeed, a power system remains insecure (in the alert state). If a contingency occurs, the system may go into the emergency state where overloading of equipment (above the short term ratings of equipment) occurs. The system can still be intact and can be brought back to the alert state by emergency control actions like fault tripping, generator tripping, load tripping, and HVDC power control. If these measures do not work, integrated system operation becomes unavailable and a major part of the system may be shut down due to equipment outages. This is in the extreme state. Load shedding and graceful or controlled system separation or islanding is necessary to prevent spreading of disturbances and a total grid failure.

If there is a widespread potential blackout, the surviving or restarted generators are connected to local loads, the restarted/surviving small power systems, the islands, are reconnected to correct the power system to alert or normal state. This is the corrective control.

Preventive and corrective actions are directed by a system operator as they involve manual control actions while the emergency control actions are done by protective relays or fast acting controls [24].

The three main functions of system security that are carried out in an energy control centre include:-

a) System monitoring

System monitoring supplies the power system operators or dispatchers with important real time data on the conditions of the power system on as load and generation change. Telemetry systems measure, monitor and transmit the data, voltages, currents, current flows and the status of circuit breakers and switches in every substation in a transmission network. Further, other critical and important information such as frequency, generator outputs and transformer tap positions can also be telemetered. Digital computers in a control centre then process the telemetered data and place them in a database form and inform the operators in case of an overload or out of limit voltage Alarms can be set and emails notification sent out if your system functions outside the normal constraints thus avoiding possible interruptions. Important data are also displayed on large size monitors. These days it also enables energy cost monitoring and accounting and real time load forecasting and trending.

b) contingency analysis

Contingency analysis is used to calculate violations e.g. in case of an unplanned outage and give information on remedial actions to remove the violations. It's a "what if" scenario simulator that evaluates, provides and prioritizes the impacts on electric power systems when problems occur. Contingency is the loss of a small part of the power system e.g. a transmission line [25].

c) preventive analysis

Corrective action analysis permits the operator to change the operation of the power system if a contingency analysis program predicts a serious problem in the event of the occurrence of a certain outage. A simple example of corrective action is the shifting of generation from one station to another. This may result in change in power flows and causing a change in loading on overloaded lines.

These three functions together consist of a very complex set of tools that help in the secure operation of a power system. These are done by energy management systems (EMS) and SCADA systems. The major function of the Energy management system (EMS) is to operate the system at minimum cost, with the guaranteed alleviation of emergency conditions. The emergency condition will depend on the severity of violations of operating limits (branch flows and bus voltage limits).The most severe violations result from contingencies. The power system should hence be able to withstand the effects of contingencies [2].

When preparing to deal with possible contingencies, operators consider:-

a) preventive actions
b) corrective actions

Preventive actions refer to the day-ahead adjustments of generation and transmission flows. Preventive actions are designed to put the system in a state such that the occurrence of a credible disturbance does not cause it to become unstable. These include regulation actions and load-following actions.

The *regulation* service is designed to handle rapid fluctuations in loads and small unintended changes in generation. This service helps maintain the frequency of the system at or close to its nominal value and reduce inadvertent interchanges with other power systems. Generating units that can increase or decrease their output quickly will typically provide this service. These units must be connected to the grid and must be equipped with a governor. They will usually be operating under automatic generation control. Generating units providing the *load-following* service handle the slower fluctuations. These units obviously must be connected to the system and should have the ability to respond to these changes in load. By keeping the imbalance close to zero and the frequency close to its nominal value, these services are used as preventive security measures.

Corrective actions refer to the dispatch of generating units and adjustments of transmission flow controls for mitigating transmission flow violations in real time. Corrective actions are intended to limit the consequences of a disturbance and are taken only if this disturbance occurs. This encompasses *reserve* services, which are designed to handle the large and unpredictable power deficits that could threaten the stability of the system.

Reserve services are usually classified into two categories. Units that provide *spinning reserve* must start responding immediately to a change in frequency, and the full amount of reserve capacity that they are supposed to contribute must be available very quickly. On the other hand, generating units providing *supplemental reserve services* do not have to start responding immediately.

As long as the production is equal to the consumption, the frequency and the interchanges remain constant. However, the balance between load and generation is constantly perturbed by fluctuations in the load, by imprecise control of the output of generators and occasionally by the sudden outage of a generating unit or of an interconnection. Large frequency deviations can lead to a system collapse. If the frequency drops too low, protection devices disconnect the generating units from the rest of the system to protect them from damage. Such disconnections exacerbate the imbalance between generation and load, causing a further drop in frequency and additional disconnections. The system operator must therefore take preventive measures to ensure that it can start applying the corrective measures as soon as large imbalances occur [26].

2.1.2. Classical Economic Dispatch

In an optimization problem, the objective is to optimize (minimize or maximize) some function *f*. This function is called the Objective function**.** For the case of economic dispatch, the aim is to minimize the total generation cost, C_T, defined as

$$C_{total} = \sum_{i=1}^{N} C_i(P_{Gi}) \qquad i = 1, \dots, N \qquad 2.01$$

Where

P_{Gi} =the active power generation of generation unit i

N = the number of generation units

$C_i(P_{Gi})$ = Generation cost of unit i

$C_i(P_{Gi})$ = is defined as

$$C_i(P_{Gi}) = \gamma_i P_{Gi}^2 + \beta_i P_{Gi} + \alpha_i \qquad 2.02$$

Whereγ_i, β_i andα_i are the cost coefficients of the ith generator. Eq2.02 therefore becomes:-

$$C_{total} = \sum_{i=1}^{n} C_i(P_{Gi}) = \sum_{i=1}^{n} \gamma_i P_{Gi}^2 + \beta_i P_{Gi} + \alpha_i \qquad 2.03$$

In most optimization problems the objective function f depends on several variables $x_1, \dots, x_n$. These are called control variables because we can "control them", that is, choose the values. For example, the total generation cost may depend on the fuel cost or marginal cost of operating each generation unit. Therefore classical economic dispatch involves finding the optimal values of these control variables $x_1, \dots, x_n$.

In many problems the choice of values of $x_1, \dots, x_n$ is not entirely free but is subject to some constraints, that is, additional restrictions arising from the nature of the problem and the variables. Two types of constraints are observed as follows:-

1. Power balance constraint - where the total power generation must cover the total power demand and the power loss. This is implied as below:-

$$\sum_{i=1}^{N} P_{Gi} = P_D \qquad \text{2.05 (Neglecting losses)}$$

$$\sum_{i=1}^{N} P_{Gi} = (P_D + P_L) \qquad \text{2.06 (Including losses)}$$

Where

$\sum_{i=1}^{N} P_{Gi}$ = is the total system generation of the network.

P_D = is the total system demand of the network.

P_L = is the system transmission losses, which is a function of the generation of each unit and system parameters related to the network model.

2. Generation capacity constraint/generating unit limits

The power output of any generator should not exceed its rating nor should it be lower than the minimum value necessary for stable boiler operation. Thus, the generations are restricted to lie within given minimum and maximum limits

$$\mathrm{P_{Gi-min}} \le \mathrm{P_{Gi}} \le \mathrm{P_{Gi-max}} \text{ for } \mathrm{i} = 1, \dots, \mathrm{N} \qquad 2.07$$

(Eq.2.05 – 2.06) refers to the balance of total generation with the total demand (P_D) and (2.07) refers to satisfying the generation level of each unit to be within its respective minimum and maximum limits [4].

The problem formulated above is a classical economic dispatch problem. The objective is to minimize total fuel cost that supplies the load demand subject to satisfying the power balance and generation units' capacity constraints on the system.

2.1.3. Security Constrained Economic Dispatch (SCED)

Security constrained economic dispatch incorporate the network security constraints to the classical economic dispatch problem. There are two constraints,

a) Equality constraints
b) Inequality constraints

2.1.3.1. Equality Constraints

The equality constraints of the SCED reflect the physics of the power system. The physics of the power system are enforced through the power flow equations which require that the net injection of the real at each bus to be zero as shown in Eq. 2.01.

1. Real Power Constraints

$$P_{GK} - P_{DK} = V_k \sum_{i=1}^{N} [V_j [G_{kj} \cos(\delta_k - \delta_j) + B_{kj} \sin(\delta_k - \delta_j)]] \qquad 2.08$$

where: $k = 1,2,\ldots,n$; $Pgk,$ active power generated at bus k; $Pdk,$: active power demand at bus k; $Vk, \delta k$: bus voltage magnitude and angle at bus k; Gkj, Bkj: conductance and susceptance of the (k,j) element in the admittance matrix [27].

2.1.3.2. Inequality Constraints

The inequality constraints of the SCED reflect the limits on physical devices in the power system as well as the limits created to ensure system security [4].
The inequality constraints to be included are as follows.

1. Generation capacity constraints:

Active power generation constraints for all units have been incorporated for stable operation. This means that the active power output of each generator in any network is restricted by lower and upper limits as follows [27].

$$\mathrm{P_{gi-min}} \le \mathrm{P_{gi}} \le \mathrm{P_{gi-max}} \text{ for } \mathrm{i} = 1\!:\mathrm{N_G} \qquad 2.09$$

Where,
Pgi = Unit MW generated by ith generator

$Pgi - max$ = Specified maximum MW generation by i[th] generator

$Pgi - min$ = Specified minimum MW generation by i[th] generator

2. Line Thermal Limit Constraints for all Transmission Lines

The power flow over a transmission line must not exceed the specified maximum limit because of the stability consideration.

$$P_{li-min} \leq P_{li} \leq P_{li-max} \quad for\ i = 1, \ldots, n_1 \qquad 2.010$$

Where,

P_{li} : The real power flow at line i

P_{li-max} : The maximum real power flow at line i

n_l : Number of transmission lines in a system

2.2. LITERATURE REVIEW ON OUT-OF-KILTER ALGORITHM

The out-of-kilter algorithm is a Network Flow Programming algorithm that was invented by Ford and Fulkerson [1962]. An improved formulation of Fulkerson's out-of-kilter algorithm which leads to more efficient computer implementation was developed by R.S. Barr, F. Glover, and D. Klingman. This method is modelled from flows in networks.

The common elements of a network are a collection of points called nodes, and a collection of arcs which cannot these nodes. The nodes are denoted by a single lower case letter i and arcs are denoted by naming the nodes they connect e.g. $arc(i,j)$. Some homogenous commodity, electrical power for our case, can flow over the arcs, and this is denoted by fij, the amount of commodity flowing on the arc(i,j) from the node i to the node j. If $f_{ij} > 0$ then the commodity flows from node j to node i [28].

Arcs have cost and capacity characteristics. Generally, some cost is incurred in moving a unit of the commodity from node i to node j, and this cost is denoted as Cij. This is $\$/MWh$ in the case of electrical power. The flow is also frequently limited by upper bounds or capacities in the arcs, these maximum arc capacities are denoted as Uij. There may also be a requirement for minimum amount of flow along any arc. This is denoted as Lij. We further assume all costs, flows and bounds are integers. The assumption of integral-valued parameters is used to demonstrate convergence of the out-of-kilter algorithm.

We associate with each node i, a variable π_i, which is considered as the price of a unit of the flow commodity at the nodes. The π_i will be related to the amount demanded at some nodes, since as the amount demanded increases, the overall difficulty in supplying it will increase.

The net cost or relative cost, $\hat{C}_{ij} = C_{ij} - \pi_i + \pi_j$. The new cost $\hat{C}_{ij}$ represents the total cost to the system of transporting one unit of flow from node i to node j. This definition compares the cost of retaining a unit at node i with the cost of moving it to node j. In moving a unit of flow from node i to node j, the commodity price at i, π_i, is foregone and an actual transportation cost is incurred. If the sum of these costs is greater than the commodity price at j, π_j, then it does not pay to ship a unit from i to j. $\hat{C}_{ij}$ will be positive. On the other hand, if a unit at j costs than at i plus the transportation cost, $\hat{C}_{ij}$ will be negative and the system benefits from the move, and hence the shipment from node i to node j is profitable. If the value at j, π_j is balanced exactly by the value at i plus the transportation cost, $(\pi_i + C_{ij})$, then $\hat{C}ij = 0$, and we are indifferent to an additional unit from node i to node j [28].

2.2.1. The Out-of-Kilter Algorithm

OKA algorithm transforms the original network into an out of Kilter network by introducing a "return arc" from sink node t to source node s while the internal flow retrains unchanged (i.e. from source nodes to sink node t). The return arc flow f_{ts} equals the original network flow r. The OKA network model, with n nodes and m arcs can be shown as below;

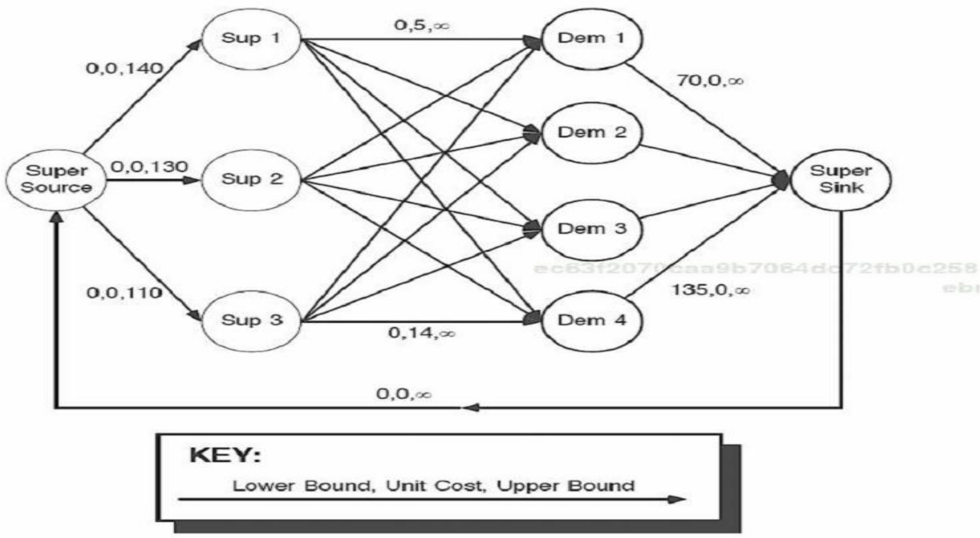

Figure 2.1: Sample out-of-kilter flow network

The general network problem requires that we find flows, f_{ii}, that minimize total cost (Eq. 2.1) while at the same time satisfying the constraining conditions (Eq. 2.3) and show that in a circulation what goes into a node must come out of the total.

Minimize

$$c = \sum_{ij} c_{ij} f_{ij} \quad \text{ij} \in m \qquad (2.1)$$

ij is an element of set m (total number of arcs in network)

Such that

$$\sum_{j \in n} \left(f_{ij} - f_{ji}\right) = r_{ij} \quad \text{i} \in n. \qquad (2.2)$$

$$L_{ij} \leq f_{ij} \leq u_{ij} \qquad \text{i}j \in m \qquad (2.3)$$

Where: c_{ij} = arc cost per unit

f_{ij}= flow on arc ij in the network

L_{ij}= lower bound of the flow on the arc ij in the network

U_{ij}= the upper bound of flow on arc ij in the network

n = total number of nodes in the network

m = total number of arcs in the network

The first constraint states that, the total outflow of a node minus the total inflow of the node must be equal to the flow balance (supply/demand value) of this node. This is known as the flow balance constraints. Next, the flow bound constraints model the physical capacities or constraints imposed on the flow's range. This optimization model describes the typical relationships between generators and load demands [3].

According to the Dual Theory, the corresponding primary problem and dual problem can be expressed as below:

Primary problem

$$\text{Max } F' = -\sum_{ij} c_{ij} f_{ij} \qquad (2.5)$$

Such that:

$\sum_{j \in n} \left(f_{ij} - f_{ji} \right) = 0$ (2.6)

$L_{ij} \leq f_{ij} \leq u_{ij}$ i$\in n$, j = n, i$j \in (m + ss + tt + 1)$ (2.7)

Dual problem

Min

G= $\sum_{ij} U_{ij} \alpha_{ij} - \sum L_{ij} \beta_{ij}$ (2.8)

Such that

$c_{ij} + \pi_i - \pi_j + \alpha_{ij} - \beta_{ij} \geq 0$ (2.9)

$\alpha_{ij} \geq 0, \beta_{ij} \geq 0, i\epsilon n, j\epsilon n, ij\epsilon(m + ss + tt + 1)$ (2.10)

π= dual variable of variable f of primary problem

α & β = dual variables of upper and lower limits L_{ij}, U_{ij} of primary problem.

If f,n,α,β meet constraints, then

$G - F' = \sum_{ij} U_{ij} \alpha_{ij} - \sum_{ij} L_{ij} \beta_{ij} + \sum_{ij} C_{ij} f_{ij}$ (2.11)

$=0(\pi_s - \pi_t) + \sum_{ij} U_{ij} \alpha_{ij} - \sum_{ij} L_{ij} \beta_{ij} + \sum_{ij} C_{ij} f_{ij}$ (2.12)

$=\sum_j \sum_i \pi_i \left(f_{ij} - f_{ji} \right) + \sum_{ij} U_{ij} \alpha_{ij} - \sum_{ij} L_{ij} \beta_{ij} + \sum_{ij} C_{ij} f_{ij}$ (2.13)

$=\sum [\pi_i - \pi_j + \alpha_{ij} - \beta_{ij} + C_{ij}] f_{ij} + \sum_{ij} (U_{ij} - f_{ij}) \alpha_{ij} + \sum_{ij} (f_{ij} - L_{ij}) \beta_{ij} \geq 0$ (2.14)

$G - F' = 0$ If the solution is optimal

Therefore from (2.14)

$\left[\pi_s - \pi_t + \alpha_{ij} - \beta_{ij} + C_{ij} \right] f_{ij} = 0$ (2.15)

$\left(U_{ij} - f_{ij} \right) \alpha_{ij} = 0$ (2.16)

$(f_{ij} - L_{ij}) \beta_{ij} = 0$ (2.17)

From Eq.1.4

$\left[\hat{C}_{ij} + \alpha_{ij} - \beta_{ij} \right] f_{ij} = 0$ (2.18)

With $\hat{C}_{ij} = C_{ij} + (\pi_s - \pi_t)$ (2.19)

Where $\hat{C}ij$ represents the marginal cost or updated cost associated with arc(i, j) and π is the node “potential” or multipliers [3].

From (2.15) - (2.19), we get the three in-kilter cases :

Case 1: $Cij > 0$

If $\beta ij = \hat{C}ij + \alpha ij,\ fij \neq 0$

Furthermore,$if\ \alpha ij \geq 0,\ \beta ij \neq 0$ then from (2.17) we can get

$$fij = Lij$$

Case 2: $Cij < 0$

If $\beta ij = Cij + \alpha ij$, then $f\ ij \neq 0$, and $\alpha\ ij > \beta i\ j$

Furthermore, if$\beta ij \geq 0$,$\alpha\ ij \neq 0$ then from (2.16) we can get

$$fij = Uij$$

Case 3: $Cij = 0$

From (1.7), we get ($\alpha\ ij - \beta\ ij$) $f\ ij = 0$, which can be analyzed as follows:

(3a) If $f\ ij = 0$, then ($\alpha\ ij - \beta\ ij$) $\neq 0$

When $\alpha\ ij > \beta\ ij$, then $\alpha\ ij > 0$, in this way, we get the following expression from (2.16):

$$fij = Uij \neq 0$$

When $\beta\ ij > \alpha\ ij$, then $\beta\ ij > 0$, in this way, we get the following expression from (2.17):

$$fij = Lij \neq 0$$

Both situations are conflicted with the assumption $f\ ij = 0$. So we can be sure $f\ ij \neq 0$ for this case.

(3b) Assuming $\alpha\ ij = 0$, then $\beta\ ij\ f\ ij = 0$

Since $f\ ij \neq 0$ from (3a), we have $\beta\ ij = 0$

Therefore, from (2.16) we get

$$fij \leq Uij$$

From (2.17) we get

$$fij \geq Lij$$

That is, if $Cij = 0$, then

$$L\,ij \leq f\,ij \leq U\,ij$$

Complementary Slackness Condition for Optimality of Out-of-Kilter Algorithm

Limitations on the permissible flow levels together with permissible levels of system cost yield the following conditions that will be satisfied by an optimal solution to the minimum cost flow problem [3].

1. If $\hat{C}_{ij} < 0$ $\quad f_{ij} = U_{ij}$ (2.20)

2. If $\hat{C}_{ij} = 0$ $\quad L_{ij} \leq f_{ij} \leq U_{ij}$ (2.21)

3. If $\hat{C}_{ij} > 0$ $\quad f_{ij} = L_{ij}$ (2.22)

Table 2.1: States of OKA arcs

Symbol	$\hat{C}_{ij}$	f_{ij}	State of Arcs
I_1	$\hat{C}_{ij} > 0$	$f_{ij} = L_{ij}$	In-kilter
I_3	$\hat{C}_{ij} = 0$	$L_{ij} < f_{ij} < U_{ij}$	In-kilter
		$f_{ij} = U_{ij}, f_{ij} = L_{ij}$	In-kilter
I_3	$\hat{C}_{ij} < 0$	$f_{ij} = U_{ij}$	In-kilter
II_1	$\hat{C}_{ij} > 0$	$f_{ij} < L_{ij}$	In-kilter
II_2	$\hat{C}_{ij} = 0$	$f_{ij} < L_{ij}$	In-kilter
II_3	$\hat{C}_{ij} < 0$	$f_{ij} < U_{ij}$	In-kilter
III_1	$\hat{C}_{ij} > 0$	$f_{ij} > L_{ij}$	In-kilter
III_2	$\hat{C}_{ij} = 0$	$f_{ij} > U_{ij}$	In-kilter
III_3	$\hat{C}_{ij} < 0$	$f_{ij} > U_{ij}$	In-kilter

The complementary slackness conditions for optimality of OKA shown in equations (2.20) – (2.22) correspond to the three “in - kilter” states of the arcs. In addition, there are six “out - of - kilter” states that do not satisfy conditions (2.20) - (2.22) as shown in Table 1.0.

The states of arcs can be explained with Figure 1.0.

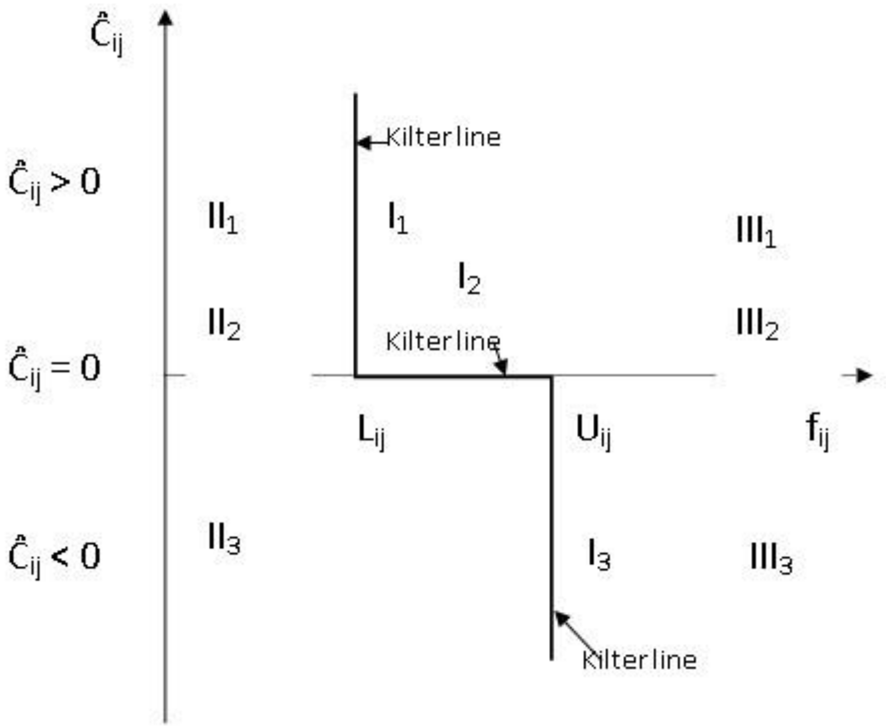

Figure 2.2: States of OKA arcs

In Figure 1.0, if the arc is in the in - kilter state, the point $(f\ ij , Cij)$ will be located on one of three dark lines $I_1 , I_2 , and\ I_3$, where the dark line I_1 corresponds to the lower bound $L\ ij$ of flow $f\ ij$; the dark line I_3 corresponds to the upper bound $U\ ij$ of flow $f\ ij$; and the dark line I_2 corresponds to the flow $f\ ij$ that is within $L\ ij < f\ ij < U\ ij$. If the flow of the arc is violated at the upper or lower limits, the point $(f\ ij , Cij)$ will be located out of three dark lines, which correspond to six “out - of - kilter” states in Figure 1.0. In these situations, the value of flow of the arc will be either less than it’s lower limit or higher than its upper limit, that is, $f\ ij > U\ ij\ or\ f\ ij < L\ ij$.

From the figure 1.0,	at	$II_{1,} II_2$	$f_{ij} < L_{ij}$
	At	$III_{2,} III_3$	$f_{ij} < U_{ij}$
	At	III_1	$f_{ij} > L_{ij}$
	At	II_2	$f_{ij} < U_{ij}$

If all the arcs are in kilter, then the optimal solution is obtained. Otherwise, we must vary the relevant arc flows or node potentials (parameter π) by the labelling technique so that the out - of - kilter states of the arcs come into kilter [3].

Labelling Rules and Algorithm of OKA

The out-of-kilter algorithm defines certain "kilter" conditions which; taken together, constitute primal and dual feasibility criteria for arcs in a network. The method brings each non-conforming ("out-of-kilter") arc into kilter by adjusting its flow or changing its node potentials. In order to accomplish this, a labelling procedure is used which, after one or more applications, identifies a loop containing the non-conforming arc.

In order to change the flow of an out-of-kilter arc (s, t), a suitable path (called a flow augmenting path) must be found from node t to node s and flow adjusted on each arc of the path by an equal amount, thus maintaining node conservation. This path is found (or its nonexistence determined) by alternating use of a labelling procedure and a potential change procedure [29].

According to labelling technique, the labelling rules of OKA for the forward arc and backward arc under nine OKA states shown in Table 1.0 above are listed in Table 1.1, where symbol “ $\uparrow$ ” stands for increase, “ $\downarrow$ ” stands for reduce, “ $\rightarrow$ ” stands for change, and “ f_k ” indicates that the flow is outside of the feasible region [25].

Table 2.2: Labelling Rules of OKA algorithms

Symbol f_{ij}		Forward arc f^+ labelling? Why?	Forward arc f^- labelling? Why?
I_1	$f_{ij} = L_{ij}$	$No, f^+ \uparrow \rightarrow f_k^+$	$No, f^- \downarrow \rightarrow f_k^-$
I_2	$L_{ij} < f_{ij} > L_{ij}$	$Yes, f^+ \uparrow \rightarrow U$	$Yes, f^- \downarrow \rightarrow L$
	$f_{ij} = U_{ij}$ $f_{ij} = L_{ij}$	$No, f^+ \uparrow \rightarrow f_k^+$	$No, f^- \downarrow \rightarrow f_k^-$
I_3	$f_{ij} = U_{ij}$	$No, f^+ \uparrow \rightarrow f_k^+$	$No, f^- \downarrow \rightarrow f_k^-$
II_1	$f_{ij} < L_{ij}$	$Yes, f^+ \uparrow \rightarrow U$	$No, f^- \downarrow \rightarrow f_k^-$
II_2	$f_{ij} < L_{ij}$	$Yes, f^+ \uparrow \rightarrow U$	$No, f^- \downarrow \rightarrow f_k^-$
II_3	$f_{ij} < U_{ij}$	$Yes, f^+ \uparrow \rightarrow U$	$No, f^- \downarrow \rightarrow f_k^-$
III_1	$f_{ij} > L_{ij}$	$No, f^+ \uparrow \rightarrow f_k^+$	$Yes, f^- \downarrow \rightarrow L$
III_2	$f_{ij} > U_{ij}$	$No, f^+ \uparrow \rightarrow f_k^+$	$Yes, f^- \downarrow \rightarrow L$
III_3	$f_{ij} > U_{ij}$	$No, f^+ \uparrow \rightarrow f_k^+$	$Yes, f^- \downarrow \rightarrow U$

According to the labelling rules, the out-of-kilter algorithm is implemented as follows.

With incremental flow Loop

When there exists an incremental flow loop, correct the values of flow for all arcs in the loop. The process is as below:

(1) For forward arcs

(A) If $\hat{C}ij > 0,\ ij < L\,ij$, the node *j* is able to be labelled. The incremental flow to the node *j* will be computed as

$$qj = min[qi\ , Lij - fij] \qquad (2.23)$$

(b) If$\hat{C}ij \leq 0\ , f\ ij < U\ ij$ the node *j* is able to be labelled. The incremental flow to the node *j* will be computed as

$$qj = \min[qi\ , Uij - fij] \qquad (2.24)$$

(2) For backward arcs

(A) If $\hat{C}ji \geq 0, > L\,ji$, the node j is able to be labelled. The incremental flow to the node j will be computed as

$$qj = \min[qi\,, fji - Lji] \qquad (2.25)$$

(b) If $\hat{C}ji < 0, f\,ji > U\,ji$, the node j is able to be labelled. The incremental flow to the node j will be computed as

$$qj = \min[qi\,, fji - Uji] \qquad (2.26)$$

The label eligible conditions are summarised in the Table 2.0 below;

Table 2.3 Label eligible conditions

Label eligible	conditions	q_j
$(i,j) \epsilon N$	$\hat{C}ij > 0, f\,ij < L\,ij$	$qj = min[qi\,, Lij - fij]$
$(i,j) \epsilon N$	$\hat{C}ij \leq 0\,, f\,ij < U\,ij$	$qj = \min[qi\,, Uij - fij]$
$(i,j) \epsilon N$	$\hat{C}ji \geq 0, f\,ji > L\,ji$	$qj = \min[qi\,, fji - Lji]$
$(i,j) \epsilon N$	$\hat{C}ji < 0, f\,ji > U\,ji$	$qj = \min[qi\,, fji - Uji]$

Without Incremental Flow Loop

When there does not exist an incremental flow loop, correct the values of the relative cost $\hat{C}ij$, or $\hat{C}ji$ by increasing the cost of the vertex π . This is because the change of $\hat{C}ij$, or $\hat{C}ji$ causes the change of the path of minimum cost flow. Consequently, a new incremental flow loop will be produced. The process of computing the incremental vertex cost is as below.

Let B and $\mathcal{B}$ stand for the set of the labelled vertexes and unlabelled vertexes, respectively. Obviously, the super source $s \in B$, and super sink $t \in$ Ɓ. In addition, define two sets of arcs $A1$ and $A2$ [28]:

$$A1 = \{ij, i \in B, j \in Ɓ, \hat{C}ij > 0, fij \leq Uij\} \qquad (2.27)$$

$$A2 = \{ji, i \in B, j \in Ɓ, \hat{C}ji < 0, fji \geq Lij\} \qquad (2.28)$$

The incremental vertex cost is determined as below

$\delta = min\{\delta 1, \delta 2\}$ (2.29)

Where

$\delta 1 = min\{|\hat{C}ij|\} > 0$ (2.30)

$\delta 2 = min\{|\hat{C}ji|\} > 0$ (2.31)

If $A1$ is an empty set, make $1 = \infty$; if $A2$ is an empty set, make $2 = \infty$. When $\delta = \infty$, it means there is no feasible flow, which is no solution for the given NFP problem. When $\delta < \infty$, update the vertex costs for all unlabelled vertexes, that is [28],

$\delta' = \pi j + \delta \; j \in \text{Ɓ}$ (2.32)

The main features of the out-of-kilter algorithm are:-

1. None zero lower bound of flow may be feasible.
2. The initial flow does not have to be feasible or zero flow.
3. Non-negative constraints $f_{ij} > 0$are released.
4. It is easy to imitate a change in network topology by changing bound values of flows as the branch outages occurs [3].

2.2.1. The improved out-of-kilter Algorithm

In our reformulation all of the above processes are redesigned and the network representation is altered to allow more efficient processing. Nevertheless, the basic logic of the original method is maintained. The reformulation modifies the labelling procedure in a manner which causes it to process less information on the forward pass and more information on the reverse pass. Net computational savings result because the reverse pass typically involves only a portion of the nodes encountered on the forward pass, and sometimes the reverse pass is not executed at all [25].

In addition to the new labelling scheme, the reformulated algorithm employs a special classification scheme for determining the "kilter status" of each arc. This scheme permits the current net capacity and marginal cost of the arc to be evaluated with increased efficiency, which in turn expedites the determination of both the flow augmenting path, when it exists, and of the new marginal cost assignment (via implicitly modified "node potentials") when the flow augmenting path does not exist or is blocked.

Basic to both of these schemes is a simple change in arc representation which reclassifies each arc into an "original" arc and a "mirror," at an almost negligible increase in computer memory requirements. The reclassification, which is only symbolic and does not introduce any structural change in the network, causes each arc to become capacitated only from above, rather than from both below and above. Special relationships between the capacities, flows and marginal prices of an arc and its mirror are maintained by which the "mirror of the mirror" is identified as the original.

The effect of this scheme is to reduce markedly the number of mathematical conditions which characterize the "kilter states" applicable to the arcs. The kilter states, which identify the degree to which the current arc flow conforms to or deviates from optimality, require repetitive monitoring in the out-of-kilter method. Thus the ability of the "mirror arc" representation to contract the range of conditions by which these states are recognized leads to a convenient streamlining of the solution process.

Label-eligible nodes are also recognized with improved efficiency by the reformulation, utilizing a "partitioned successor" technique. After an initialization stage, the partitioned successor technique enables label eligibility to be ascertained with only a portion of the customary effort.

The reformulation of the network proceeds quite simply as follows. Each arc (i, j) of the original network N is split into two interdependent pseudo-arcs. The first pseudo-arc is directed from node i to node j and has the same cost $c\ ij$ and upper bound Kij as the original arc (i, j), but no lower bound. The other pseudo-arc is directed from node j to node i with its cost equal to the negative of the cost cij, and its upper bound equal to the negative of the lower bound Lij of the original arc. This pseudo-arc, like the first, has no lower bound. The flows on the two pseudo-arcs are respectively equal to the flow of the original arc xij and the negative of this flow [25]. If we denote the flow variables of the pseudo-arcs by $x_{ij}^{'}$ and $x_{ij}^{''}$, respectively, we thus have

$$x_{ij}^{'} = x_{ij} \tag{2.33}$$

And

$$x_{ij}^{''} = -x_{ij} \tag{2.34}$$

This further implies

$$x_{ij} = \frac{1}{2}(x_{ij}^{'} - x_{ij}^{"}) \quad (2.35)$$

Replacing Xij In the objective function (1) and the node conservation we obtain (discarding the one-half multiple in the objective function) the equivalent network problem N [25]:

Minimize

$$C = \sum_{ij} C_{ij}(x_{ij}^{'} - x_{ij}^{"}) \quad ij \epsilon N \quad (2.37)$$

Subject to

$$\sum_{ij}[(x_{ij}^{'} - x_{ij}^{"}) - (x_{ij}^{'} - x_{ij}^{"})] = 0 \quad i = 1,2 \dots m \quad (2.38)$$

$$x_{ij}^{"} \leq -L_{ij} \quad ij \epsilon N \quad (2.39)$$

$$x_{ij}^{'} \leq U_{ij} \quad ij \epsilon N \quad (2.40)$$

$$(x_{ij}^{'} + x_{ij}^{"}) = 0 \quad ij \epsilon N \quad (2.41)$$

The new network disposes of lower bounds in the original network at the expense of doubling the number of arc variables and introducing the extra constraints (2.41). Although this appears to be a trade-off in the wrong direction, we will now show how to reverse this seeming disadvantage. To begin, observe the following structural properties of this network:

1. for each arc into a node there exists a "mirror" arc out of the node.
2. Each arc is only bound from above. This property may be used to define a net capacity for each pseudo-arc as follows:

$$nC_{ij}^{'} = U_{ij} - x_{ij}^{'} \quad (2.42)$$

$$nC_{ij}^{"} = -L_{ij} - x_{ij}^{"} \quad (2.43)$$

That is, net capacity corresponds to the capacity of a pseudo-arc to accept a flow increase without exceeding its upper bound. From constraint (12) a change of flow on a pseudo-arc induces the negative of this change in the flow on its mirror, and thus correspondingly changes the net capacities of these arcs by equal but opposite amounts.

The marginal cost $\hat{C}'ij$ of the pseudo-arc associated with x'_{ij} is equal to the marginal cost of the original arc (ij); namely, $\hat{C}_{ij}$. On the other hand, the marginal cost $\hat{C}"ij$ of the mirror pseudo-arc is equal to $-\hat{C}_{ij}$. Thus a change in the marginal cost of a pseudo-arc produces an equal but opposite change in the marginal cost of its mirror. A foundation for exploiting these properties is given by the following scheme for data organization [25].

With each main arc associate a unique integer a_{ij} which is a number between 1 and n; and with its mirror arc, associate the number (aij + n). These integers will be called arc numbers.

Also associate with each node i the set Bi of all arc numbers associated with pseudo-arcs directed out-of node i. Thus B_i= {b: b = aij if (i,j) ϵN or b =(a ij+ n) if (j,i) ϵN}. In addition, let Ĉ(b), nc(b) and m(b) denote functions whose values correspond to the margin cost, net capacity, and mirror arc number, respectively, associated with arc number b; i.e.,

$$\text{Margin cost, } \hat{C}(b) = \begin{cases} C_{ij}^{'} \ if\ b\ =\ aij \\ C_{ij}^{"} \ if\ b\ =\ aij + n \end{cases} \tag{2.45}$$

$$\text{Net capacity, } nC(b) = \begin{cases} nC_{ij}^{'} \ if\ b\ =\ aij \\ nC_{ij}^{"} \ if\ b\ =\ aij + n \end{cases} \tag{2.46}$$

$$\text{Mirror arc number, } m(b) = \begin{cases} b + n\ if\ b\ =\ aij \\ b - n\ if\ b\ =\ aij + n \end{cases} \tag{2.47}$$

Finally, let ϒ(b) denote a function such that for each arc number b, its value is the "to-node" index j corresponding to the direction of the pseudo arc; i.e.

$$\gamma(b) = \begin{cases} j\ if\ b\ =\ aij \\ j\ if\ b\ =\ aij + n \end{cases} \tag{2.48}$$

This data organization allows us (1) to associate a unique number with each pseudo-arc in a reformulated network, (2) to quickly determine the mirror's arc number, given the main's number and vice versa, (3) to identify the set of outward-directed pseudo-arcs for a given node, and (4) to determine the marginal cost and net capacity associated with an arc number. Ways of restructuring the out-of-kilter method to afford more significant advantages will now be considered.

New Labelling Procedure

To provide a point of reference, suppose the label process of the original out-of-kilter algorithm is to be applied at node i, which is currently labelled. Let Ĺ denote the set of unlabelled nodes. Then, using Ĉ(b), nc(b), m(b), and ϒ(b) as defined above, if arc (i,j)εN (implying that the arc number (aijϵBi), the

λ and μ label eligibility conditions may be stated as:

λ: Ĉ(b) > 0, nc(m(b)) < 0, γ(b)εĹ

μ: Ĉ(b) > 0, nc(b) < 0, γ(b)εĹ

Similarly if arc (J,i) εN (implying that the arc number b = (aji + n)εBl, then the ν and ρ label eligible conditions may be stated as:

ν: Ĉ(b) ≤ 0, nc(b) > 0, γ(b)εĹ

ρ: $\hat{C}(b) > 0$, $nc(m(b)) < 0$, $\gamma(b)\varepsilon\acute{L}$

Thus, using the new data structure, the μ and ν conditions become identical (as a consequence of the relationship between marginal costs for a pseudo arc and its mirror). Similarly the λ and ρ conditions become identical. Thus our restructuring has reduced the label eligibility conditions from four to two, and the standard labelling procedure can be simplified to examining each arc number in Bi one at a time, checking the new label eligibility states λ and μ (=ν , and ρ). However, the calculations can somewhat be improved more than this [25].

Partitioning

The labelling process can be facilitated by partitioning the set Bi into those arc numbers whose associated arcs are label-eligible and those which are not. We denote the set of label eligible arc numbers by Pi and the remaining arc numbers by Qi = Bi - Pi.

Once the sets Pi and Qi of the partition have been identified, the forward tree generation of the labelling process is accelerated since checking of net capacity, marginal cost, and direction of each arc is eliminated. Further, once the partition is determined it requires minimum maintenance. In particular, this maintenance involves checking the arcs in the flow augmentation cycle during breakthrough and checking those arcs affected by a change of node potentials. Another benefit of the partition is that knowledge of the set Qi shortcuts the determination of the sets used to determine node potential changes [25].

To implement the labelling process, the label for a given node j is simplified to consist only of the predecessor arc number b corresponding to pseudo-arc (i,j) across which node j is labelled. This change coupled with the partitioning of Bi has the following theoretical and computational advantages [25]:

1. The use of predecessor arc numbers instead of a node index label avoids possible ambiguity in the reverse pass when a network contains multiple arcs between a given pair of nodes.

2. By eliminating the αj values in the label, less information is processed on the forward tree generation and more information is processed on the reverse pass; however, the flow augmentation pass typically involves only a portion of the nodes labelled on the forward pass, and sometimes the reverse pass is not executed at all.

3. Due to the network reformulation the direction of each arc in Pi is out-of-node i; thus, node labelling can be carried out quite rapidly using the function y. That is, each arc number b ε Pi

is checked to determine whether node y(b) is labelled. If it isn't, y(b) is labelled with b; otherwise, the next arc number in Pi is examined.

A concise statement of the new labelling procedure (for the objective of increasing flow on pseudo-arc (s,t) or decreasing flow on pseudo-arc (t,s) is as follows:

1. Label node t with the arc number of pseudo-arc (s,t).

2. For any arbitrary labelled node i (i.e., i_1, L), label each un labelled node j (where j = y(b), bεPi) with b.

3. If node s is labelled, go to breakthrough. If s cannot be labelled, go to the potential change procedure before returning to step 2.

Breakthrough

In determining the flow change α, the new network representation once again contributes to decreased computational effort in this case by using net capacity nc_{ij} to eliminate a subtraction operation and by using marginal costs to distinguish states [25].

The exact calculations performed in updating the flows are as follows:

1. Let C denote the node labels in the flow augmenting path which are in label-eligible state μ(ϑ); similarly,, let C_λ denote the node labels, in this path which are in the label-ineligible state λ(=ρ). Then

α=min [min nc(b), min nc(m(b))]

$b \varepsilon C_\mu$ $b \varepsilon C_\lambda$

2. Retrace the flow augmenting path to update the net capacity of each pseudo-arc and its mirror by setting:

nc(b) = nc(b) - α

nc(m(b)) = nc(m(b)) +α for all bεC, U C_λ

3. Simultaneously update the sets P4 and, for each node i on the flow augmenting path as follow:-

a) Move the arc number from Qi to Pi for the pseudo-arc (if it exists) whose mirror lies in the flow augmenting path, provided the arc's net capacity becomes positive during the flow change and its marginal cost equals zero. (Only such arcs can change from label ineligibility to label eligibility.)

b) Move the arc number from Pi to Qi for the pseudo-arc (if it exists) that lies in the flow augmenting path, provided either:

1. its marginal cost is non-positive and its new net capacity is zero;
2. Or its marginal cost is positive and its new mirror capacity is zero.

New Potential Change Procedure

By keeping a list of the labelled nodes L and using the sets Bi, the membership conditions of S1, S2, RI, and R2 can be halved., The arcs contained in S1,S 2, *R 1 ,* and R 2 can then be found simply by examining the arcs in the sets Q associated with labelled nodes. To see this, note that both nodes of any arc bεPi will be labelled if node i itself is labelled. Thus, attention may be restricted to arcs bcQi to find all labelled unlabelled (and unlabelled-labelled) arcs. An additional simplification facilitates the determination of arcs in S1 and S2, by means of the following observation [25].

If iεL and bεQi, then arc b (or its mirror) is in S1 or S 2 if and only if γ(b)εĹ, Ĉ(b) > 0, and nc(b)>0.

To take advantage of this observation in updating the current marginal costs let Si denote the set of all arc numbers bεQi such that arc b (or its mirror) is in S1 or S2. Also let Ri denote the set of all arc numbers bεQi such that arc b is a labelled-unlabelled arc and b¢Si; i.e.

R = {b: γ(b)εĹ, b¢Si}. Finally let S = UŜi and R = UR̂i, where these unions are taken over the index set of labelled nodes. Then we identify the marginal cost increment Θ by

Θ = min Ĉ(b) bεS

(If this calculation cannot be performed because S is empty, the problem has no feasible solution.) The new marginal costs may accordingly be found by updating the current marginal costs as follows [25]:

Ĉ(b) = Ĉ(b) - Θ bεSUR

Ĉ(m(b)) = Ĉ(b) bεSUR

Ĉ(b) = Ĉ(b), for all other arc numbers.

In order to maintain the partition efficiently, arc numbers are checked for movement from set Pi to Qi or from Qi to Pat the same time the marginal costs are updated.

The checks required are the following:

1. If $b \varepsilon Si$, $nc(b) > 0$ and the updated marginal cost $\hat{C}(b) = 0$, then arc b has entered v state and should be moved from Qj to Pi.
2. If $b \varepsilon Ri$, $nc(b) = 0$, the old marginal cost $\hat{C}(b) > 0$, and the new marginal cost $\hat{C}(b) < 0$, then arc m(b) becomes label-ineligible and should be moved from Pi to Qi.

To minimize effort in the determination of sets S and R where more than one change of potential must be effected to bring a given arc in-kilter, the previous sets S and R are purged of currently nonconforming arcs and then augmented by appropriate elements of the sets Qi (where node i has been labelled since the previous potential change). Specifically, let L' denote the set of nodes which have been labelled since the last potential change, and let L and $\acute{L}$ denote all currently labelled and unlabelled nodes, respectively (i.e., L' c L). Then we can define the updated form of sets S and R as follows:

$S' = \{(b \varepsilon S: \gamma(b)\ \varepsilon \acute{L}, \hat{C}(b) > 0\}\ U\ \{b: \gamma(m(b)) \varepsilon L', y(b) \varepsilon \acute{L}, \hat{C}(b) > 0,$ (2.49)

$R' = \{beR: \gamma(b)\ \varepsilon \acute{L}\ U\ \{b \varepsilon S: \gamma(b)\ \varepsilon \acute{L}, \hat{C}(b) > 0\}\ U\ [\{b: y(m(b))L', \gamma(b)\ \varepsilon \acute{L}\} - \hat{S}]$ (2.50)

Thus S is screened to form subsets of S' and R', R is purged of non-conforming arcs and added to R', and the sets Qt for $i \varepsilon L'$ are scanned to form the remaining elements of S' and R'. This "recycling" of the sets S and R further restricts the population from which the new cut set is generated and, thus, reduces the effort required to effect multiple potential changes [25].

The basic steps of the new potential change procedure are:

1. If a previous potential change has been made for the current breakthrough attempt, perform step 3; otherwise go to step 2.

2. For each labelled node i, inspect Qi and create the sets $\hat{S}i$ and $\hat{R}i$. Let $S = U\hat{S}i$, and $R = U\hat{R}i$ and go to step 4.

3. Form new sets S and R as in (2.49) and (2.50).

4. If $S = \emptyset$, stop; there is no feasible solution to the problem. Otherwise, let $\Theta = \min_{b \varepsilon S} \hat{C}(b)$.

5. If $b \varepsilon SUR$ update the relative costs as follows:

$$c(b) = c(b) - \Theta$$

$$c(m(b)) = c(m(b)) + \Theta$$

While simultaneously updating the partition [25]

New Kilter Conditions

The six original out-of-kilter conditions may be compacted to the following four states for a given pseudo-arc number b:

Condition	corrective action
A: $nc(b) < 0$	decrease flow on b
E: $nc(m(b)) < 0$	increase flow on b
F: $nc(b) > 0$, $\hat{C}(b) > 0$	increase flow on b
G: $nc(m(b)) > 0$, $\hat{C}(m(b)) < 0$	decrease flow on b

The equivalence of these conditions to the six original out-of-kilter conditions is easily established. From the symmetric properties of these conditions, it follows that whenever a pseudo-arc is in-kilter then its mirror is also in-kilter. Once a pseudo-arc is in-kilter it will, moreover, always remain in-kilter due to the equivalence of the new labelling, breakthrough, and potential change procedures to the original procedures. Thus by successively putting the "main" pseudo-arcs in-kilter (i.e., those pseudo-arcs corresponding to original arcs), an optimal solution can be obtained by examining each of these arcs once. Alternatively, if pseudo- arcs are inspected as main-mirror pairs in the state determination process, only states A, E, and F need be checked for the main pseudo-arc and only state G need be checked for the mirror (i.e., the main pseudo-arc can be considered in-kilter if it violates states A, E, and F, whereupon the mirror is immediately put in-kilter via state G) [25].

CHAPTER 3

SOLUTION OF THE SECURITY CONSTRAINED ECONOMIC DISPATCH USING IOKA

3.1. FORMULATION OF THE SECURITY CONSTRAINED ECONOMIC DISPATCH BY IMPROVED OUT-OF-KILTER (IOKA)

For N thermal units, the formulation of the economic dispatch problem is as given below:-

Min $F = \sum_{i=NG}^{n} a_i P_{Gi}^2 + b_i P_{Gi} + c_i$ (3.1)

Such that

$\sum_{i(\omega)} P_{Gi}^2 + \sum_{j(\omega)} P_{Tj}^2 + \sum_{k(\omega)} P_{Dk}^2 = 0 \quad \omega \in n$ (3.2)

$P_{Gi-min} \leq P_{Gi} \leq P_{Gi-max}$ (3.3)

$P_{Ti-min} \leq P_{Ti} \leq P_{Ti-max}$ (3.4)

$$i \in NG, \qquad j \in NT, \ \ k \in ND$$

a_i, b_i, c_i are the cost coefficients of the i^{th} generator.

The IOKA network model of economic power dispatch consists of three types of arcs. These are the generation arc, the transmission arc and the load arc. Obviously, each generation arc corresponds to a generator, each transmission arc corresponds to a line or transformer, and each load arc corresponds to a real power demand. In addition, there is a return arc. Comparing economic dispatch shown in (3.0 – 3.4) with IOKA model shown in (2.37) – (2.41) the average cost and flow limits of each arc are:-

1. The generation arc

$\overline{C_{ij}} = a_i P_{Gi} + b_i$ (3.5)

$L_{ij} = \underline{P_{Gi}}$ (3.6)

$U_{ij} = \overline{P_{Gi}}$ (3.7)

2. The transmission arc

$\overline{C_{ij}} = hR_j P_{Tj}$ (3.8)

$L_{ij} = \underline{P_{Ti}}$ (3.9)

$$U_{ij} = \overline{P_{Ti}} \quad (3.10)$$

3. The load arc

$$\overline{C_{ij}} = 0 \quad (3.11)$$

$$L_{ij} = P_{Dk} \quad (3.12)$$

$$U_{ij} = P_{Dk} \quad (3.13)$$

4. The return arc

$$\overline{C_{ij}} = 0 \quad (3.14)$$

$$L_{ij} = \sum_{k \in ND} P_{Dk} + \frac{1}{2} \sum_{j=NT}^{n} R_j P_{Tj}^2 \quad (3.15)$$

$$U_{ij} = \sum_{k \in ND} P_{Dk} + \frac{1}{2} \sum_{j=NT}^{n} R_j P_{Tj}^2 \quad (3.16)$$

3.2. IOKA ALGORITHM FOR SECURITY CONSTRAINED ECONOMIC DISPATCH

The essence of the OKA is to revise the out - of - kilter states of arcs to in - kilter states according to complementary slackness conditions for optimality equations (2.20) – (2.22).

The steps followed in solving the IOKA based SCED problem is as follows [25].

1. Set initial values of the flows f_{ij} and node potentials π_{ij}.
2. Calculate the kilter status for all arcs. Arbitrarily choose an out of kilter arc. If none exists, stop: The current flows and node potential values are optimal.
3. Apply the labelling procedure for the out of kilter arc. If the arc is either in state L1, B1 or K1, set s = i and t = j, otherwise, if arc is in state L2, B2 or K2 set s = j and t = i.

L1:	$\overline{C_{ij}} > 0$	$f_{ij} < L_{ij}$
L2:	$\overline{C_{ij}} > 0$	$f_{ij} > L_{ij}$
B1:	$\overline{C_{ij}} = 0$	$f_{IJ} < U_{IJ}$
B2:	$\overline{C_{ij}} = 0$	$f_{IJ} > U_{IJ}$
K1:	$\overline{C_{ij}} < 0$	$f_{IJ} < U_{IJ}$
K2:	$\overline{C_{ij}} < 0$	$f_{IJ} > U_{IJ}$

Check if S is labelled, if labelled go to step 4. If not labelled, go to step 5.

4. Find a flow augmenting path from s to t if the selected arc is in states L1, B1, K1, or a flow augmenting path from t to s if the selected arc is in states L2, B2 or K2. If such a path can be found, augment the flow by the appropriate amount α. A is the minimum of the amount needed to bring arc(s,t) into kilter and the maximum allowable flow increase on the flow augmentation path.

$$\alpha = min[\alpha_s, L_{st} - f_{st}] \;\; if\ (s,t) is\ in\ state\ L1\ or\ B1$$

$$\alpha = \min[\alpha_s, U_{st} - f_{st}] \;\; if\ (s,t) is\ in\ state\ K1$$

$$\alpha = \min[\alpha_s, f_{st} - L_{st}] \;\; if\ (s,t) is\ in\ state\ L2$$

$$\alpha = \min[\alpha_s, f_{st} - U_{st}] \;\; if\ (s,t) is\ in\ state\ K2\ or\ B2$$

Go to step 2. If no such path exists, go to step 5 to revise the node potentials.

5. Apply the node potential change procedure. If Θ = ∞ stop. The problem has no feasible solution. If Θ ≠ ∞ update marginal costs. If arc(s,t) is now in kilter go to step 2, otherwise go to step 3 and continue labelling procedure using the old labels and the current marginal costs.

3.3 FLOWCHART FOR THE SOLUTION OF THE SCED PROBLEM USING GA

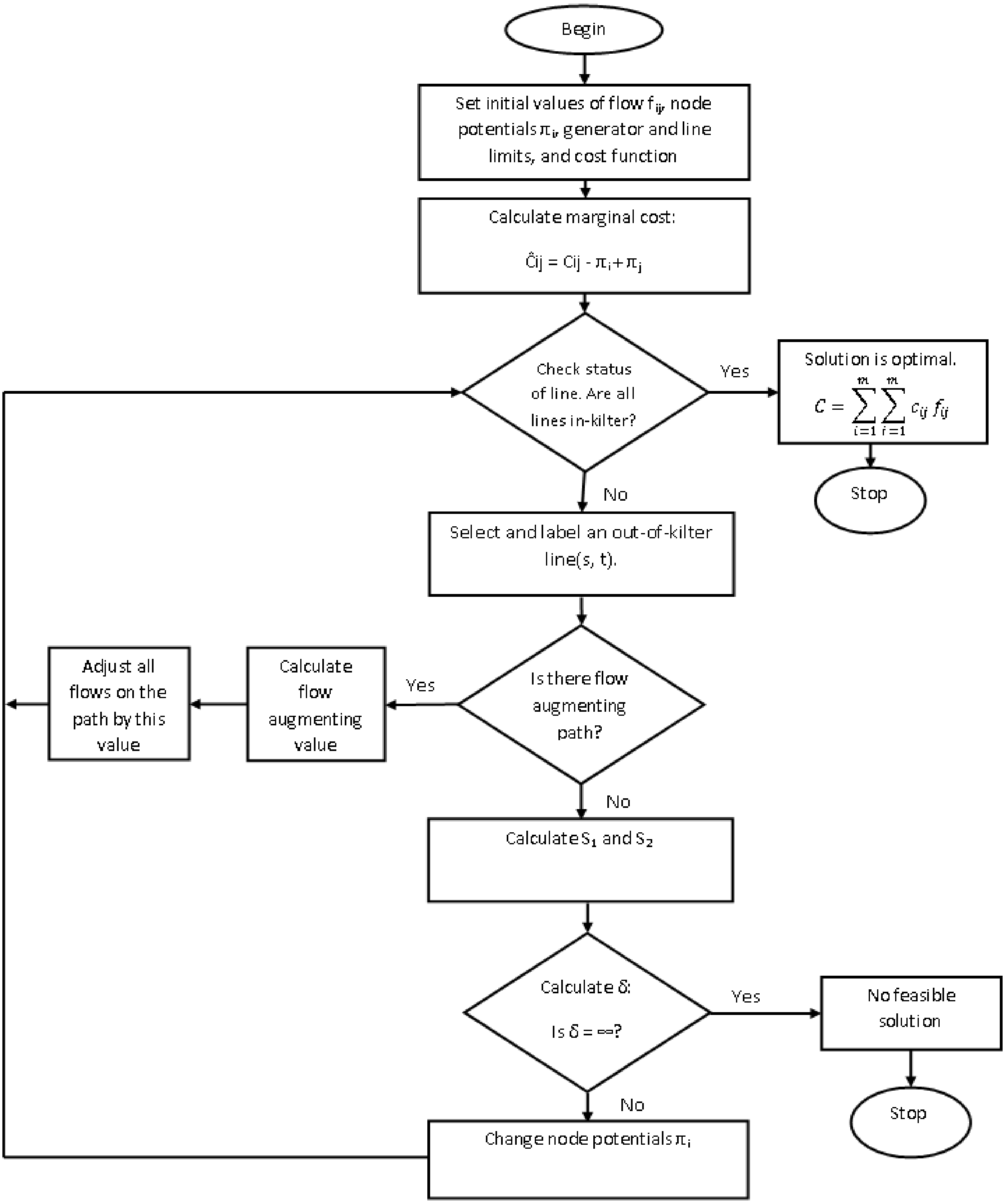

Figure 3. 1: Flowchart for the solution of the SCED problem using GA

CHAPTER 4

RESULTS AND ANALYSIS

The proposed algorithm was tested on IEEE-30 bus test system. The results were compared with those obtained from the Out-of-Kilter algorithm and Conventional Linear Programming and also compared with results from the classical economic dispatch neglecting the losses. The generator data, generation cost data, load data and line limits for the system are taken from [31] while the network topology is taken from [30].

4.1. CASE STUDY: IEEE 30 BUS NETWORK

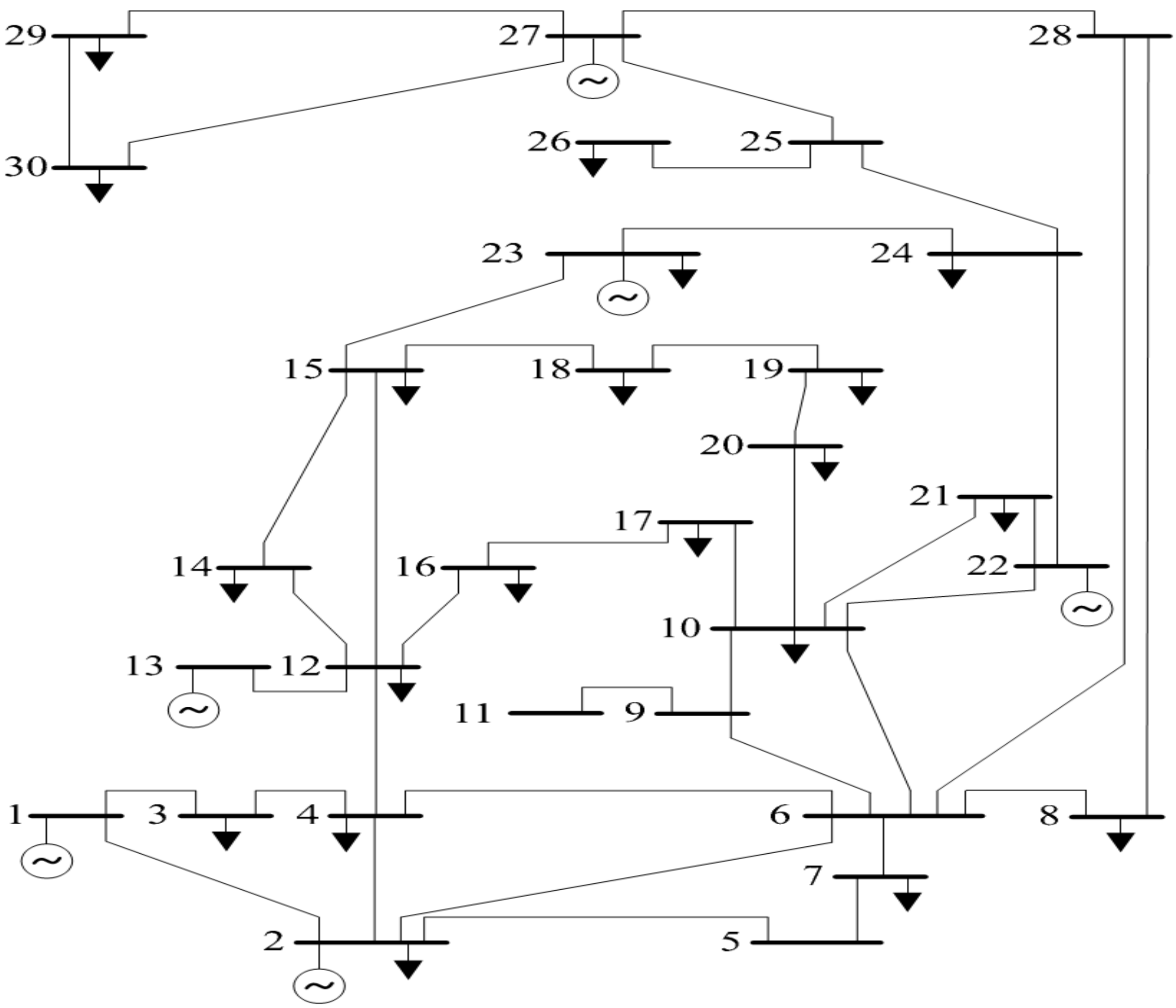

Figure 4.1: Single line diagram of the IEEE 30-bus test system

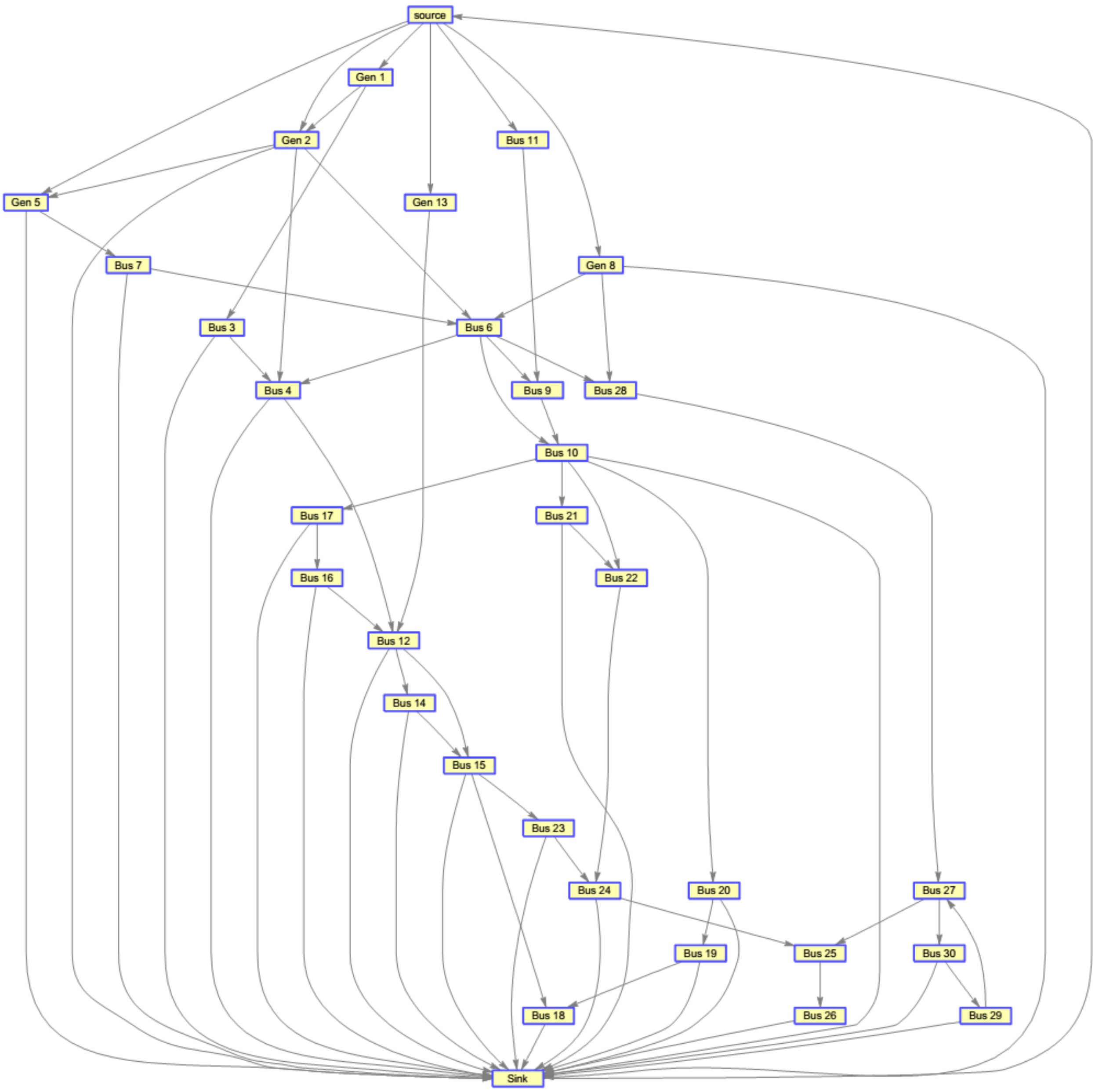

Figure 4.2: Representation of the IEEE 30-bus Network as an OKA Network Model in MATLAB

4.2. RESULTS

The optimal total generation, generation for the six generating units, the optimal generation costs and the system power losses using IOKA are shown in tables 4.1, 4.3, 4.4, 4.5 for both SCED and ED, for a system demand of 283.4 MW, 380 MW, 460 MW and 540 MW. Table 4.2 shows the comparison of the proposed IOKA method to those reported using OKA and LP from [31], for a total load demand of 283.4 MW.

Table 4.1: Optimal generation for SCED and Classical ED using IOKA, Demand = 283.4 MW

Generation No.	**SCED**	**ED**
PG1	178.347	175.54
PG2	49.0073	49.25
PG5	20.9276	21.63
PG8	21.9933	22.29
PG11	11.8439	12.63
PG13	10.9203	11.48
Total generation(MW)	293.0395	292.82
Total cost($/hr)	802.34	802.35
Total loss(MW)	9.6395	9.42

Table 4.2: Comparison of Economic Dispatch by IOKA, OKA and LP, Demand =283.4 MW

Generation No.	IOKA	OKA	LP
PG1	178.347	175.88	176.26
PG2	49.0073	48.81	48.84
PG5	20.9276	21.51	21.51
PG8	21.9933	22.36	22.15
PG11	11.8439	12.30	12.14
PG13	10.9203	12.00	12.00
Total generation(MW)	293.0395	292.86	292.9
Total cost($/hr)	802.34	802.51	802.4
Total loss(MW)	9.6395	9.46	9.395

Table 4.3: Optimal generation for SCED and Classical ED using IOKA, Demand = 380 MW

Generation No.	SCED	ED
PG1	199.6674	199.9781
PG2	63.7206	65.5647
PG5	25.3440	26.0815
PG8	56.8663	53.4134
PG11	22.9928	21.6505
PG13	23.8006	25.7982
Total generation(MW)	392.3917	392.4864
Total cost($/hr)	1176.31	1176.46
Total loss(MW)	12.3917	12.4864

Table 4.4: Optimal generation for SCED and Classical ED using IOKA, Demand = 460 MW

Generation No.	SCED	ED
PG1	199.6974	258.8656
PG2	82.0582	68.8267
PG5	28.6842	26.9822
PG8	84.8428	69.6569
PG11	34.4458	27.9912
PG13	46.6082	29.1942
Total generation(MW)	476.3367	481.5167
Total cost($/hr)	1603.11	1523.96
Total loss(MW)	16.3367	21.5167

Table 4. 5: Optimal generation for SCED and Classical ED using IOKA, Demand = 540 MW

Generation No.	SCED	ED
PG1	197.8287	296.7202
PG2	97.4788	77.3291
PG5	36.9553	29.8813
PG8	137.9802	90.8837
PG11	31.5865	36.0142
PG13	60.0670	40.1588
Total generation(MW)	561.8965	570.9874
Total cost($/hr)	2248.16	1914.45
Total loss(MW)	21.8965	30.9874

4.3. ANALYSIS AND DISCUSSIONS

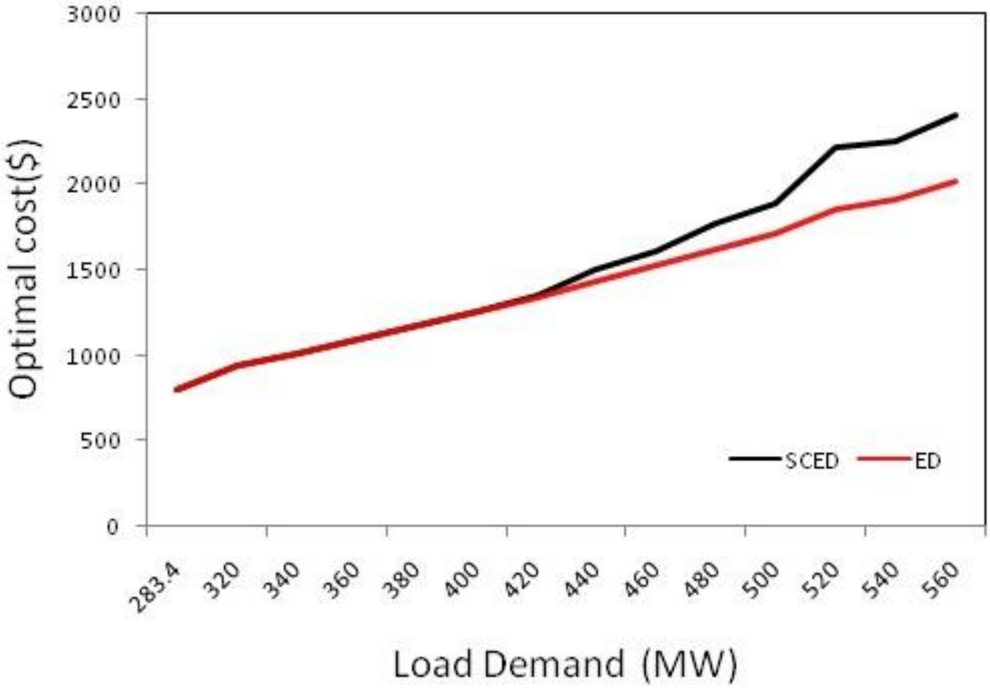

Figure 4.3: Variation of Optimal generation cost with total system demand for SCED and ED

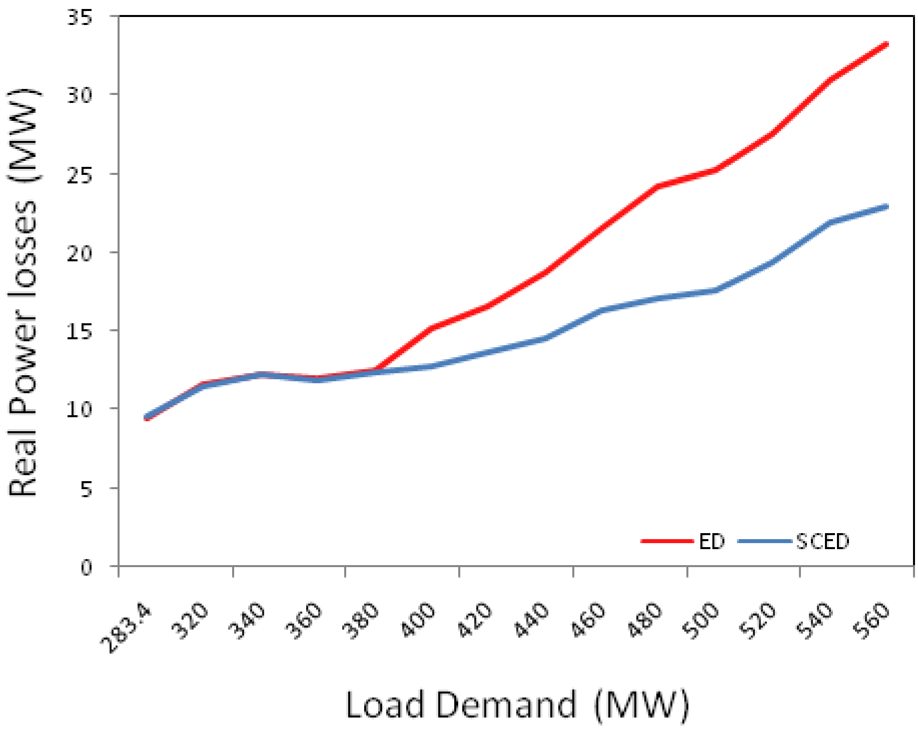

Figure 4.4: Variations of Real power loss with total system demand for SCED and ED

Figure 4.3 shows the variation of optimal cost with power demand for SCED. The cost of generation increases with increase in the load demand. The **Figure 4.3** also attests that the cost of generation with the security constraint considered (SCED) is higher than the cost of generation for the classical ED; this increased cost is due to the cost for ensuring the power system security.

In **Figure 4.4**, it is vital to observe that the total real power losses obtained with IOKA SCED are low as compared to those obtained from IOKA ED. this is more evident at high load demands of 400 MW and above and could be attributed to the fact that while performing SCED, the line thermal limits are maintained. This leads to reduced system losses compared to when the lines are carrying power that violates their permissible line power limits.

From **Table 4.2**, it can be seen that IOKA method give better results than other methods in terms of the total generation 293.0395 MW as compared to OKA = 292.86 MW and LP = 292.9 MW. This is also achieved at a slightly reduced total cost of 802.34 $/hr. This could account for a measurable saving in fuel cost hence better attainment of our objective function. This demonstrates the potential and effectiveness of the proposed method to solve the Security Constrained optimization problems.

In **Table 4.2**, by comparing the results obtained with IOKA algorithm and those realized by OKA and LP as cited in published works [31], the IOKA algorithm was able to achieve close to similar results as both methods.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1. CONCLUSION

In this project, IOKA algorithm was applied to solve the economic load dispatch problem with security constraints. The approach was tested on the IEEE 30-bus 6-generator network.
The IOKA SCED results were compared with those obtained from published works using Linear Programming and the Out-of-Kilter algorithm to validate the effectiveness of the proposed algorithm. The main security constraints considered are the generated active power as well as the active power flow limits of transmission lines.
Considering the cases and comparative study presented, IOKA algorithm appears to be very efficient in particular for its fast convergence to the global optimum and its slight low optimal cost of generation as compared to the LP and OKA method. This method is highly appropriate for network flow problems either in power systems or other systems e.g. the transportation problem.

5.2. RECOMMENDATIONS

1. This project considered only the cost of generation of power for the system. The project could be extended to also cover the cost incurred in transmission of the power through the network.
2. The project could also be broadened to perform N-1 security analysis which caters for cases of contingencies such as line outage in addition to the present operating condition of the system.
3. The method could be used to carry out SCED for other IEEE test buses i.e. 14-bus and 57-bus, in order to determine the effect of size of the network on the algorithm computation time in converging to the optimal solution.

REFERENCES

[1] "Security Constraint Economic Dispatch: Definition, Practices, Issues and Recomendations," Federal Energy Regulation Commision, New York, 2006.

[2] D. K. a. I. Nagrath, Modern Power Systems Analysis, New Delhi: Mcgraw Hill, 2003.

[3] J. Zhu, in *Optimization of power system operation*, New Jersey, Wiley-IEEE Press, 2009, pp. 1-407.

[4] M. S. S. Hossein Seifi, in *Electrical power System Plannig: Issues, algorithms and Solutions*, London, Springer, 2011, pp. 1-230.

[5] J. J. Stanislaw Sieniutycz, in *Energy Optimization in Process Systems*, Oxford, Elsevier, 2009, p. 44.

[6] C. K. Kulvinder Singh Handa, " Load Balancing With Genetic Algorithm," *International Journal of Emerging Technologies in Computational and Applied Sciences,* vol. 2, no. Aug-Nov, pp. 1-4, 2012.

[7] S. A.-H. M. a. A.-A. H. Soliman, Modern Optimizations Techniques with Applications in Electric Power Systems, New York: Springer, 2012.

[8] C. B. Marco Dorigo, "Theoretical Computer Science 344," Brussels, elsevier, 2005, pp. 243-278.

[9] M. B. N. M. M. D. Mark Zlochin, "Model-based search for combinatorial optimization: a critical survey," *Annals of Operations Research,* vol. 131, no. 1-4, p. 373–395, 2004.

[10] W.-S. L. Andreas Antoniou, in *Practical Optimization Algorithms and Engineering Application. Springer Science & Business Media*, Springer , 2007, pp. 321-450.

[11] K. Z. Samir Sayah, " Economic Load Dispatch with Security Constraints of the Algerian Power System using Successive Linear Programming Method," *Leonardo Journal of Sciences,* Vols. uly-December, no. 9, pp. 73-86, 2006.

[12] C. B. P. A. ,. C. L. J. A. G. Enrique Alba, in *Optimization Techniques for Solving Complex Problems,* New Jersey, Wiley, 2009, p. 267.

[13] S. S. K. V. D. Saxena, " Application of computational intelligence in emerging power systems," *International Journal Of Engineering, Science And Technology,* vol. 2, no. 3,

pp. 1-7, 2010.

[14] Q. Yang, " No Fit Polygon for Nesting Problem Solving with Hybridizing Ant Algorithms," *Journal of Software Engineering and Applications,* vol. 7, no. 1, pp. 433-439, 2014.

[15] G. D. C. L. M. G. Marco Dorigo, " Artificial Life," in *Ant Algorithms for Discrete Optimization*, 1999, pp. 130-170.

[16] J.-H. G. N.-N. L. X.-D. H. W.-L. Y. Yong-Feng Dong, " Combination of Genetic Algorithm And Ant Colony Algorithm For Distribution Network Planning. Machine Learning and Cybernetics," in *2007 International Conference on. 2 (1) p999 - 1002.*, 2007.

[17] A. J. W. a. B. F. Wollenberg, Power Generation, Operation and Control, New York: John Wiley, 1996.

[18] F. Glover, "Tabu Search," *ORSA journal on Computing*, vol. 1, no. 1, pp. 190-206, 1989.

[19] T. S. Marco Dorigo, in *Ant Colony Optimization*, Cambridge, MIT Press, 2004, pp. 49-60.

[20] C. D. G. A. M. V. S. Kirkpatrick, " Optimization by simulated annealing," vol. Science.220, no. 1, pp. 671-680, 1983.

[21] H. N. P. S. Akbar Karimi, "Computational Optimization and Applications," in *Continuous Ant Colony Search and Tabu Search Algorithms hybridized for global minimization of continuous multi-minima functions*, 2008, pp. 639-661.

[22] I. H. O. ,. T. S. Sam R. Thangiah, "Artificial Intelligence Review," in *Hybrid Genetic Algorithm, Simulated Annealing and Tabu Search Methods for Vehicle Routing Problems with Time Windows*, 1993, pp. 421-451.

[23] P. Kadar, "Applications of Optimization Technics in the power system control," *Acta polytechnica Hungarica,* vol. 10, no. 5, pp. 221-236, 2013.

[24] G. Gross, in *Power System Operations and Operations*, Illinois, Urbana-Champaign.

[25] F. G. D. K. R.S. Barr, An Improved Version Of The Out-Of-Kilter Method And A Comparative Study Of Computer Codes, . .

[26] G. S. Daniel S. Kirschen, in *Fundamentals of Power System Economics*, Chichester, Wiley, 2004, pp. 105-130.

[27] Z.-L. G. a. R.-F. Chang, "Security Constrained Economic scheduling of generation

considering generator constraints,” in *international Conference on power system*, 2006.

[28] E. P. D. a. D. M. Kroenke, The out-of-kilter algorithm: A Primer, Carlifonia: The Rand Corporation, 1967.

[29] D. Fulkerson, “An Out-of-Kilter Method for Solving Minimal Cost Flow Problems,” *Journal for the Society for Industrial and Applied Mathematics*, vol. 9, no. 1, pp. 18-27, 1961.

[30] L. Yaeger, *Intro to Genetic Algorithms*, lecture 3 1400/1590.

[31] j. Zhu, Power Systems Application of graph theory, Nova Science , 2009.

[32] M. D. a. T. Stutzle, in *Ant Colony Optimization*, Cambridge, MIT Press, 2004, pp. 49-60.

APPENDIX

Appendix Table 1: Generator data for IEEE 30-bus system [31]

Generators	#1	#2	#5	#8	#11	#13
P_{gimax}(p.u.)	2.00	0.80	0.50	0.35	0.30	0.40
P_{gimin}(p.u.)	0.50	0.20	0.15	0.10	0.10	0.12
Q_{gimax}(p.u.)	2.50	1.00	0.80	0.60	0.50	0.60
Q_{gimin}(p.u.)	-0.20	-0.20	-0.15	-0.15	-0.10	-0.15
Quadratic cost function						
a_i	0.00375	0.0175	0.0625	0.0083	0.0250	0.0250
b_i	2.00000	1.7500	1.0000	3.2500	3.0000	3.0000
c_i	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Appendix Table 2: Load data for IEEE 30-bus system [31]

Bus no.	**P_D(p.u)**	**Q_D(p.u)**	**Bus no.**	**P_D(p.u)**	**Q_D(p.u)**
1	0.000	0.000	16	0.035	0.016
2	0.217	0.127	17	0.090	0.058
3	0.024	0.012	18	0.032	0.009
4	0.076	0.016	19	0.095	0.034
5	0.942	0.190	20	0.022	0.007
6	0.000	0.000	21	0.175	0.112
7	0.228	0.109	22	0.000	0.000
8	0.300	0.300	23	0.032	0.016
9	0.000	0.000	24	0.087	0.067
10	0.058	0.020	25	0.000	0.000
11	0.000	0.000	26	0.035	0.023
12	0.112	0.075	27	0.000	0.000
13	0.000	0.000	28	0.000	0.000
14	0.062	0.016	29	0.024	0.009
15	0.082	0.025	30	0.106	0.019

Appendix Table 3: Line flow limits data for IEEE 30-bus system [31]

Line No.	From Bus	To Bus	Flow limit(MW)	Annual Cost (K$/year)	Line No.	From Bus	To Bus	Flow limit(MW)	Annual Cost (K$/year)
1	1	2	130	216.6125	22	15	18	16	80.6000
2	1	3	130	307.2875	23	18	19	16	235.6000
3	2	4	65	509.9500	24	19	20	32	186.0000
4	3	4	130	700.0000	25	10	20	32	117.8000
5	2	5	130	721.5250	26	10	17	32	167.4000
6	2	6	65	168.1750	27	10	21	32	160.4250
7	4	6	90	474.3000	28	10	22	32	195.3000
8	5	7	70	62.0000	29	21	22	32	166.2375
9	6	7	130	130.2000	30	15	23	16	100.7500
10	6	8	32	104.6250	31	22	24	16	40.3000
11	6	9	65	306.9000	32	23	24	16	65.1000
12	6	10	32	20.9250	33	24	25	16	210.8000
13	9	11	65	83.7000	34	25	26	16	204.6000
14	9	10	65	927.6750	35	25	27	16	83.7000
15	4	12	65	554.1250	36	28	27	65	223.2000
16	12	13	65	15.1125	37	27	29	16	160.4250
17	12	14	32	30.2250	38	27	30	16	90.6750
18	12	15	32	97.6500	39	29	30	16	216.6125
19	12	16	32	179.0250	40	8	28	32	54.2500
20	14	15	16	124.7750	41	6	28	32	210.8000
21	16	17	16	146.4750					

PROGRAM LISTING

```
% Improved Out-of-Kilter Algorithm solution for SCED
clear
clc

%POWER FLOW ANALYSIS (NR)
nrpflow = runpf('case_ieee30');
clc
branch(41,3) =0;
power = [nrpflow.branch(:,1) nrpflow.branch(:,2) nrpflow.branch(:,14)];
for j =1:2
for i = 1:41
branch(i,j) = power(i,j) +1;
end
end
powerflow = [branch(:,1) branch(:,2) power(:,3)];

%declare required variables
mpc = case_ieee30b;
node_potentials(32,32)=0; %node potentials
min_capacity(32,32)=0; %minimum  line capacities
max_capacity(32,32)=0;  %maximum line capacities
line_cost(32,32)=0; %line transmission costs
initial_flow(32,32)=0; %initial line flows

%set the initial values
for i = 1:32
for j = 1:32
for l = 1:69
if((i == mpc.branch(l,1)) && ((j == mpc.branch(l,2))))
node_potentials(i,j) = 0;
min_capacity(i,j) = mpc.branch(l,3);
max_capacity(i,j) = mpc.branch(l,4);
line_cost(i,j) = mpc.branch(l,5);
initial_flow(i,j) = mpc.branch(l,6);
end
end
end
end

%initialise initial flow from power flow by NR
for i=1:41
for j=1
if(initial_flow(powerflow(i,j),powerflow(i,j+1))==1)
   if(powerflow(i,j+2) > 0)
   initial_flow(powerflow(i,j),powerflow(i,j+1)) = powerflow(i,j+2);
   end
end
end
end
gen = [0 260 40 0 0 20 0 0 23 0 0 67 0 45];
   for j=1:14
initial_flow(1,j) = gen(1,j);
   end
```

```
%GENERATOR LIMITS
pgmax = [200 80 0 0 50 0 0 35 0 0 30 0 40];
pgmin = [50 20 0 0 15 0 0 10 0 0 10 0 12];
for i = 1:13
max_capacity(1,i+1) = pgmax(1,i);
min_capacity(1,i+1) = pgmin(1,i);
end

%CHANGE LOAD DEMAND
loaddemand = [0 0 288.3 2.4 7.6 94.2 0 22.8 30 0 5.8 0 11.2 0 6.2 8.2 3.5 9 3.2 9.5 2.2 17.5 0 3.2 8.7
0 3.5 0 0 2.4 10.6 0];
for i = 1:32
min_capacity(i,32) = loaddemand(1,i);
max_capacity(i,32) = loaddemand(1,i);
min_capacity(32,1) = sum(loaddemand);
max_capacity(32,1) = sum(loaddemand);
end

  flow = initial_flow;
  for iteration = 1:200 % maximum number of iterations, Stopping criteria
    %CALCULATE MARGINAL COST
    marg_cost = marginal_cost(line_cost,node_potentials);

    %CHECK LINES KILTER STATUS AND LABEL AN OUT OF KILTER LINE
    [kilter, s, t] = kilter_status(max_capacity,min_capacity,flow,marg_cost);

    %FLOW OPTIMALITY CHECK
    [~, no_nodes] = size(flow);
    if (s == 0 && t == 0) %Stopping criteria
      %If optimal,break to end iteration and calculate total generation and fuel cost at the end
    break
    else %If not optimal, find flow augmenting path
    [path, labelled_nodes] = bfs_augmentingpath(s,t,flow,max_capacity,no_nodes);
    end

    %FLOW AUGMENTING PATH CHECK
    pathsize = max(size(path));
    if ~isempty(path)%flow augmenting path is present
    flow = ff_updated_flow(s,t,flow, max_capacity,no_nodes); %calculate new flows
    else %flow augmenting path is absent

    %INCREMENTAL VERTEX COST CALCULATION
    [b labellednode_potentials] = calc_theta(marg_cost,node_potentials, flow,
max_capacity,min_capacity, labelled_nodes);
      if b ~= inf %feasible incremental vertex cost
      node_potentials = node_price(node_potentials, labellednode_potentials,b); %change node
potentials
      else
      disp('THE SOLUTION IS INFEASIBLE'); %Stopping criteria, flow is infeasible
      break
      end
    end
  end
```

```
  %FUEL COST EVALUATION
   fcost = generatorfuelcost(flow);
   fuelcost = strcat('TOTAL GENERATION COSTS =', num2str(fcost), ' $');
   %generation of each generator
   PG1 = strcat('PG1 =',num2str(flow(1,2)),' MW');
   PG2 = strcat('PG2 =',num2str(flow(1,3)),' MW');
   PG5 = strcat('PG5 =',num2str(flow(1,6)),' MW');
   PG8 = strcat('PG8 =',num2str(flow(1,9)),' MW');
   PG11 = strcat('PG11 =',num2str(flow(1,12)),' MW');
   PG13 = strcat('PG13 =',num2str(flow(1,14)),' MW');
   %total load demand
   load_demand = strcat('TOTAL LOAD DEMAND =', num2str(sum(loaddemand)), ' MW');
   %total generation
   tgen = flow(1,2)+flow(1,3)+flow(1,6)+flow(1,9)+flow(1,12)+flow(1,14);
   tgeneration = strcat('TOTAL POWER GENERATION =', num2str(tgen), ' MW');%concatenation
   %total real power losses
   tloss = tgen-flow(32,1);
   tlosses = strcat('TOTAL POWER LOSSES =', num2str(tloss), ' MW');
   %dispay results
   disp(flow);
   disp('OPTIMAL GENERATION');
   disp(load_demand);
   disp(tgeneration);
   disp(tlosses);
   disp(fuelcost);fprintf('\n');
   disp(PG1);
   disp(PG2);
   disp(PG5);
   disp(PG8);
   disp(PG11);
   disp(PG13);

%MARGINAL COST FUNCTION
function marg_cost = marginal_cost(line_cost,node_potentials)
%get the rows and columns matrices
 [rows1, cols1] = size(line_cost);
 %calculate marginal cost
 marg_cost(rows1, cols1) = 0;
  for i = 1:rows1
     for j = 1:cols1
         %check for existence of the line before calculating marginal cost
         if line_cost(i,j) ==0
             marg_cost(i,j) = 0;
         else
         marg_cost(i,j) = line_cost(i,j) - node_potentials(i,i)+ node_potentials(j,j);
         end
     end
  end
 end

%CHECK KILTER STATUS FUNCTION
function [kilter, s, t] = kilter_status(max_capacity,min_capacity,flow,marg_cost)
%check kilter status of the lines
[rows1, cols1] = size(flow);
kilter(rows1,cols1) = 0;
```

```
for m = 1:rows1
   for n = 1:cols1
    %definition of variables
      mc = marg_cost(m,n); %set x to represent marginal cost of line
      f = flow(m,n);  %set y to represent flow on line
      l = min_capacity(m,n);%set l to represent minimum capacity of line
      u = max_capacity(m,n);%set U to represent maximum capacity of line
      if ((mc > 0 && f < l) || (mc > 0 && f > l))
          kilter(m,n) = 0; %let 0 represent an out-of-kilter arc
      elseif ((mc == 0 && f < l) || (mc == 0 && f > u))
          kilter(m,n) = 0; %let 0 represent an out-of-kilter arc
      elseif ((mc < 0 && f < u) || (mc < 0 && f > u))
          kilter(m,n) = 0; %let 0 represent an out-of-kilter arc
          else
          kilter(m,n) = 1; %let 1 represent an in-kilter arc
      end
      if (m == n)
          kilter(m,n) = 1; %set nodes as in-kilter by default
      end
    end
end
%label out-of-kilter line
s = 0;
t = 0;
for i = 1:rows1
    for j = 1:cols1
        if kilter(i,j) ==0  %check if any line violates its limits(is out-of-kilter), label the out-of-kilter line
             if i ~= j
                 s = i;
                 t = j;
             else
                 disp('Error in labelling out-of-kilter line');
             end
         break
       else
           %empty statement, ensures the next line is checked
          end
     end
end
end

%CHECK FLOW AUGMENTING PATH FUNCTION
function [augmentingpath, labelled_nodes] = bfs_augmentingpath(start,target,initial_flow,capacity,n)
    WHITE =0; %shows the path has not been traversed
    GRAY=1;
    BLACK=2; %shows the path has not been traversed
    color(1:n)=WHITE; %shows if all the paths in the network have been checked
    head=1;
    tail=1;
    q=[];
    augmentingpath=[];
    labelled_nodes=[];
    %ENQUEUE
    q=[start q];
    color(start)=GRAY;
```

```
    pred(start) = -1;
    pred=zeros(1,n);
    while ~isempty (q)
    %    [u,q]=dequeue(q);
            u=q(end); % u controls the rows of the matrices
            q(end)=[];
            color(u)=BLACK;
    %     dequeue end here
            for v=1:n % v controls the columns of the matrices
                if (color(v)==WHITE && capacity(u,v)>initial_flow(u,v) ) %check if the line has been
treversed and if the flow is less than line capacity enqueue(v,q)
                    q=[v q];
                    color(v)=GRAY;
    % enqueue end here
                    pred(v)=u;
                end
            end
    end
if color(target)==BLACK %confirm if there is a path from source to sink
       temp=target;
       %set the augmenting path from source to sink
       while pred(temp)~=start
        augmentingpath = [pred(temp) augmentingpath];
        labelled_nodes = [pred(temp) augmentingpath];
        temp=pred(temp);
       end
       augmentingpath=[start augmentingpath target];
    else
        augmentingpath=[]; % set augmenting path to empty if there is no path from source to sink
    end

%UPDATE FLOW FUNCTION
function updated_flow =ff_updated_flow(s,t,flow,capacity,no_nodes) %you can remove the output to
prevent multiple
%outputs of final value while 'choose_arc' is checking line violations
updated_flow=0; %initialise updated_flow
%call function bfs_augmentingpath to look for a flow augmenting value
augmentingpath = bfs_augmentingpath(s,t,flow,capacity,no_nodes);
%check if a flow augmenting path has been returned
while ~isempty(augmentingpath)
    increment = inf;
    for i=1:length(augmentingpath)-1
    %get the flow augmenting value
        increment=min(increment,                    capacity(augmentingpath(i),augmentingpath(i+1))-
flow(augmentingpath(i),augmentingpath(i+1)));
    end
    %flow exists,increase the flow
    for                                                   i=1:length(augmentingpath)-1
flow(augmentingpath(i),augmentingpath(i+1))=flow(augmentingpath(i),augmentingpath(i+1))+incre
ment;%increase              flow             on              forward              arc
flow(augmentingpath(i+1),augmentingpath(i))=flow(augmentingpath(i+1),augmentingpath(i))-
increment; %decrease flow on reverse arc
    end
    updated_flow =updated_flow+increment; %update the initial flows with the incremented values
    updated_flow = nflow(s,t,flow,capacity,no_nodes);%recalculate new flow
```

```
    augmentingpath = bfs_augmentingpath(s,t,flow,capacity,no_nodes); %find new flow augmenting
path after every update
end

%INCREMENTAL VERTEX COST FUNCTION
function [b labellednode_potentials] = calc_theta(marg_cost,node_potentials, flow,
max_capacity,min_capacity, labelled_nodes)
[rows1, cols1] = size(node_potentials);
S1 = [];
S2 = [];
labellednode_potentials(rows1, cols1) = 0;
[rows2, cols2] = size(labelled_nodes);
%store the node potentials of the labelled nodes
while cols2 ~= rows2
    ro = labelled_nodes(1,cols2);
    kol = labelled_nodes(1,cols2);
        labellednode_potentials(ro,kol) = node_potentials(ro,kol);
        cols2 = cols2-1;
end
%create two sets S1 and S2
for i = 1:rows1
   for j = 1:cols1
        if (marg_cost(i,j) > 0 && flow(i,j) <= max_capacity(i,j))  %vector A1
       S1(i,j) = marg_cost(i,j);
       elseif(marg_cost(i,j) < 0 && flow(i,j) >= min_capacity(i,j))
       S2(i,j) = marg_cost(i,j);
     else
    end
   end
%get the incremental vertex cost
if isempty(S1)
    b1 = inf;
else
   b1 = min(S1(S1>0));
end
if isempty(S2)
    b2 = inf;
else
   b2 = min(S2(S2>0));
end
theta = min(min(b1, b2));
b(1:rows1,1:cols1) = theta;
end

%UPDATE NODE PRICE FUNCTION
function node_potentials = node_price(node_potentials, labellednode_potentials,b)
[rows1, cols1] = size(node_potentials);
%calculate the new node potentials of all the nodes
for i = 1:rows1
        for j = 1:cols1
        node_potentials(i,j) = node_potentials(i,j) + b(i,j);
        end
end
 %keep the node potentials of the labelled nodes unchanged
for i = 1:rows1
```

```
    for j = 1:cols1
        if labellednode_potentials(i,j) ~= 0
        node_potentials(i,j) = labellednode_potentials(i,j);
        end
    end
end
 end

%COST EVALUATION FUNCTION
function fuelcost = generatorfuelcost(flow)
fuelcost = 0; %initialize fuel cost variable
%set generator power outputs from optimal solution
if size(flow)==32
    P = [flow(1,2) flow(1,3) flow(1,6) flow(1,9) flow(1,12) flow(1,14)];
elseif size(flow)==16
    P = [flow(1,2) flow(1,3) flow(1,4)];
end
%IEEE 30 bus generator cost coefficients
% a      b       c
abc = [
0.00375 2.0000 0.000
0.01750 1.7500 0.000
0.06250 1.0000 0.000
0.00830 3.2500 0.000
0.02500 3.0000 0.000
0.02500 3.0000 0.000
];
%CALCULATE FUEL COST
[~,n] = size(P);
   for i = 1:n
    fuelcost=fuelcost +(abc(i,1)*P(1,i)*P(1,i)+abc(i,2)*P(1,i)+abc(i,3));
    end
  end
```