

NETWORK FLOW METHOD FOR POWER SYSTEM ANALYSIS

by

Najib.H. Dandachi, B.Sc.(Eng). M.Sc.(Eng)

A thesis submitted for the degree of Doctor of
Philosophy in Engineering of the University of
London and for the Diploma of Membership of the
Imperial College

Department of Electrical Engineering
Imperial College of Science, Technology and Medicine
University of London
London SW7 2AZ
August, 1989

ABSTRACT

The Network Flow (NF) methods are known for their simplicity, rapid computation and have many potential uses when applied to problems of a flow transportation nature. The interest in studying NF methods was stimulated by the surprising range of power system problems that can be cast into a network structure of nodes and branches; these include Load Flow calculations, active power economic dispatch, contingency analysis, reliability analysis, etc.

The network flow method we have adopted is the Out-Of-Kilter Algorithm (OKA) which is an optimization method to minimize an objective function subject to certain constraints. It is the most general of a family of algorithms that solve static single commodity network flow problems.

This research has proved that NF methods, in general, are capable of solving operational power system problems such as active power economic scheduling, contingency analysis and load flow calculations in a fast and robust manner, where in many cases their solutions are superior to those using Linear Programming. Applications to on-line real time use and for operational planning are recommended and guidelines for further investigation are suggested.

ACKNOWLEDGEMENT

My sincere gratitude and appreciation are to, my supervisor, Dr. B.J. Cory, D.sc(Eng), CEng, FIEE, Sen. Mem. IEEE, Head of Energy & Power Systems Group (EPSG) and industrial Liaison, for his guidance, support and encouragement throughout the course of this Ph.D research project.

A special acknowledgement is due to Mr. E.D. Farmer for the useful discussions and ideas. I also would like to express my appreciation to my colleagues in the EPSG for their contribution towards making the time spent at IC friendly and enjoyable. Another acknowledgement is to Mr. M.W. Rowe for his continuous help with the computing facilities.

Last but not least, I am indebted to Hariri Foundation (Lebanon) for the financial support provided and the Committee of Vice Chancellors and Principals of Universities of United Kingdom for having awarded me the ORS-award.

And he makes comparisons for Us and forgets his own creation he says who can give life to dry bones and decomposed ones (78) Say He will give them Life who created them for the first time For He is well versed in every kind of creation (79)

THE HOLY QUR'AN
Chapter 36

To my mother, my father
and dearest wife Sahar

CONTENTS

	page
ABSTRACT	2
ACKNOWLEDGMENT	3
DEDICATION	4
CONTENTS	5
LIST OF SYMBOLS	9
LIST OF FIGURES	11
LIST OF TABLES	13
CHAPTER I: INTRODUCTION	15
1.1 Why the Network Flow ?	17
1.2 Thesis contents	19
1.2.1 The Out-Of-Kilter method	19
1.2.2 Power system applications of the OKA	20
a. Active power economic scheduling	20
b. Load flow calculations	21
c. Reliability analysis	22
d. Other NF applications	23
1.2.3 Notes for Future uses	24
CHAPTER II: THE THEORY OF THE OUT-OF-KILTER ALGORITHM	25
2.1 Introduction	25
2.2 Basic terminology	26
2.3 Network flow	29
2.4 The Out-Of-Kilter algorithm	32
2.4.1 Optimality conditions	33

2.4.2	Arc states	35
2.4.3	The labeling procedure	37
2.4.4	Flow change	40
2.4.5	Node price increase	41
2.5	A graphical interpretation of the OKA	41
2.6	Algorithmic steps	45
2.7	Example solution	47
2.8	Improved versions of the OKA	55
2.9	Summary and conclusion	57
CHAPTER III: ACTIVE POWER ECONOMIC SCHEDULING		59
3.1	Introduction	59
3.2	Economic scheduling	61
3.2.1	Transmission line model	62
3.2.2	Generating plant model	64
3.2.3	Demands and imports	65
3.3	Computer implementation	67
3.4	Experimental results and discussion.	67
3.5	Consideration of Kirchoff's voltage law	74
3.6	Further discussion and comments	82
CHAPTER IV: THE LOAD FLOW CALCULATION		84
4.1	Introduction	84
4.2	The load flow problem	85
4.3	The load flow solution	88
4.3.1	The DC load flow	88
4.3.2	The NF as a DC load flow	90
4.3.3	Contingency analysis	101
4.3.4	The AC load flow	104

4.3.5	The NF as an AC load flow	. 105
4.4	Alternative AC solution by NF methods	 115
4.5	Conclusion	 116
CHAPTER V:	RELIABILITY ANALYSIS	 118
5.1	Historical background	 118
5.2	Basic probability	 120
5.3	Probabilistic reliability assessment	 122
5.3.1	State classification	 124
5.3.2	Defining the upper and lower critical states	 125
5.3.3	Evaluation of LOLP	 129
5.3.4	Exhausting the Unclassified subset $\mathcal{U}$	 131
5.4	Results and discussion	 137
5.5	Conclusion	 143
CHAPTER VI:	OTHER POTENTIAL NETWORK FLOW APPLICATIONS	 145
6.1	Introduction	 145
6.2	Operational planning of hydro thermal power system	 146
6.3	Short term operational planning of large hydro thermal system	 147
6.4	Group transfer method	 149
6.5	Emergency rescheduling	 150
6.6	Power interchange scheduling	 151
6.7	Security constrained economic dispatch	 152

6.8	Optimal corrective switching	154
6.9	Transmission system expansion	155
6.10	Conclusion	156
CHAPTER VII: CONCLUSIONS		157
7.1	Thesis contribution	157
7.2	Improvement in labeling sequence	159
7.3	Guidelines for future uses	163
7.4	Conclusion	165
APPENDICES		
APPENDIX A: COMPUTER IMPLEMENTATION		
	OF THE OKA	167
APPENDIX B: THE OKA SOURCE PROGRAM		183
APPENDIX C: SATISFACTION OF KIRCHOFF'S		
	VOLTAGE LAW	196
REFERENCES		198

LIST OF SYMBOLS

A	Node-Arc incidence matrix
$\mathcal{A}$	Acceptable subset
A	Set of arcs in a network
b_T	No load generating cost
c	Cost of shipping one unit of flow along an arc
$\bar{C}_{ij}$	Optimizing cost
δ	Permissible node price increase
Δf	Flow correction
EDNS	Expected demand not served
ϵ_i	Permissible flow increase up to node i
ϵ_t	Permissible flow increase along the path found
ϵ_k	Capacity for flow change of the scanned arc
F_i	Generation cost at bus i
FOR	Forced outage rate
f_k	Flow through arc k
φ_n	Real power injection
G	A graph
g_k	Real power generation of generator k
H	Path label
h	System marginal cost
J_{12}	Imaginary component of current from bus 1 to 2
k	An arc or branch connecting two nodes (i,j)
L_{en}	Line length in km
LOLP	Loss of load probability
L_k	Lower bound on arc k
L_t	Transpose of branch to loop incidence matrix

λ_k	Incremental cost
M	Set of labeled nodes
$\bar{M}$	Set of unlabeled nodes
μ, σ	Dual multipliers
η	Expected failure rate
v_m	State of element m
P_D	Real power demand
P_{Gi}	Real power generation at bus i
P_L	System real losses
P_{fl}	Full load real power rating
p	Probability of success
q	Probability of failure
$\mathcal{R}$	Rejected subset
$\mathcal{S}$	Sink node
s	Source node
τ	Imaginary current injection
$\mathcal{U}$	Unclassified subset
$\bar{\xi}$	System upper limiting state
$\underline{\xi}$	System lower limiting state
ξ^o	System upper critical state
ξ_o	System lower critical state
ξ_i	System capacity state
χ	Expected success rate
U_k	Upper bound on arc k

LIST OF FIGURES

Fig.		page
2.1	A directed arc	28
2.2	A simple network	28
2.3	Network with multi sources & sinks	28
2.4	A path along a network	30
2.5	The return arc from sink to source	30
2.6	Mnemonic device for the OKA	43
2.7a	[Valid horizontal moves	43
2.7b		
2.7c	[Valid vertical moves	43
2.7d		
2.8	The flow chart of the OKA	46
2.9	Capacitated network	48
2.10	Closed capacitated network	48
2.11	The resulting flows	56
3.1	Piecewise linearization	63
3.2	Real power flow along a line	63
3.3	Demand and import models	66
3.4	The CEGB 23-bus system	68
3.5	The IEEE 24-bus system	78
4.1	A two bus system	87
4.2	A small example system	87
4.3	First stage line flows	91
4.4	The independent loops	91
4.5	Final flow distribution	91
4.6	New mnemonic device for the OKA	94
4.7	New valid displacements	94
4.8	The flowchart for the DC load flow	96

4.9	The IEEE 14-bus system	97
4.10	The flowchart for the AC load flow	109
4.11	Equivalent π circuit	111
4.12	Real power flows for the 9-bus system	113
5.1	Probability scale	121
5.2	Pictorial representation of two state model	123
5.3	A three bus system	123
5.4	Venn diagram representing $\mathcal{A}$, $\mathcal{R}$ and $\mathcal{U}$	126
5.5	Decomposition of $\mathcal{R}$	126
5.6	A pictorial illustration of $\mathcal{U}$ boundary's	132
5.7a 5.7b	[Exhausting subset $\mathcal{U}$	133
5.8	Flowchart of the Reliability analysis	134
5.9	Flowchart of subroutine RELIABILITY	135
5.10	Flowchart of subroutine CALCPROB	136
5.11	The WSCC9 test system	138
6.1	Benders' method for operational planning	148
7.1a	The flow augmentation path	161
7.1b	The path of the second call	161
7.2a	The path for different arc	162
7.2b	The path of the second call	162
A.1	KILTAL flowchart	169
A.2	NODARC flowchart	171
A.3	CPIPI flowchart	173
A.4	FINDST flowchart	175
A.5	LABELR flowchart	177
A.6	FLOCHA flowchart	179
A.7	NODPRI flowchart	181

LIST OF TABLES

Table	page
2.1 Optimality conditions	36
2.2 Possible conditions for arcs	44
2.3 The flow change needed to put arc in-kilter	44
3.1 Transformer and series reactor data and power flows	71
3.2 The demand data	71
3.3 Line data and power flows	72
3.4 Generator data and output	73
3.5 Final real power line flows	76
3.6 Transformer and series reactor final power flows	77
3.7 Bus generation and demand for the IEEE 24-bus system	79
3.8 IEEE 24-bus line data and power flows	80
4.1 The correspondence between the NF and the DC method	95
4.2 Real power flows for the 24-bus system	98
4.3 The load shedding caused by line outages	103
4.4 The correspondence between the NF and the AC method	110
4.5 The converged Voltage magnitude and phase angles	114
5.1a Transmission lines data	139

5.1b	Bus generation data	139
5.1c	Demand data	139
5.2	The active power flow distribution	140
5.3	The unclassified subsets $\mathcal{U}_j$'s	141

CHAPTER I

INTRODUCTION

The prime objective of an electric power system is to meet the consumer demand connected to the system. The operating and expansion planning strategies then aim at satisfying the load in full while maintaining a high reliability of electric power service. At the same time, the division of the load between the generating sets should be dispatched such that for a particular load pattern the total production cost is always minimized (subject to certain constraints). Thus, the overall function of an electric power system is to allocate optimally the demand to the power sources, taking into consideration the security aspects of generation/transmission and ensuring the quality and continuity of the electric power supply.

This large complex problem, which is spread over a wide time scale, is best decomposed into interrelated sub-problems where each has its own mathematical model and solution methodology and can be solved separately. Because of the ever increasing size of power system networks it is almost impossible to develop a technique capable of solving the whole problem; usually the subproblems are defined on the basis of their time scales [1]:

- * short term operational planning, one day to one week
- * seasonal operational planning, 12 months
- * long term operational planning, 2 to 5 years

The use of digital computers in power system studies has enabled large complex problems to be analyzed more effectively and economically. Their use, whether for off-line or on-line applications [2], is now widespread in all aspects of planning, operation and control of electric power systems [3].

The rapid evolution in computer technology has introduced new scopes to the job of the system engineer, as it becomes more feasible and economical to use digital computers for the continuous evaluation of the system performance. Also, the exhaustive amount of data and the extensive necessary computations are better handled by microprocessors. On the other hand, the continuous development of fast processors is opening the way towards a high level secure and optimal automatic operation of power systems. Although power utilities are still reluctant to implement it, the theoretical concepts and appropriate basic methods exist [4]. For off-line applications, the system reliability is greatly enhanced as computers made possible the study and assessment of greater variety of operating states.

However, all this calls for the development of appropriate mathematical tools and algorithms that possess a combination of features necessary to tackle a particular problem.

1.1 WHY THE NETWORK FLOW ?

In modern control centers, real time network analysis functions are available for the operator to make more informed decisions about the security and economy of the power system. These functions include contingency analysis, security constrained dispatching to produce power economically and optimal reactive scheduling to support the voltage profile. Thus it is essential, as far as the operator is concerned, to have available a library of algorithms that can be called upon for study purposes. Obviously these algorithms should be selected on the basis of their ability to satisfy defined requirements, such as accuracy, computational efficiency, reliability, memory, practical implementation, etc.

The Network Flow (NF) methods are known for their simplicity and speed of computation. They have many potential uses when applied to problems of a flow transportation nature. The interest in studying NF methods was stimulated by the surprising range of power system problems that can be cast into a network structure of nodes and branches. On the other hand, NF solutions have proved to be superior to those using Linear Programming, for some particular uses (up to 100 times faster [5]). They possess the accuracy, the reliability and the speed potentials required for on-line applications.

The network flow algorithm used in this work is the Out-Of-Kilter Algorithm (OKA) which is an optimization method that minimizes an objective function subject to constraints [6]. Its optimization procedure is monotonic i.e. forward moving toward optimality. The OKA was developed specifically for network programming and Kennington & Helgason [5] believe it is unique in the mathematical programming literature. Although several computational comparisons of implementations have been made with the LP primal simplex algorithm, there is still disagreement as to which one is superior. The latest reported essay, in a series of attempts [7-9] to improve the OKA implementation, is given by Singh [10] who claims to have developed a faster OKA coding with less memory requirements, compared to a previously reported code by Barr et al [8]. (The theory of the OKA is presented in the next chapter).

The major purposes this work is trying to fulfill are to:

- 1- propose some potential applications of NF methods to some planning and operational power system problems.
- 2- perform a detailed analysis of selected objectives and to assess the merits of the NF solutions when compared to conventional existing methods.
- 3- present some guidelines for further investigation and uses of NF algorithms.

1.2 THESIS CONTENTS

As mentioned earlier, the primary objective of this research work is to study thoroughly the potentials of the network flow methods and to highlight their merits. Before proceeding to the application of NF, their underlying theory is presented to provide a common ground for understanding. Thus the thesis is segmented into three main parts: the theory of the Out-Of-Kilter Algorithm (OKA), the power system applications of the OKA and the recommendations (in the author's opinion) for further investigation.

1.2.1 The out-of-kilter method

To familiarize the reader with the concept of network flow, it is natural to present some basic notations and conventions relevant to graph theory and transportation algorithms before proceeding to the underlying theory of the OKA. Such basic conventions include definitions of *network*, *arcs*, *path*, *circulation*, etc.

The OKA, a primal-dual algorithm, was developed for network optimization purposes such as distribution, production scheduling, resource allocation and many other important applications. It is a linear network flow model i.e. a special case of linear program. Its mathematical form is stated and the optimality conditions [11], which terminate the OKA when satisfied, are illustrated with the aid of graphs and tables. Then an example system is solved

to complete the picture of the structure and the methodology of the solution procedure.

A brief reference is made to the attempts made by Barr et al [8] and Singh [10] to enhance the performance of the OKA through a better implementation: the solution becomes faster and the memory requirement is minimized.

However, this presentation of the OKA, although it follows [5,6,12], does not cover the proof of the finiteness of the OKA i.e. that the algorithm will terminate in a finite number of steps, either to an optimum or to an unfeasibility; this is covered in [5,6].

1.2.2 Power system applications of the OKA

The applications of the OKA to operational and planning power system problems are given in the second part of the thesis. This part, which is spread over four chapters, illustrates the modeling versatility of the algorithm and presents formulations and solutions to some power systems applications (such as active power economic scheduling, load flow calculations, reliability analysis, etc).

a. Active power economic scheduling

The problem of optimal allocation of active power to generating sets is formulated as a capacitated network

flow problem. The OKA divides the load between the generators in such a way that the total cost of generation is kept to a minimum and the capacity constraints imposed on generators/transmission lines are recognized (by taking into consideration their upper and lower limits).

The objective function (the cost of generation) is assumed linear. A non-linear cost function could be linearized through a piecewise linearization technique. Obviously the lower the number of segments used in the linearization process the higher is the speed of computation i.e. a tradeoff exists between the degree of accuracy required and the computation time.

Transportation algorithms, when applied to the analysis of an electric network, model Kirchoff's current law through the flow conservation constraint at each node of the network. As for Kirchoff's Voltage Law (KVL), the NF methods are unable to model it directly and the line flows obtained may lack accuracy (see appendix C). However, this has been accounted for when KVL is applied to the basic loop equations (the voltage sum around each loop is equal zero).

b. Load flow calculations

In chapter IV, a network flow algorithm is developed to solve the AC and DC load flow calculations. The idea is not to compete with existing conventional solutions but to be used for applications where the load flow solution is

only part of a larger problem (such as security analysis).

The NF formulation of the load flow problem uses the loop frame of reference; thus the advantages of the solution are more significant for systems having fewer loops relative to the number of nodes. The AC solution iterates between the decoupled active and reactive parts until convergence is reached. The algorithm developed is applied to standard test systems and the results are compared to those obtained by conventional methods.

c. Reliability analysis

One of the major problems that faces systems engineers is to find out how reliable or how safe will the system be during its operating life. The study that discloses the solution to the above problem, known as reliability assessment, is an area where NF algorithms and graph theory methods have considerable benefit because reliability studies involve network modeling and graph analysis where the definition of the cut set (a cut set is a set of system components which, when failed, causes failure of the system) is crucial for the evaluation of the system reliability.

Using the NF algorithm, a reliability assessment study is carried out, based on probabilistic methods, and an index for LOLP is evaluated.

d. Other NF applications

Chapter VI highlights some other potential applications of the NF methods. It is a brief survey collected from the power system literature which clearly shows what might be called the modeling versatility of NF algorithms.

The operational planning of a hydrothermal power system is perhaps the most researched area for NF methods. Many papers in the literature report on the use of NF methods to solve the optimal operational planning of large hydrothermal systems. Some of these attempts cast the whole problem into a network flow formulation, while others use hybrid methods where NF is combined with methods such as Benders' Decomposition. Hence, the NF is applied to the subproblem which is network flow compatible, retaining its advantages in the overall solution.

Security analysis applications are promising potential uses of NF methods. Problems such as Group transfer, Emergency rescheduling, Security constrained economic dispatch and Optimal corrective switching have been solved using NF algorithms. In all cases, the power system is modeled so as to obtain the same power flow distribution as with DC load flow; and the emphasis is on the speed of solution when the accuracy is satisfactory. The algorithms developed were intended for real time uses.

One interesting application (particularly now that the Electric power supply industry in Britain is partitioned into several independent bodies) is the power scheduling between different generating utilities, aimed at the minimization of the total cost of generated power. The model yields active power flows (as with a DC model) from generating plant to the customer from which 'wheeling' costs can be deduced.

1.2.3 NOTES FOR FUTURE USES

The final part of the thesis (chapter VII) discusses the previous chapters and makes some recommendations to be considered in future applications of NF algorithms.

The chapter is divided into two segments; in the first, the computer implementation of the OKA and suggested improvements (particularly in the Labeling procedure) are illustrated. The aim is to obtain a better computer code which takes advantage of NF algorithms to give significant improvements in terms of computation speed and memory size. In the second part, the power system applications of NF methods are addressed in an attempt to define new areas of interest for power system engineers. Finally, the references provided consist of a bibliography on Network Flow methods and related research topics reported in the power systems literature. As far as the author is aware such a bibliography has not existed before.

CHAPTER II

THE THEORY OF THE OUT-OF-KILTER ALGORITHM

2.1 INTRODUCTION:

In 1961 D.R. Fulkerson presented an *Out-Of-Kilter Algorithm* (OKA) for network flow problems. This was a major step forward in the optimal solution of difficult transportation systems by an efficient and fast method. It represents a significant contribution to the theories of network analysis and is considered to be the most general and widely used algorithm when dealing with capacitated, deterministic network flows. The algorithm is carefully developed using the concepts of linear programming duality theory and complementary slackness condition. In this chapter the *Out-Of-Kilter Algorithm* (OKA) will be discussed and the underlying theory presented.

The actual mechanics of the solution technique is explained in a general but technically sound framework and a set of decision rules leading to the problem solution is stated. The entire algorithm is then summarized and condensed to five basic steps guided by tabular decision rules.

Finally, by means of a small example problem, manually solved, the use of the algorithm is demonstrated. Other presentations of the OKA are given by [5,12].

2.2 BASIC TERMINOLOGY

To provide a common ground for communication it is necessary to define some basic terminology relevant to network flows problems.

(i) NODE

A *node* is the basic entity of a network diagram and visually represents a physical origin or terminating point, such as a factory, retailer or source of manpower etc. A node that generates flow is called a *source node*, a node that consumes flow is called a *sink node*. Originally the nodes of the network may have been identified with unique names or numbers. Instead, for efficient computation they are renumbered using the first n positive integers, n being the number of nodes in the network. The set of nodes in a network will be designated N .

(ii) ARCS

Arcs are the lines that connect the various nodes in the network and are sometimes called *branches*. Each arc is considered to be directed from the first, or start node of the node-pair, to the second or terminal node of the node-pair. A *directed arc* is one in which flow is only permitted in a predesignated direction. Diagrammatically arc k is represented as a line joining point i to point j with direction indicated by an arrowhead as in Fig 2.1.

The set of arcs in a network will be numbered with the first a positive integers, a being the number of arcs in the network. This permits an easy consideration of parallel-arc problems. The set of arcs in a network will be designated A .

(iii) NETWORK

A network, or graph $G=[N,A]$, is an abstract object made up of a set N of nodes i ; and also a set A of ordered node-pairs or arcs $k=(i,j)$ with $i,j \in N$ (fig 2.2). It represents a physical process in which units move from source(s) to sink(s). A network may have multiple sources and sinks such as that given in Fig 2.3. In this case nodes 2 and 3 are source nodes and nodes 4 and 6 are sink nodes. Nodes 1 and 7 have been added with dotted arcs and are called *supersource* and *supersink* respectively. If, in a network, the flows in arcs are bounded by upper and/or lower bounds (these may be zero or infinity) the network is said to be capacitated. The following notations will be used:

f_{ij} =flow through arc (i,j)

L_{ij} =lower bound on arc (i,j)

U_{ij} =upper bound on arc (i,j)

c_{ij} =cost associated with shipping one unit
of flow from node i to node j



Fig 2.1 A directed arc

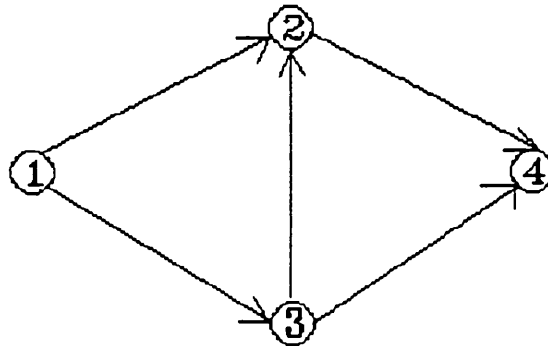


Fig 2.2 A simple network

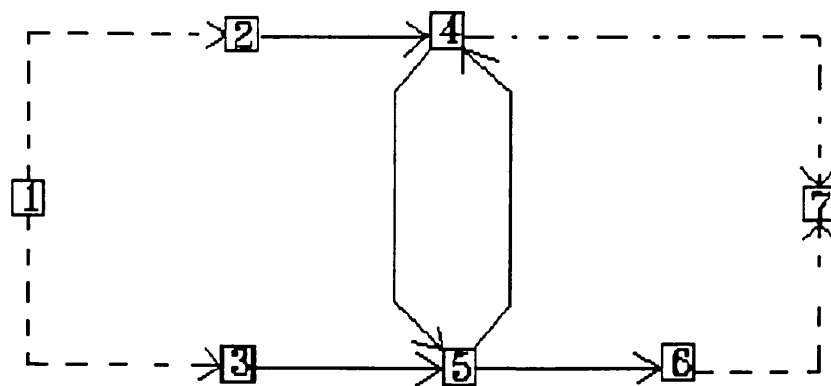


Fig 2.3 Network with multi sources & sinks

(IV) PATH

A path, in a network, may be described as a continuous sequence of connected arcs leading from one node to another. The path could include arcs of both forward and reverse orientations. The diagram in Fig 2.4 provides an example of a path from node 3 to node 7. A path in which the origin node coincides with the terminal node is called a cycle or a closed path.

(V) CIRCULATION

A circulation is an assignment of flow to arcs such that flow is conserved at each node. Thus it is often necessary to modify the original network to provide for circulations and an additional arc is required to connect the sink to the source. This is called the return arc. For the network in Fig 2.2, the dotted arc is the arc which connects the sink with the source (Fig 2.5).

2.3 NETWORK FLOW

The network flow problem may be represented as a special linear programming problem. The steady state flow of a single commodity through a given network is the central concept of our discussion.

Let f_k be the amount of flow passing through arc $k=(i,j)$ from the start node i to the terminal node j .

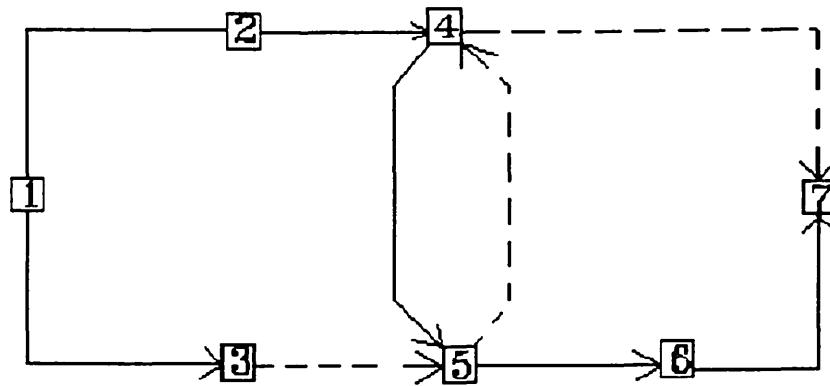


Fig 2.4 A path along a network

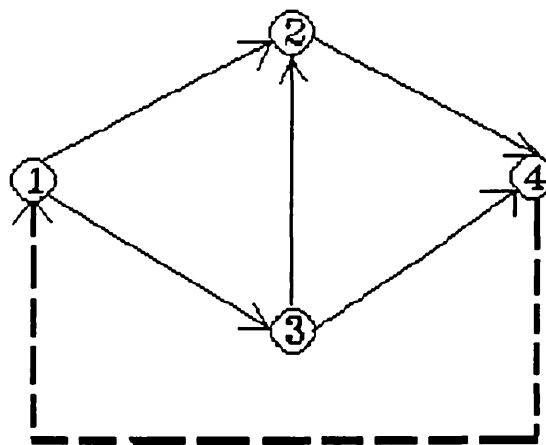


Fig 2.5 The return arc from sink to source

Since the arc flow f_k is considered as a component of the overall network flow, the latter can be denoted by the vector

$$f = (f_1, f_2, \dots, f_k, \dots, f_A)$$

The cost of shipping one unit across arc (i, j) is c_{ij} ; hence the problem becomes to minimize total cost:

$$\text{Min } \sum c_{ij} f_{ij} \quad (2.1)$$

subject to capacity constraints

$$L_{ij} \leq f_{ij} \leq U_{ij} \quad (2.2)$$

The "conservation of flow" constraint is also imposed in order to ensure that the flow entering a node balances the flow leaving; this is expressed by the conservation equations

$$\sum f_{ij} - \sum f_{ji} = 0 \quad \text{for all } i \in N, i \neq j \quad (2.3)$$

and nonnegativity is usually required of the flows:

$$f_{ij} \geq 0 \quad \text{for all arcs } (i, j) \quad (2.4)$$

A network flow which satisfies (2.2) and is conservative is called a *feasible flow*. A feasible flow which minimizes the linear cost form (2.1) is an *optimal flow*.

Associated with this problem is a dual problem which may be stated as

$$\text{Max } \sum (\sigma_{ij} L_{ij} - \mu_{ij} U_{ij}) \quad (2.5)$$

subject to

$$c_{ij} - \pi_i + \pi_j - \sigma_{ij} + \mu_{ij} \geq 0 \quad (2.6)$$

$$\sigma, \mu \geq 0$$

The out-of-kilter method will find a minimal cost network flow solution, if it exists. In doing so the algorithm makes use of the additional set of variables π called the *node price vector*. The node price vector is given by

$$\pi = (\pi_1, \pi_2, \dots, \pi_n) \quad (2.7)$$

Where the π variables are associated with the conservation of flow constraints of the primal problem. The π 's are unrestricted because those constraints are equalities. The μ variables of the dual problem are associated with the upper bound constraints of the primal and the σ variables are associated with the lower bound constraints.

2.4 THE OUT-OF-KILTER ALGORITHM

The out-of-kilter algorithm attempts to find values for the π 's and f_{ij} 's that satisfy the optimality conditions (in sec 2.4.1). The complete algorithm segments naturally into several interrelated sub-tasks (each of which is formulated as a subroutine). These are :

- Arc state determination
- Optimizing cost calculation
- Path searching by a labeling technique
- Flow changing
- Node price augmentation

- Construction of network topology tables.

There is a property associated with each arc of the network called its state. An arc is either in an *in-kilter* state or in an *out-of-kilter* state (*kilter* denotes feasible and optimal).

The algorithm proceeds by examining arcs one at a time in their numerical order. First the state of the arc being inspected is determined; if the arc is *in-kilter* no action is taken since the algorithm is constructed so that subsequent operations will not put this arc *out-of-kilter*. However, if the arc is *out-of-kilter* the algorithm undertakes one or more flow changes or node price increases or both until the arc is left in an *in-kilter* state. Having scanned all arcs and left each one *in-kilter* the algorithm has constructed an optimal, feasible solution. The other terminating condition, that of no feasible solution, is detected when an infinite node price increase is computed.

2.4.1 OPTIMALITY CONDITIONS

There is a set of conditions, which when satisfied, allow termination of the OKA with the conclusion that an optimum has been reached. These conditions are a specialization of the Kuhn-Tucker conditions [11] and can be stated as follows:

$$A f = r \quad , \quad 0 \leq f \leq U \quad (2.8)$$

$$c_{ij} + \pi_i - \pi_j = \sigma_{ij} - \mu_{ij} \quad (2.9)$$

$$(f_{ij} - U_{ij}) \mu_{ij} = 0 \quad (2.10)$$

$$f_{ij} \sigma_{ij} = 0 \quad (2.11)$$

and

$$\mu_{ij} \cdot \sigma_{ij} \geq 0 \quad (2.12)$$

where A is a node-arc incidence matrix and ' r ' represents the node injection: if $r_i > 0$, then node i is a supply node; if $r_i < 0$, node i is a demand node. σ and μ are multipliers.

Hence, if one obtains a set of vectors (f, π, μ, σ) satisfying (2.8)-(2.12), f is an optimum for network programming. $\bar{C}_{ij}$ is calculated from the arc cost and the prices of the nodes of the given arc according to:

$$\bar{C}_{ij} = c_{ij} + \pi_i - \pi_j \quad (i, j) \in A \quad (2.13)$$

this is referred to here as *optimizing cost*.

By considering the three cases below, we can eliminate the unknowns μ and σ .

case 1: Suppose for some arc k , $\bar{C}_k > 0$. Then $\sigma > 0$ from (2.9) and (2.10). Also, $f_k = 0$ to satisfy (2.11).

Furthermore $\sigma_k = \bar{C}_k$, $\mu_k = 0$, $f_k = 0$ solve (2.9)-(2.12) for all arcs k .

case 2: Suppose that $\bar{C}_k < 0$. Then $\mu > 0$ from (2.9) and

(2.12). Also $f_k = U_k$ to satisfy (2.10) and $\mu_k = -\bar{C}_k$.

$\sigma_k = 0$, i.e. $f_k = U_k$ solves (2.9)-(2.12).

case 3: If $\bar{C}_k = 0$, then $\sigma_k = \mu_k$ solves (2.9)-(2.12).

These three cases taken together, allow us to restate the optimality conditions in a more convenient form as in table 2.1.

2.4.2 ARC STATES

Looking at the individual arcs in terms of the optimality conditions, nine mutually exclusive states of an arc are possible as the algorithm proceeds to optimality (table 2.1). Every arc of the network must be in one of the nine states listed.

Two main groups of states may be distinguished; three in-kilter states and six out-of-kilter states. The latter group divides into two subgroups of three states each; the first with subscript 1 and the second with subscript 2. An arc in a 1-subscript state requires an increase in flow to put it in kilter; while an arc in a 2-subscript state must have its flow decreased before reaching an in-kilter state. In addition, an arc in state 2 or 1 where the flow is feasible but non optimal, can be put in-kilter through a node price increase.

The general thrust of the algorithm is to bring each of the out-of-kilter arcs in-kilter by adjusting its flow or increasing its node potential. In order to change the

KILTER	FLOW	ARC STATE	OPTIMIZING COST	CONSTRAINTS
IN	CORRECT	α	$\bar{C}_{ij} > 0$	$f_{ij} = L_{ij}$
	"	β	$\bar{C}_{ij} = 0$	$L_{ij} \leq f_{ij} \leq U_{ij}$
KILTER	"	γ	$\bar{C}_{ij} < 0$	$f_{ij} = U_{ij}$
OUT OF KILTER	TOO	α_1	$\bar{C}_{ij} > 0$	$f_{ij} < L_{ij}$
	LOW	β_1	$\bar{C}_{ij} = 0$	$f_{ij} < L_{ij}$
		γ_1	$\bar{C}_{ij} < 0$	$f_{ij} < U_{ij}$
		α_2	$\bar{C}_{ij} > 0$	$f_{ij} > L_{ij}$
	TOO	β_2	$\bar{C}_{ij} = 0$	$f_{ij} > U_{ij}$
	HIGH	γ_2	$\bar{C}_{ij} < 0$	$f_{ij} > U_{ij}$

Table 2.1 The optimality conditions

flow of an out-of-kilter arc (s,t) a suitable path (called a flow augmentation path) must be found from node t to node s which in conjunction with (s,t) forms a cycle. Flow is then adjusted on each arc by amounts which maintain node conservation and contribute towards bringing (s,t) in kilter.

The path is found (or its nonexistence determined) by alternative use of a labelling procedure and a potential change procedure. This is achieved with :

- no in-kilter arcs are thrown out-of-kilter
- no out-of-kilter arcs are thrown further out-of-kilter.

2.4.3 THE LABELING PROCEDURE

This consists of generating from node t a tree of arcs through which flow may be changed without violating their bounds and without increasing the total cost unnecessarily. (Flow can be algebraically increased from node i to node j either by increasing the flow in the arc (i,j) or by decreasing the flow in the arc (j,i) , if it exists. Total costs need not be increased so that boundary conditions on a given arc can be met).

Labelling is begun from a given node called the *origin* by attempting to reach some other given node called the *terminal*. The following labelling rules are applied:

If node i is labelled $[H+, \epsilon_i]$, and node j is unlabelled, and if (i, j) is an arc such that either

$$\begin{aligned} (a) \quad \bar{C}_{ij} &> 0 \quad , \quad f_{ij} < L_{ij} \\ (b) \quad \bar{C}_{ij} &\leq 0 \quad , \quad f_{ij} < U_{ij} \end{aligned} \tag{2.13}$$

then node j receives the label $[K+, \epsilon_j]$, where for

$$(a) \quad \epsilon_j = \min(\epsilon_i, L_{ij} - f_{ij}) \quad \text{and} \quad (b) \quad \epsilon_j = \min(\epsilon_i, U_{ij} - f_{ij}).$$

If node i is labelled $[H+, \epsilon_i]$, and node j is unlabelled, and if (j, i) is an arc such that either

$$\begin{aligned} (c) \quad \bar{C}_{ji} &\geq 0 \quad , \quad f_{ji} > L_{ji} \\ (d) \quad \bar{C}_{ji} &< 0 \quad , \quad f_{ji} > U_{ji} \end{aligned} \tag{2.14}$$

then node j receives the label $[K-, \epsilon_j]$ where for

$$(c) \quad \epsilon_j = \min(\epsilon_i, f_{ji} - L_{ji}) \quad \text{and} \quad (d) \quad \epsilon_j = \min(\epsilon_i, f_{ji} - U_{ji}).$$

The first label, $K+$, notes that node j was reached over the forward arc k . The second, ϵ_j , is a record of the maximum permissible flow increase along the path up to node j . Note that the origin for labelling, i.e. the node from which the labelling is begun, will be given a large number as its second label ($\epsilon = \infty$).

Looking at the arc states table (table 2.1) the following points can be noted:

suppose an arc (i, j) is found to be out-of-kilter due to

low flow and needs a flow increase to put it in kilter. The arc must be in either states β_1 , γ_1 or α_1 : we label node j , $[K+, \epsilon_j]$; this tells us that node j receives ϵ_j additional units over the forward arc k . Then

$$\begin{aligned} \text{if the arc is in state } \alpha_1, \text{ then } \epsilon_j &= \min [\epsilon_k, L_k - f_k] \\ \text{if in states } \beta_1 \text{ or } \gamma_1, \text{ then } \epsilon_j &= \min [\epsilon_k, U_k - f_k] \end{aligned}$$

Yet, if the arc flow needs to be decreased to bring the arc in-kilter, this arc must be in either states α_2 , β_2 or γ_2 . (Note that in case of β_2 , even though an increase in flow to the lower bound will bring the arc in-kilter, since $\bar{C}_{ij} = 0$, the flow can actually be increased to the upper bound without driving the arc's state out-of-kilter). Thus, we label i , $[K-, \epsilon_i]$. This says that the flow leaving node i and entering node j can be reduced by ϵ_i . Then

$$\begin{aligned} \text{if the arc is in state } \gamma_2, \text{ then } \epsilon_i &= \min [\epsilon_i, f_k - U_k] \\ \text{if in state } \alpha_2 \text{ or } \beta_2, \text{ then } \epsilon_i &= \min [\epsilon_i, f_k - L_k] \end{aligned}$$

Lastly, if the arc is in-kilter (states α , β or γ) the flow should not be altered, except for state β in which flow might be increased or decreased without violating any conditions.

The labelling procedure terminates in one of two ways called *breakthrough* and *nonbreakthrough* respectively: either the terminal receives a label or no more labels can be assigned and the terminal has not been labeled.

2.4.4 FLOW CHANGE

After each breakthrough comes the task of changing the flow in the cycle just found by an amount ϵ

$$\epsilon = \min(\epsilon_t, \epsilon_k)$$

ϵ is the flow change label on the terminal for labelling and so ϵ_t denotes the maximum permissible flow increase along the path portion of the cycle. Similarly, ϵ_k gives the capacity for flow change of the scanned arc. The cycle flow change that does take place, ϵ , is the lesser of these two numbers.

First, increase by ϵ the flow in the path from origin to terminal for labelling. The path label, $K+$, enables the path to be traced in the opposite direction by backtracking from the terminal for labelling. The label on this terminal will point back over the correct arc to the previous node, where will be found another label leading further, and so on until the origin for labelling is reached. The rules are:

when traversing a forward arc of the path, $K+$, increase the arc flow by ϵ . Decrease the arc flow by ϵ when a reverse arc, $K-$, is encountered. Lastly, change the flow in the scanned arc itself by ϵ . If arc k is in a 1 subscript state it requires a flow increase and if in a 2-subscript state it requires a flow decrease by ϵ .

2.4.5 NODE PRICE INCREASE

A *nonbreakthrough* means that the labelling procedure cannot find a path over which to change the flow in the arc under examination. Consequently, an attempt must be made to modify the node price vector in such a way as to permit a later successful path search.

Let M and $\bar{M}$ denote the sets of labelled and unlabeled nodes respectively, and define two subsets of arcs as:

$$\begin{aligned} A_1 &= \{ ij \mid i \in M, j \in \bar{M}, \bar{C}_{ij} > 0, f_{ij} \leq U_{ij} \} \\ A_2 &= \{ ji \mid i \in M, j \in \bar{M}, \bar{C}_{ji} < 0, f_{ji} \geq L_{ji} \} \end{aligned} \quad (2.15)$$

Increase the prices of the unlabeled nodes $\bar{M}$ by an amount δ equal to the smallest absolute value of optimizing cost $\bar{C}_k$ of arcs defined above. Now define :

$$\begin{aligned} \delta_1 &= \min(\bar{C}_{ij}), A_1 \text{ non empty} & \delta_2 &= \min(-\bar{C}_{ji}), A_2 \text{ non empty} \\ \delta_1 &= \infty, A_1 \text{ empty} & \delta_2 &= \infty, A_2 \text{ empty} \end{aligned}$$

and

$$\delta = \min(\delta_1, \delta_2) \quad (2.16)$$

If sets A_1 and A_2 are both empty, an infinite node price increase is required. This is the second terminating condition where no feasible flow exists.

2.5 A GRAPHICAL INTERPRETATION OF THE OKA

The purpose of this section is to present and discuss a graphical approach that reproduces the corrective actions by the out-of-kilter algorithm.

At any stage of the method, the solutions can be represented by points $(f_{ij}, \bar{C}_{ij})$. The state of any arc (i,j) can be identified by examining the location of the point relative to the corresponding bounds L_{ij} and U_{ij} as shown in fig 2.6. If the arc is in-kilter, the point $(f_{ij}, \bar{C}_{ij})$ falls on the dotted three segment line. The upward vertical segment represents the in-kilter state α , the downward vertical segment represents the in-kilter state γ , and the in-kilter state β is represented by the horizontal segment.

If the arc is out-of-kilter the corresponding point falls somewhere else, so corrective actions need attempting. There are two categories of possible corrective action:

- 1.modification of flows (f_{ij} 's)
- 2.modification of adjusted costs ($\bar{C}_{ij}$'s).

Table 2.2 shows all possible conditions for arcs, while table 2.3 states the flow increase/decrease, for corresponding arcs, needed to put it in-kilter. The modification of flows corresponds to horizontal displacements in the graph (fig 2.7) and the modification of adjusted costs (change of $\bar{C}_{ij}$ values) to vertical displacements. The directions of valid horizontal moves are shown in fig 2.7a and 2.7b, for each of the six out-of-kilter states. Valid vertical displacements are shown in fig 2.7c and 2.7d.

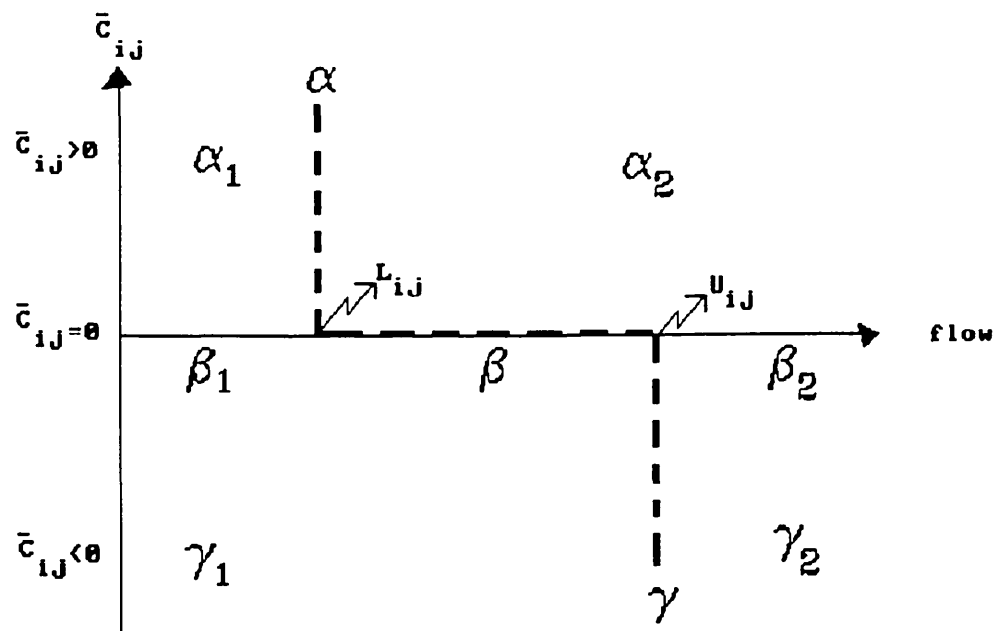


Fig 2.6 Mnemonic device for the OKA

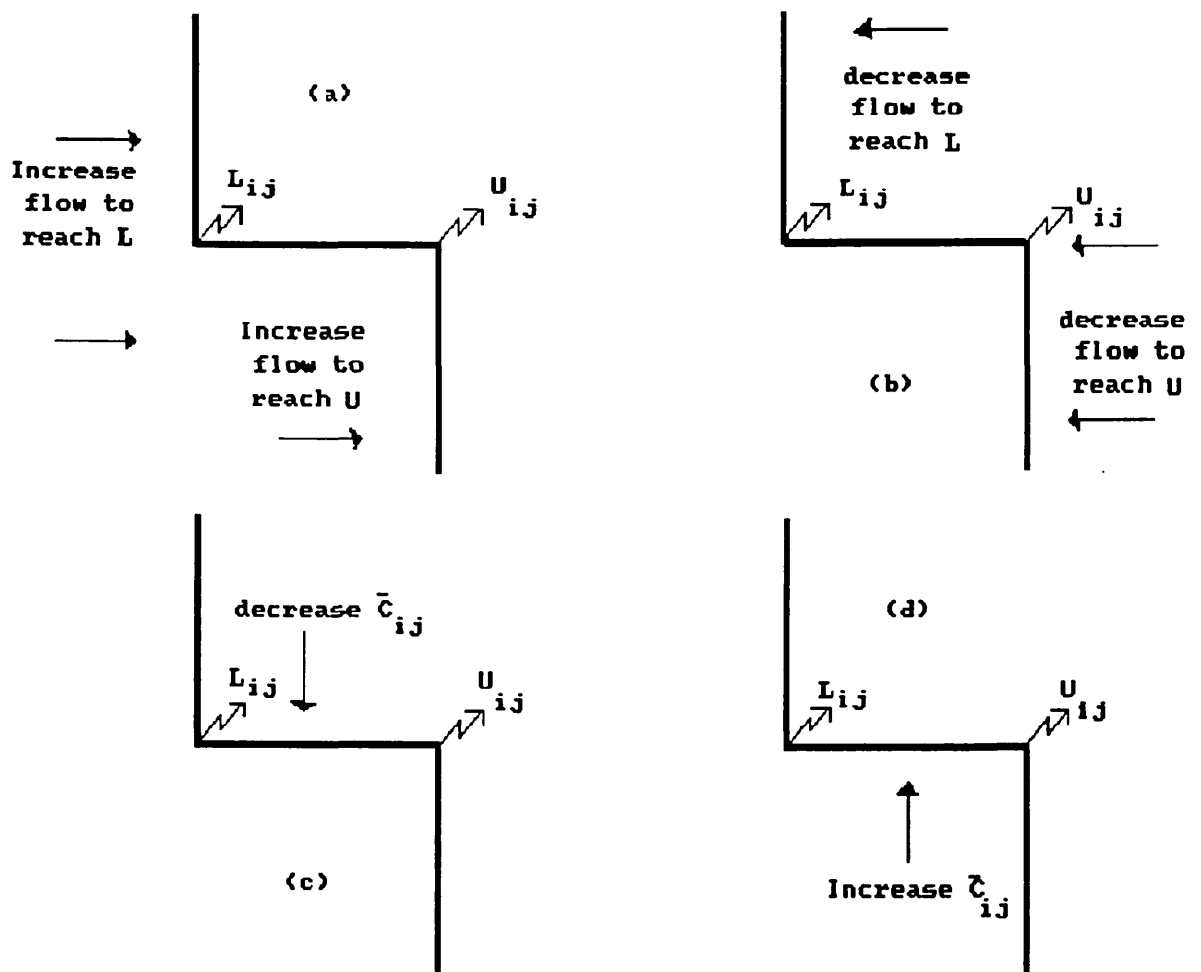


Fig 2.7 Valid directions for displacements

	$\bar{c}_k > 0$	$\bar{c}_k = 0$	$\bar{c}_k < 0$
$f_k = u_k$	Out-Of-Kilter	In-Kilter	In-Kilter
$L_k < f_k < u_k$	Out-Of-Kilter	In-Kilter	Out-Of-Kilter
$f_k = L_k$	In-Kilter	In-Kilter	Out-Of-Kilter

Table 2.2 Possible conditions for arcs

	$\bar{c}_k > 0$	$\bar{c}_k = 0$	$\bar{c}_k < 0$
$f_k = u_k$	f_k	0	0
$L_k < f_k < u_k$	f_k	0	$u_k - f_k$
$f_k = L_k$	0	0	$u_k - f_k$

Table 2.3 The flow change needed
to put arc in-kilter

2.6 ALGORITHMIC STEPS

step 1: Find an out-of-kilter arc (i, j) ; if none, stop.

step 2: Determine if the flow of the arc should be increased or decreased to bring the arc in-kilter. If it should be increased go to step 3; if it should be decreased go to step 4.

step 3: Find a path in the network, using the labeling procedure, from j to i along which the flow can be passed without causing any arc on the path to become further out-of-kilter. If a path is found, adjust the flow in (i, j) . If (i, j) is now in-kilter go to step 1. If it is still out-of-kilter, repeat step 3. If no path can be found, go to step 5.

step 4: Find a path from i to j along which the flow can be passed without causing any arcs to become further out-of-kilter. If a path is found adjust the flow in the path and decrease the flow in (i, j) . If (i, j) is now in-kilter go to step 1. If (i, j) still out-of-kilter repeat step 4. If no path is found, go to step 5.

step 5: Change the π values and repeat step 2 for arc (i, j) keeping the same labels on all arcs already labeled. If the node numbers become ∞ ,

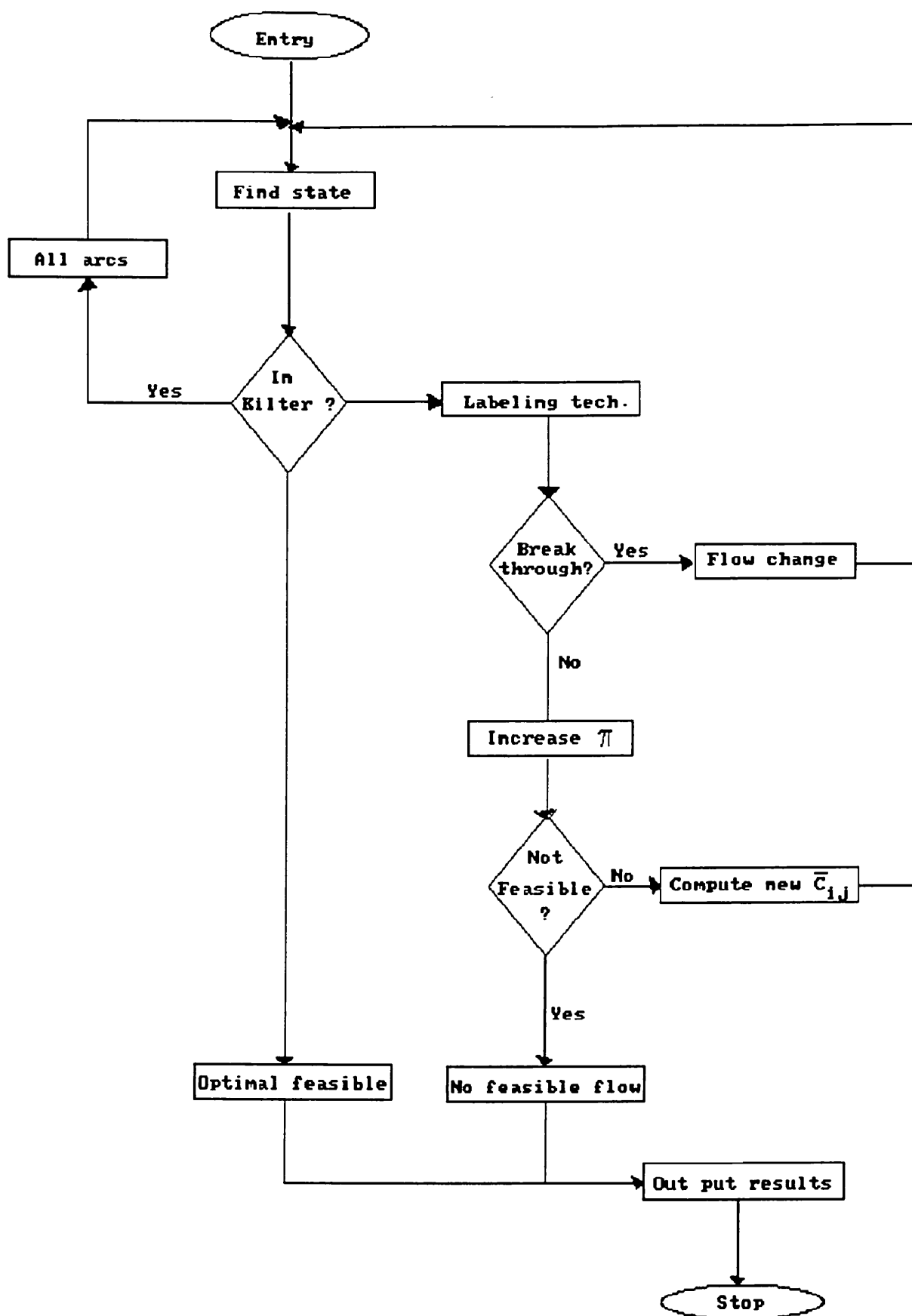


Fig 2.8 The flow chart of the OKA

stop: there is no feasible flow. The algorithmic steps are illustrated in the flow chart (fig 2.8).

2.7 EXAMPLE SOLUTION

To demonstrate the solution methodology of the OKA an example system will be used (Fig 2.9). Suppose that it is desired to ship 3 units of product from source 1 to sink 4 through the network shown in Fig 2.9, where the triplet on each arc represent (U_k, L_k, C_k) . To solve this problem, the network must be "closed" by adding the return arc (4,1). The amount of flow that we wish to ship from source node 1 to sink node 4 will be equal to the amount shipped across arc (4,1). Thus the shipping quantity can be specified by setting $L_{41} = U_{41} = 3$. No cost is associated with that arc, so that $C_{41} = 0$. The complete circulation network is represented by figure 2.10.

The out-of-kilter algorithm will now be used to completely solve the minimum-cost circulation problem. Only two steps need to be taken in order to apply the algorithm:

- 1-choose initial values for the dual variables π
- 2-choose an initial network flow that satisfies the conservation of flow constraint.

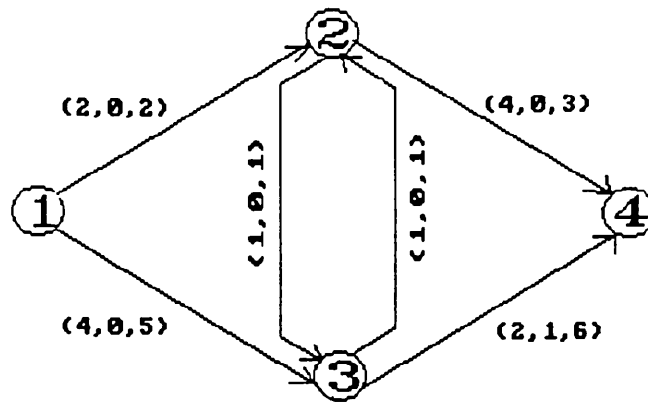


Fig 2.9 Capacitated network

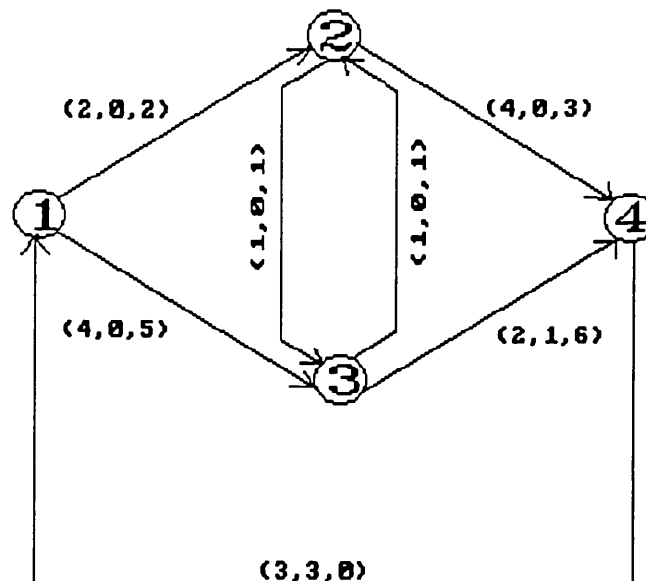


Fig 2.10 Closed capacitated network

The π values can be put equal to zero. A set of network flows satisfying the conservation of flow constraint is given by (if near-optimal solution are chosen initially, the OKA will terminate more rapidly) :

$$f_{12} = 0, f_{13} = 2, f_{23} = 0, f_{32} = 2, f_{24} = 2, f_{34} = 0, f_{41} = 2.$$

Arcs	$\bar{C}_{ij}$	f_{ij}	state	Kilter
(1,2)	2	0	α	Yes
(1,3)	5	2	α_2	No
(2,3)	1	0	α	Yes
(2,4)	3	2	α_2	No
(3,2)	1	2	α_2	No
(3,4)	6	0	α_1	No
(4,1)	0	2	β_1	No

Iteration 1

1. Pick out-of-kilter arc (4,1) (any out-of-kilter arc could have been chosen at this point).
2. State of arc (4,1) is β_1 ; so increase flow by 1 unit to the lower bound = 3.
3. Find path from node 1 to node 4 by labeling technique

Labeling procedure

Node	Label
1	[4+,1] Node 1 is now labelled
2	Cannot be labelled; (1,2) is in-kilter and a flow increase would force it out-of-kilter.
3	Cannot be labelled (a flow increase would drive the arc (1,3) further out-of-kilter).

Non-breakthrough has occurred.

Sets of nodes: $M = \{1\}$, sets of arcs: $A_1 = \{(1,2), (1,3)\}$

$$\bar{M} = \{2,3,4\} \quad A_2 = \{\phi\}$$

and $\delta_1 = \min[2,5]=2$, $\delta_2 = \min[\phi] = \infty$ thus $\delta = 2$.

Thus, increase by δ the node prices of all unlabelled nodes; the new dual variables are now:

$$\pi_1 = 0, \pi_2 = 2, \pi_3 = 2, \pi_4 = 2.$$

Arcs	$\bar{C}_{ij}$	f_{ij}	state	Kilter
(1,2)	0	0	β	Yes
(1,3)	3	2	α_2	No
(2,3)	1	0	α	Yes
(2,4)	3	2	α_2	No
(3,2)	1	2	α_2	No
(3,4)	6	0	α_1	No
(4,1)	2	2	α_1	No

Iteration 2

1. Arc (4,1) is still out of kilter
2. State of (4,1) is α_1 ; so increase flow to lower bound=3.
3. Find path from node 1 to 4 by labelling procedure.

Labeling procedure

Node	Label
1	[4+,1]
2	[1+,1]
3	[2-,1] Flow can be decreased to upper bound without making arc further out-of-kilter
4	[3+,1] Breakthrough has occurred, so change flows in network:

Arcs	$\bar{C}_{ij}$	f_{ij}	state	Kilter
(1,2)	0	1	β	Yes
(1,3)	3	2	α_2	No
(2,3)	1	0	α	Yes
(2,4)	3	2	α_2	No
(3,2)	1	1	α_2	No
(3,4)	6	1	α	Yes
(4,1)	2	3	α	Yes

Iteration 3

1. Pick an out-of-kilter arc, say arc (1,3).
2. State of arc is α_2 ; so decrease flow to lower bound; decrease by 2.
3. Find path from 1 to 3 by labelling procedure.

Labeling procedure

Node	Label
1	[3-,2]
2	[1+,1] Even though in kilter, flow can be increased on arc (1,2) by 1
3	[2-,1] Breakthrough accomplished and flow is changed as indicated by the labelling procedure.

Arcs	$\bar{c}_{ij}$	f_{ij}	state	Kilter
(1,2)	0	2	β	Yes
(1,3)	3	1	α_2	No
(2,3)	1	0	α	Yes
(2,4)	3	2	α_2	No
(3,2)	1	0	α	Yes
(3,4)	6	1	α	Yes
(4,1)	2	3	α	Yes

Still there are two arcs out-of-kilter (1,3) and (2,4).

Iteration 4

1. Pick an out-of-kilter arc, say arc (1,3)
2. State of arc is α_2 ; so decrease flow to lower bound: decrease flow by 1 unit.
3. Find path from 1 to 3 by labelling procedure.

Labeling procedure

<u>Node</u>	<u>Label</u>
1	[3-, 1]
2	Cannot be labelled
4	Cannot be labelled

Thus non-breakthrough has occurred and

$$\begin{aligned}
 M &= \{1\} & A_1 &= \{(1,3)\} \\
 \bar{M} &= \{2,3,4\} & A_2 &= \{\phi\} , \\
 \delta_1 &= \min[3]=3, & \delta_2 &= \min[\phi]=\infty \\
 & \text{so } \delta = \min [3, \infty] = 3
 \end{aligned}$$

The new dual variables: $\pi_1=0$, $\pi_2=5$, $\pi_3=5$, $\pi_4=5$.

Iteration 5

Arcs	$\bar{C}_{ij}$	f_{ij}	state	Kilter
(1,2)	-3	2	γ	Yes
(1,3)	0	1	β	Yes
(2,3)	1	0	α	Yes
(2,4)	3	2	α_2	No
(3,2)	1	0	α	Yes
(3,4)	6	1	α	Yes
(4,1)	5	3	α	Yes

At this point, there is only one arc still out-of-kilter. Once again no-flow augmentation path can be found. Two successive iterations, under the same conditions, will change the dual variables to values

$$\pi_1 = 1, \pi_2 = 5, \pi_3 = 6, \pi_4 = 8,$$

to yield the case below:

Arcs	$\bar{C}_{ij}$	f_{ij}	state	Kilter
(1,2)	-2	2	γ	Yes
(1,3)	0	1	β	Yes
(2,3)	0	0	β	Yes
(2,4)	0	2	β	Yes
(3,2)	2	0	α	Yes
(3,4)	4	1	α	Yes
(4,1)	7	3	α	Yes

Since the algorithm has left each arc in-kilter, the network is in-kilter and conditions for optimality have now been met with optimal minimum cost of 21. Fig 2.11 shows the resulting flows.

2.8 IMPROVED VERSIONS OF THE OKA

The out-of-kilter method of Fulkerson has been used by many researchers, as in [6], to solve network flow problems like the transportation problem, the assignment problem, the maximum flow, the shortest path tree and many others [12].

Nevertheless, there have been many attempts to improve the performance of the OKA in terms of memory requirements and speed of convergence to the optimal solution [7-9].

Barr et al [8] proposed a new formulation of the OKA and presented an extensive computational comparison of a code based on their new formulation with three widely used out-of-kilter production codes, including the SHARE code, which has been used [13] in this research. Barr's formulation is based on a faster procedure to determine the direction of the arc under consideration during the labeling technique.

The study discloses that the new formulation does indeed provide the most efficient solution and it was found to be faster by a factor of 2-5 on small and medium

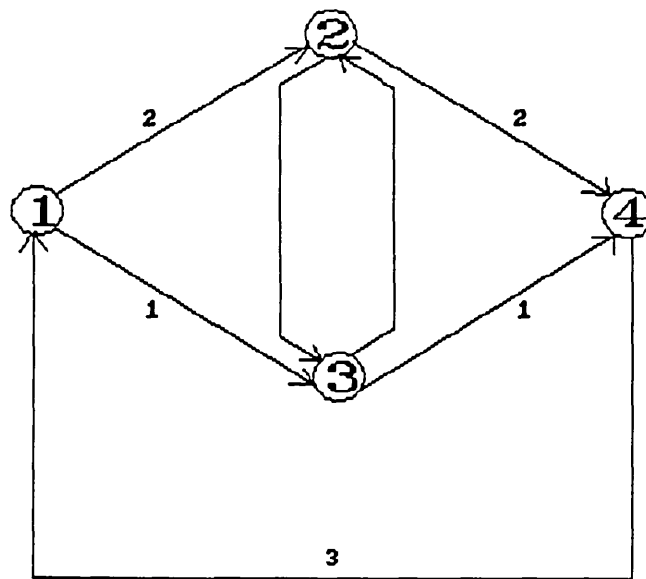


Fig 2.11 The resulting flows

size problems and by a factor of 4→15 on large problems. The new formulation is not without its price in terms of memory space, however.

Recently, another improved code was reported in the literature [10] claimed to be faster than Barr's code [8] by a factor of 1.4→ 5.5 and also uses three fewer arc-length arrays of core memory. This code discriminates among problems of different structures.

2.9 SUMMARY AND CONCLUSION

The primary purpose of this chapter was to present and state in a logical and organized fashion the basic theory and methodology of the out-of-kilter method to solve network flow problems. The steps leading to an optimal solution have been examined and illustrated with the aid of a numerical example solved manually.

A computer program for the OKA is given in appendix B, written in Fortran, with flowcharts for the subroutines involved (Computer codes of the OKA are available in Fortran, Algol and Pascal [5,10]).

The out-of-kilter method has a wide variety of potential uses. The following capacitated network flow problems are discussed in [12]:

1. The transportation problem

2. *The assignment problem*
3. *Maximum flow problems*
4. *The shortest path tree*
5. *The transshipment problem*
6. *Minimum cost/maximum flow problems.*

Many situations seem to be easily described by flow networks and could be transformed into problems appropriate to the out-of-kilter structure. Such problems could be more efficiently solved when using specialized network flow algorithms that may converge to a solution 100 times faster than an ordinary L.P [5].

The out-of-kilter algorithm was developed specifically for network programming and is unique in the mathematical programming literature. Although several comparisons of implementations with other methods (such as the primal-simplex), have been reported in the literature, there is still disagreement as to which method is superior [5,6].

The following chapters address the applicability of the OKA to power systems problems and compare solutions with existing methods. Because the power system optimal flow problem is primarily one of dealing with nodes, arcs, flows and costs, it is capable of solution by network flow methods: this is addressed in chapter III.

CHAPTER III

ACTIVE POWER ECONOMIC SCHEDULING

3.1 INTRODUCTION

Economic dispatch stems from the time that two or more generating units were committed to take up load on a power system where the total generating capacities exceeded the generation required. The problem was how to divide the power between the generating units in an economic way i.e. to minimize the total generation cost. In other words, economic dispatch is a computational process whereby the total generation required is allocated among the generating units available so that the constraints imposed are met and the energy requirements are minimized [14]. Economic dispatch assumes that units are committed to take up load [15] and therefore that no-load running costs are incurred. Various methods have been used, to solve the economic dispatch problem, such as:

- (1) "*the base load method*" where the most efficient unit is loaded to its maximum capability, then the second most efficient unit is loaded and so on.
- (2) "*the best point loading*" where units are successively loaded to their lowest heat rate point beginning with the most efficient unit and proceeding down to the least efficient.

However, it was proved [16] as early as 1943 that the incremental method gives the most economic results [17]: for economic operations the incremental costs of all machines should be equal. The effect of losses in terms of incremental efficiency was included, similar to the penalty factor form applied to a single transmission line [18].

In the early 60's Carpentier[19] placed the economic dispatch, or its more general counterpart, the optimal power flow, on a firm mathematical basis. This was considered as the first step toward the optimal power flow [20] that minimizes the fuel cost, or some other quantity, while recognizing limitations on power system equipment (such as LTC tap limits, voltage limits on buses, etc.).

Recently, network flow methods have gained wide spread interest in their application to power systems problems such as the economic dispatch [21-26]. These methods fall into the category which Ringlee & Wollenberg [27] describe as special optimization methods, characterized as powerful, fast and logic based with very little of the mathematical analysis associated with the conventional methods of economic dispatch.

In this chapter, the economic dispatch problem will be addressed, using the OKA, with appropriate models, to solve the economic scheduling problem of real power when generation and transmission costs are minimized.

3.2 ECONOMIC SCHEDULING

Here, power system economic scheduling of generation is formulated as a capacitated network flow problem and then solved using the OKA. The objective of the OKA is to send flows from sources to sinks by assigning appropriate flows to the network arcs so that all demands at the sinks are satisfied and no flow along any arc violates its upper or lower flow limits. Furthermore, the total cost of generation must be minimal. The problem formulation is:

$$\text{Min} \quad \sum c_{ij} f_{ij} \quad (3.1)$$

$$\text{subject to} \quad \sum f_{ij} = 0 \quad (3.2)$$

$$L_{ij} \leq f_{ij} \leq U_{ij} \quad (3.3)$$

The system is assumed not to have losses included. However, there is a cost associated with transmission and this must be included in the objective function. The cost of transmission is system loss P_L multiplied by the system marginal cost "h". The effect of this transmission cost is to cause an adjustment in the generation schedule. In actual fact, the effect is equivalent to the penalty factor introduced when losses are included in the constraint equation. Hence, the problem formulation becomes:

$$F = \sum F_i (PG_i) + h P_L \quad (3.4)$$

subject to the equality constraint

$$\varphi_n = \sum PG_i - D_n + \sum P_i = 0 \quad (3.5)$$

and subject to the inequality constraints

$$PG_{i \min} \leq PG_i \leq PG_{i \max} \quad \text{for all } i \in G_s \quad (3.6)$$

$$L_k \leq P_k \leq U_k \quad \text{for all } k \in A \quad (3.7)$$

This set of equations is similar to the set obtained from the classical formulation [28]. A set of real generation variables PG_i must now be found that minimizes the cost function (3.4) while recognizing the constraint relations (3.5)-(3.7). The OKA searches for the minimum cost solution and then determines the line flows. The objective function has to be linear or linearized [29-31] through a piecewise linearization (Fig 3.1). Also, the power system components must be represented appropriately to enable a network flow structure of the problem. The next section presents the models adopted.

3.2.1 Transmission line model

The real power flow P_k in MW is shown in fig 3.2 flowing from bus i to bus j along link k which may be a line, transformer or reactor. This flow must satisfy upper and lower limits. Line thermal limits, transformer and reactor ratings and system security limits can be approximated by values in MW and chosen for P_k .

Power flows in the opposite directions to those initially assumed (i,j) can be modeled by including arc (j,i) in the formulation.

The model of the lines, transformers and reactors differs only in the derivation of the cost term as it depends on the losses. The incremental cost for a line is [13]:

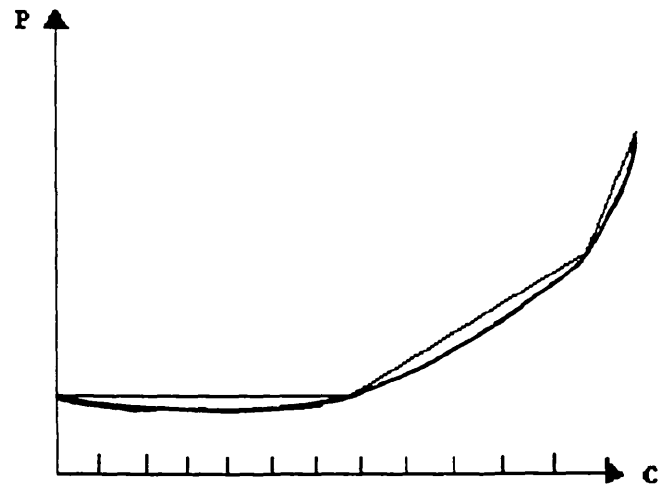


Fig 3.1 Piecewise linearization

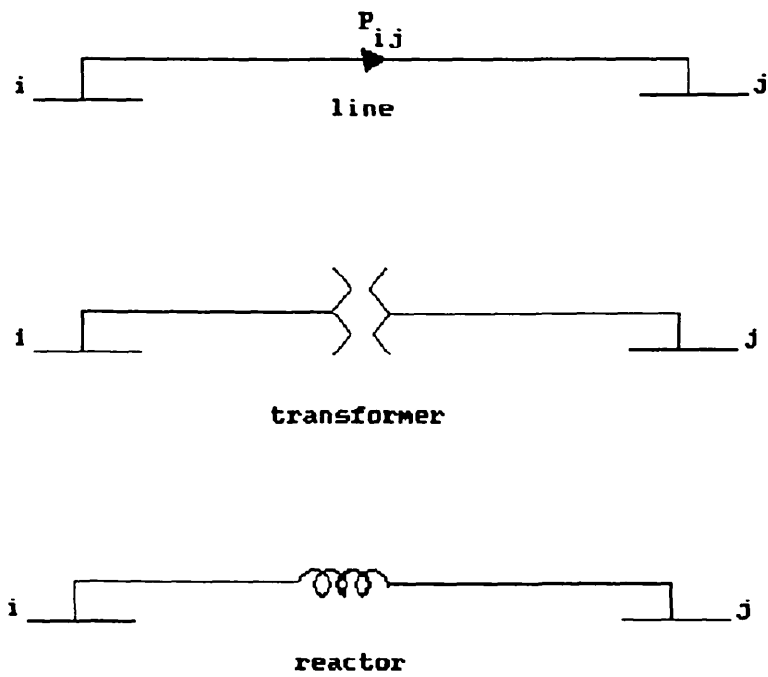


Fig 3.2 Real power flow along a line

$$\lambda_k = \lambda_{avg} W.L_{en} / P_{fl} \quad (3.8)$$

where λ_{avg} is the average incremental cost of generation based on the λ_k of the generators.

W is the three-phase full load loss per km in MW/km

P_{fl} is the full load power rating of the line in MW

and L_{en} is the line length in km.

The formula for transformer and reactor incremental cost is :

$$\lambda_k = \lambda_{avg} P_L / P_{fl} \quad (3.9)$$

where P_L is the three phase full load loss in MW.

As the link incremental costs are decimal fractions, a scale factor is used $Sc = 10^5$ to shift the third significant digit to the units position of the arc cost.

3.2.2 Generating plant model

Since each generator must not be operated above its rating or below some minimum power, the generator power PG cannot lie outside the range stated by the inequality

$$PG_{i \min} \leq PG_i \leq PG_{i \max} \quad (3.10)$$

For instance, the real power generated PG_i has a lower limit $g_{k \min}$ set by the ability of the boiler to run on low load or in some cases by overheating in the low pressure stage of the turbine. Upper limit $g_{k \max}$ corresponds to the designed capacity of the boiler turbine combination to produce power at the alternator shaft. Hence :

$$g_{k \min} \leq g_k \leq g_{k \max} \quad (3.11)$$

If, as is usually the case, a number of generators supply power to bus j , for each generator, k , a dummy node i and a dummy arc (Δ, i) are created (Δ is the source node), the arc having in-active bounds $L_{\Delta i}$ and $U_{\Delta i} = X$, a large number. The generation cost curve for set k is given by the linear equation:

$$c_k = \lambda_k g_k + b_k \quad (3.12)$$

where c_k is the cost of generation in \$/ Hr

and b_k is the no-load cost in \$/ Hr

The total generation cost for the system is :

$$c_T = \sum \lambda_k g_k + b_T \quad (3.13)$$

where b_T is the total no-load generating cost. This can be omitted since the optimal solution is independent of this constant. Cost scale $Sc=10^5$ was also applied to the generator k . The dummy arc cost $c_{\Delta i}$ was set equal to zero.

3.2.3 Demands and imports

At any bus i , power imported or exported can be modeled by a single arc connected to the sink $\mathcal{V}$ as in fig 3.3. For import the arc is $(\mathcal{V}, i)$ and for export $(i, \mathcal{V})$. The flow in this arc can be achieved by letting

$$P_{Di} = L_{i\mathcal{V}} = U_{i\mathcal{V}} \quad (3.14)$$

and the algorithm will ensure that $f_{i\mathcal{V}} = P_{Di}$, the demand

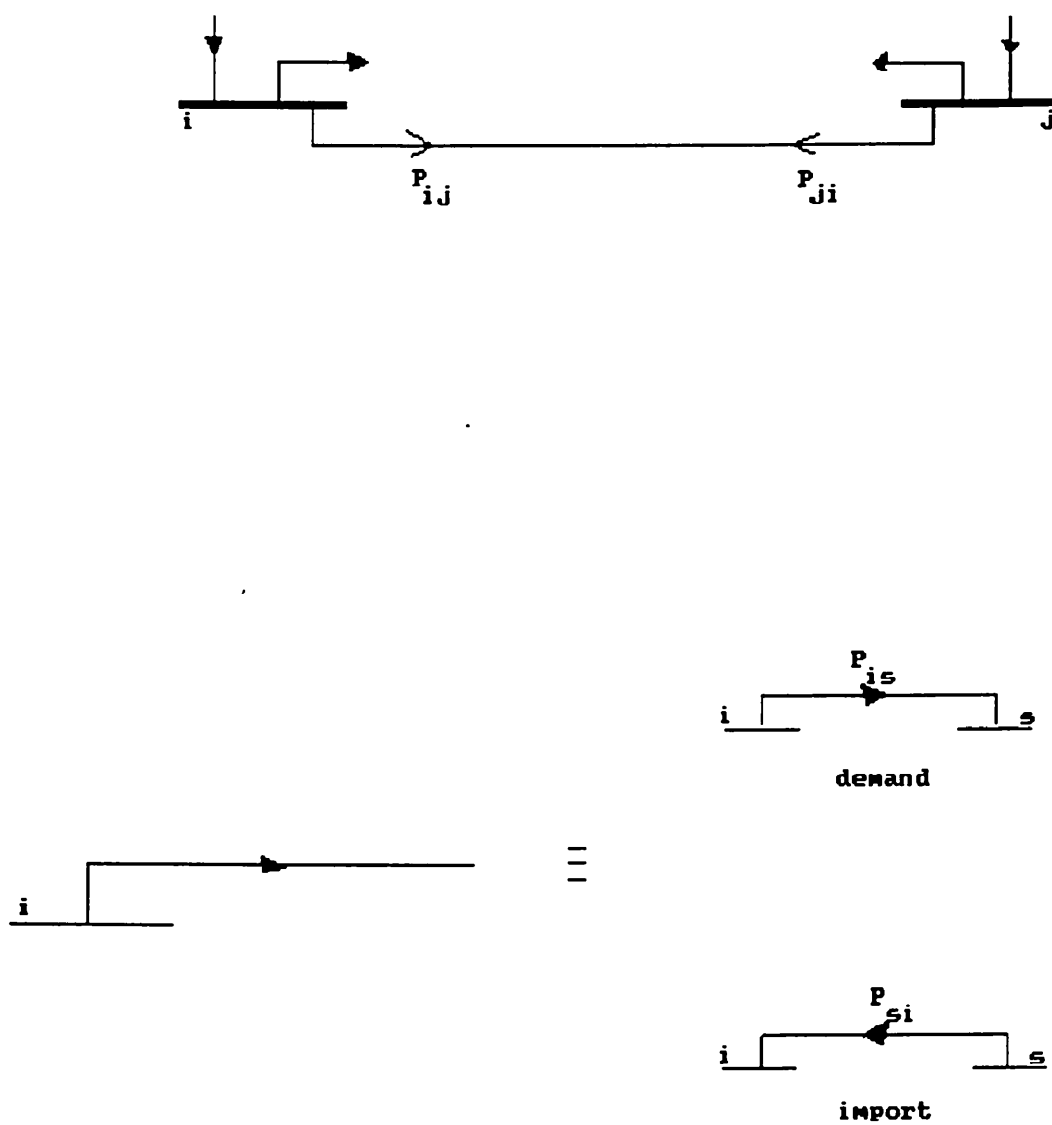


Fig 3.3 Demand and import models

value at i (or $f_{gi} = |P_{Di}|$ in case of import). The demand is not costed, therefore $c_{ig}=0$.

3.3 COMPUTER IMPLEMENTATION

The OKA was written as a group of seven co-operating subroutines and total agreement was maintained between the mathematical derivation [6] and the computer implementation. Also some power system subroutines were included in the input system data, for example sort generators into order of merit table, tabulated output and results. The Fortran source program is simple and easy to follow as many comment statements are included.

In appendix A, the computer implementation is discussed in detail and the formulation methodology of the subroutines used is illustrated with flowcharts. The Fortran source program is listed in appendix B.

3.4 EXPERIMENTAL RESULTS AND DISCUSSION

The system used was the 23 bus system in Fig 3.4 abstracted from 400/275/132 Kv system of England and Wales. The OKA took 14 ms, on the Vax 8600, to converge and produce as output the transmission line flows and the generators economic loading (the transmission losses were not included). The convergence is always guaranteed, if a solution exists. Table 3.3 shows the line data and power flows.

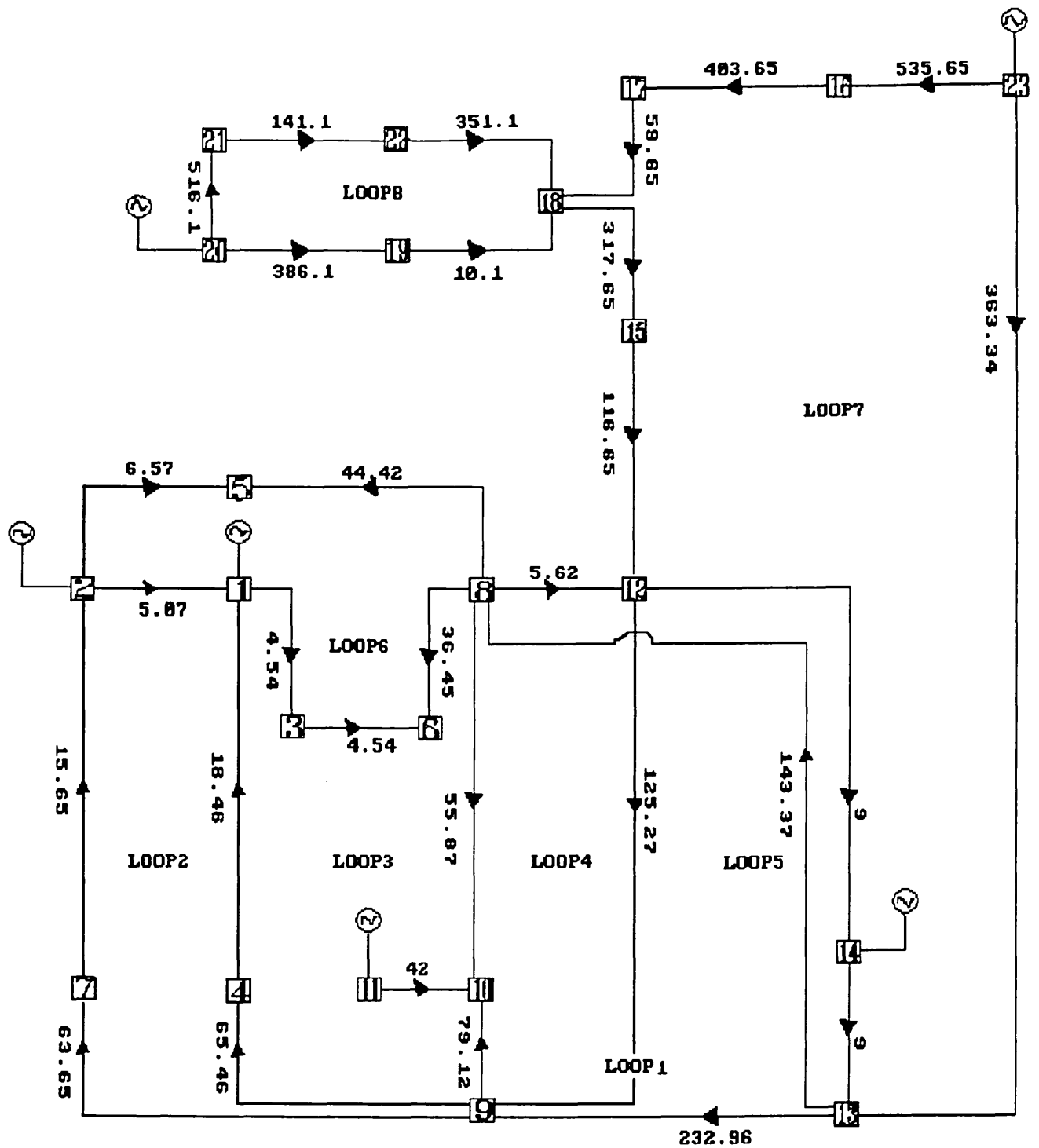


Fig 3.4 The CEGB 23-bus system

As the algorithm, in its present form, does not obey Ohm's law and hence Kirchhoff's voltage law, the line flows are not accurate. However, it was shown [32,33] that close agreement between network flow results and load flow methods is possible if the system under consideration is uniform (i.e. elements with same X/R ratio). In the case of a non-uniform system (elements with different X/R ratios), the computed line flows deviate from actual values. A post dispatch load flow is capable of correcting the errors in the line flow calculation. Also, the line flows in the system were decided by line resistances; this is due to the cost factor imposed on transmission losses. So improvement is feasible by making these line flows mainly decided, instead, by line reactances.

In theory, results obtained from the load flow model would only compare well if there were fairly consistent X/R ratios for the lines and transformers in the system. In practice, there seems to be a compensating effect within the loops of a power system so that generally comparable results are obtained even if the X/R ratios are dissimilar [33].

However, to be completely general, non-uniform ratios must be incorporated into the dispatch model. When there are non-uniform lines in a loop, a circulating flow will exist in the loop to balance the loop voltage difference according to Kirchhoff's voltage law. The calculation of line flows of optimal operation has to take this

situation into consideration. This is addressed in the next section (sec 3.5).

Generator data and output are shown in table 3.4. They illustrate the operational inflexibility of the set of generators at bus Nb 14 as

$$PG_{\min} = PG = PG_{\max}$$

The upper and lower limits were set equal to 83 MW forcing the optimal schedule to ensure an output of 83 MW on all generators at bus 14; or else the branches involved, from the source node to bus 14, would go out-of-kilter. In this particular case, as the incremental cost of generation at bus 14 is the lowest, this could be considered as an advantage to the cost minimization process.

In table 3.1 the transformers and reactor flows are presented, while table 3.2 illustrates that the demand at all busses is satisfied. Although, in the present formulation, the transmission losses are not considered, the OKA has the ability to solve the economic dispatch problem when losses are included by simply assigning these losses as demand to busses at both ends of each line. Hence, the algorithm could ensure that these losses are satisfied.

As the network flow formulation, including the economic dispatch problem, involves only logic and add/subtract statements, it is much faster than the DC load flow. Thus, because of the need for high speed network solutions in real time power system control, the application of the network flow methods is promising even if it only obeys Kirchoff's current law.

Nb	From	To	Cost /MWhr	MVA limit	Flow MW
26	12	8	0.0088	155	6
27	13	8	0.0088	155	155
28	12	9	0.0085	180	180
29	13	9	0.0088	155	155
30	1	2	0.0039	90	4

Table 3.1 Transformer and series reactor data
and power flows

Bus Nb	Demand MW	Bus Nb	Demand MW
1	64	13	-4
2	101	14	480
3	0	15	201
4	47	16	132
5	51	17	344
6	41	18	104
7	48	19	376
8	1	20	-100
9	150	21	375
10	177	22	-210
11	130	23	129
12	6		

Table 3.2 The Demand data

Nb	From	To	Cost /MWhr	Thermal limit	Flow MW
1	1	3	0.0489	100	0
2	1	4	0.0585	100	-23
3	2	5	0.0761	100	0
4	8	5	0.0610	100	51
5	2	7	0.1322	100	0
6	3	6	0.1238	100	0
7	4	9	0.0572	100	-70
8	9	7	0.0430	100	48
9	8	6	0.0842	100	41
10	11	10	0.0901	200	42
11	8	10	0.1127	100	68
12	9	10	0.1127	100	67
13	13	14	0.0495	620	0
14	14	12	0.0495	620	18
15	15	12	0.0433	620	174
16	18	15	0.0411	620	375
17	23	13	0.0998	620	306
18	16	17	0.0114	620	461
19	17	18	0.0220	620	117
20	19	18	0.0167	620	0
21	20	19	0.0519	620	376
22	22	18	0.0271	620	362
23	20	21	0.0229	620	527
24	21	22	0.0146	620	152
25	23	16	0.0231	620	593

Table 3.3 Line data and power flows

Bus Nb	Generator Nb	Lower limit	Out put MW	Upper limit	Cost /MWhr
1	1	15	15	61	3.22
1	2	15	15	61	3.22
1	3	15	15	61	3.22
2	4	15	15	61	3.22
2	5	30	52	61	2.20
2	6	30	30	61	2.20
11	1	43	43	58	2.16
11	2	43	43	59	2.19
11	3	43	43	59	2.17
11	4	43	43	59	2.14
14	1	83	83	83	0.85
14	2	83	83	83	0.85
14	3	83	83	83	0.85
14	4	83	83	83	0.85
14	5	83	83	83	0.85
14	6	83	83	83	0.85
20	5	22	112	112	1.71
20	10	135	334	334	1.42
20	12	143	357	357	1.21
23	6	22	112	112	1.67
23	7	22	112	112	1.71
23	8	22	112	112	1.67
23	9	135	334	334	1.35
23	11	143	358	358	1.15

Table 3.4 Generator data and power output

3.5 CONSIDERATION OF KIRCHOFF'S VOLTAGE LAW

The NF method, being a transportation algorithm, models only Kirchhoff's current law by the flow conservation constraint which ensures that the flow entering a node must also leave. The economic dispatch formulation of the OKA, discussed earlier, does not take into consideration the voltage drop on the lines. This means that the results obtained are considerably different from the DC load flow values [34].

However, it was possible to include the effect of voltage in the economic dispatch solution using a superposition technique where the line flows, obtained by the OKA, are modified by superposing the circulating loop flows. A set of independent loops, in the network under consideration, is defined and the line flows in these loops are adjusted to give zero voltage drop around the loops. In a matrix form, this can be expressed by the following relation

$$[f].[Z]=[L_t][V]$$

where $[f]$ is a column vector whose elements are flows around the independent loops,

$[Z]$ is the loop impedance matrix. The diagonal elements of $[Z]$ are the impedances of the independent loop i.e. the sum of the impedances of the branches contained in one loop, while the off-diagonal elements are the mutual impedances between the independent loops,

$[L_t]$ is the transpose of the branch-to-loop incidence matrix and

$[V]$ is any voltage source in the branches.

Although the set of independent loops cannot always be determined uniquely, this will not affect the final solution. The task of determining the independent loops has to be done once only and considered among the input data of the system.

The formulation of the economic dispatch now has to be altered to allow for the voltage constraint. The solution methodology comes in two stages: First the OKA is used to dispatch the generation and transmission costs while satisfying Kirchoff's current law only; secondly, the flows along the branches, with the generator outputs fixed, are modified to satisfy the voltage law. In other words, the second part of the problem is to find the solution for a set of simultaneous linear equations.

Recalling the economic dispatch solution of the 23 bus system, the results in tables 3.1 and 3.3 were modified to give zero voltage drop around the independent loops. The final solution is shown in tables 3.6 and 3.5. The generator outputs and the demand data are not changed.

Nb	From	To	Cost /MWhr	Thermal limit	Flow MW
1	1	3	0.0489	100	4.5
2	1	4	0.0585	100	-18.5
3	2	5	0.0761	100	6.5
4	8	5	0.0610	100	44.4
5	2	7	0.1322	100	-15.6
6	3	6	0.1238	100	4.5
7	4	9	0.0572	100	-65.5
8	9	7	0.0430	100	63.6
9	8	6	0.0842	100	36.4
10	11	10	0.0901	200	42.0
11	8	10	0.1127	100	55.8
12	9	10	0.1127	100	79.1
13	13	14	0.0495	620	-9.0
14	14	12	0.0495	620	9.0
15	15	12	0.0433	620	116.6
16	18	15	0.0411	620	317.6
17	23	13	0.0998	620	363.3
18	16	17	0.0114	620	403.6
19	17	18	0.0220	620	59.6
20	19	18	0.0167	620	11.0
21	20	19	0.0519	620	387.0
22	22	18	0.0271	620	351.1
23	20	21	0.0229	620	516.1
24	21	22	0.0146	620	141.1
25	23	16	0.0231	620	535.5

Table 3.5 Final real power line flows

Nb	From	To	Cost /MWhr	MVA limit	Flow MW
26	12	8	0.0088	155	-5.6
27	13	8	0.0088	155	143
28	12	9	0.0085	180	125
29	13	9	0.0088	155	233
30	1	2	0.0039	90	-5.0

Table 3.6 Transformer and series reactor final
power flows

The algorithm has been applied, also, to the IEEE 24-bus reliability test system [35] (fig 3.5). The system has a nonuniform X/R ratio. The OKA was used to solve the optimal active power dispatch when the first Kirchoff's law was satisfied. The transmission power losses were minimized; in other words, if line flows were not binding, the voltage law was also satisfied, as proved in appendix C. The results obtained by the OKA, when applied to the 24-bus system of fig 3.5, are shown in tables 3.7 and 3.8.

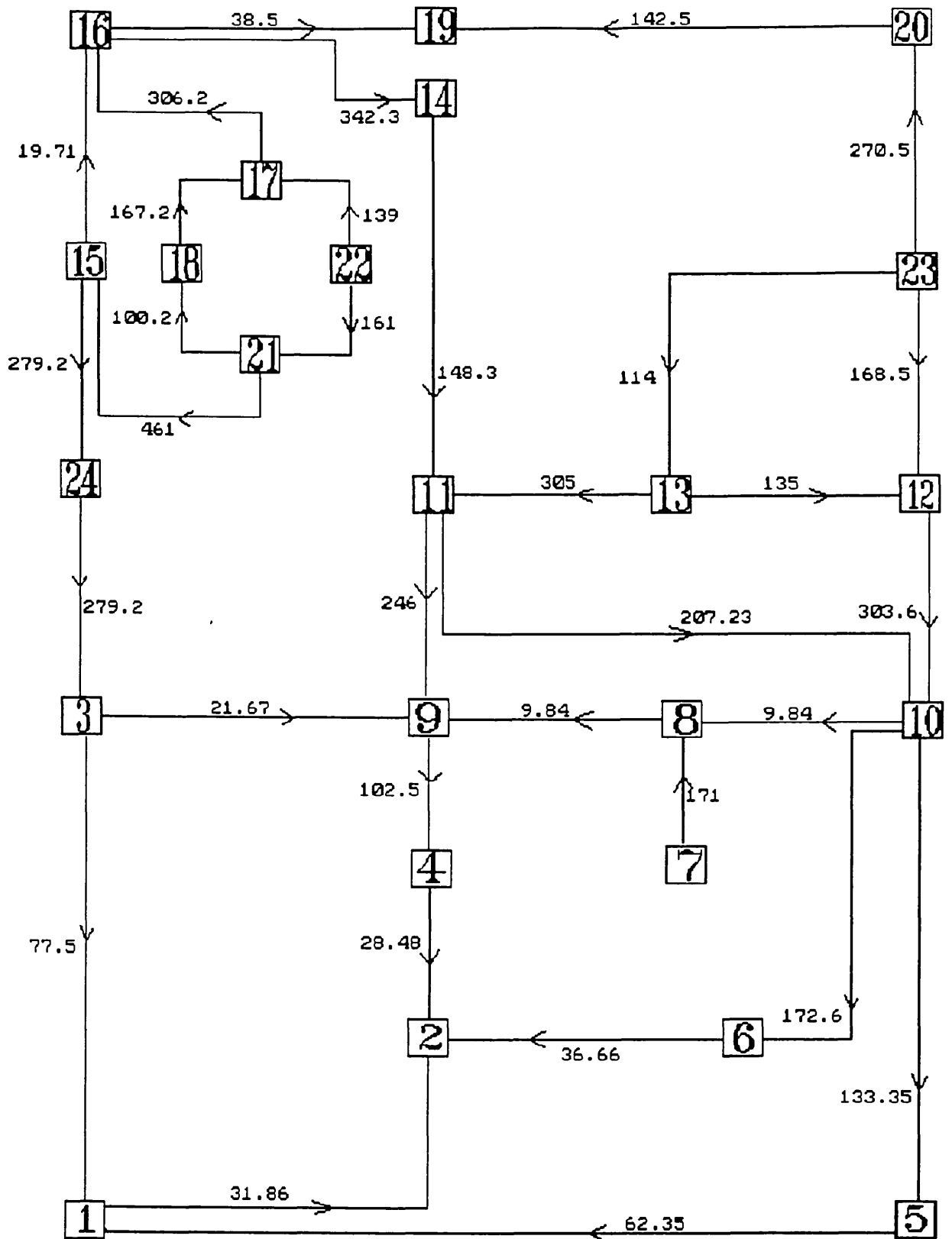


Fig 3.5 The IEEE 24-bus system

BUS Nb	POWER GENERATION MW	DEMAND MW
1	0	108
2	0	97
3	0	180
4	0	74
5	0	71
6	0	136
7	296	125
8	0	171
9	0	175
10	0	195
11	0	0
12	0	0
13	591	265
14	0	194
15	155	317
16	155	100
17	0	0
18	400	333
19	0	181
20	0	128
21	400	0
22	300	0
23	553	0
24	0	0

Table 3.7 Bus Generation and Demand for
the IEEE 24-bus system

FROM BUS	TO BUS	R	X	U LIMIT	FLOW MW
1	2	.0026	.0139	175	19.7
1	3	.0546	.2112	"	-70.1
1	5	.0218	.0845	"	-57.5
2	4	.0328	.1267	"	-44.5
2	6	.0497	.1920	"	-32.7
3	9	.0308	.1190	175	-23.9
3	24	.0023	.0839	400	-226.1
4	9	.0268	.1037	175	-118.5
5	10	.0228	.0883	"	-128.5
6	10	.0139	.0605	"	-168.7
7	8	.0159	.0614	"	171.0
8	9	.0427	.1651	"	27.4
8	10	.0427	.1651	"	-27.4
9	11	.0023	.0839	400	-97.8
9	12	.0023	.0839	"	-192.2
10	11	.0023	.0839	"	-240.2
10	12	.0023	.0839	"	-279.5
11	13	.0061	.0476	500	-201.4
11	14	.0054	.0418	"	-136.5
12	13	.0061	.0476	"	-244.2
12	23	.0124	.0966	"	-227.5
13	23	.0111	.0865	"	-119.7

Table 3.8a IEEE 24-bus Line Data and power flow

FROM BUS	TO BUS	R	X	U LIMIT	FLOW MW
14	16	.0050	.0389	500	-330.5
15	16	.0022	.0173	"	62.6
15	21	.00315	.0245	"	-450.8
15	24	.0067	.0519	500	226.7
16	17	.0033	.0259	"	-316.1
16	19	.0030	.0231	"	103.2
17	18	.0018	.0144	"	-175.8
17	22	.0135	.1053	"	-140.2
18	21	.00165	.01295	"	-108.8
19	20	.00225	.0198	"	-77.7
20	23	.0014	.0108	"	-205.7
21	22	.0087	.0678	500	-159.7

Table 3.7b IEEE 24-bus Line Data and power flow

3.6 FURTHER DISCUSSION AND COMMENTS

The fundamental difficulty that faces the use of NF methods for solving electric networks is the consideration of Kirchhoff's voltage law (KVL). In the previous section, it was shown that KVL could be satisfied using the superposition technique which generates a set of simultaneous equations called the loop equations.

However, appendix C demonstrates that KVL can be taken into account, when using NF methods, without the need for the loop equations. This comes from the fact that in a resistive network, if the losses are minimized, the line flows take such values so that KVL is satisfied, providing the power flow limits are not binding i.e. the branches are unsaturated. Thus, the problem becomes one of minimizing the losses associated with each arc, which has a quadratic function, subject to flow conservation constraint at each node and capacity limits on power flows.

This is implicitly verified using the OKA if line flows are not limited. The accuracy of the solution depends on the number of linear segments used in approximating the quadratic function. Higher degree of accuracy means a higher computational time i.e. we have a trade off between accuracy and speed of solution. In other words, the operator has the opportunity to choose the appropriate degree of accuracy needed for a given application.

In conclusion, the present formulation of the OKA allows for the consideration of KVL using either of two approaches:

- 1- when minimizing losses without saturating branches
- 2- through the loop equations, if power flow limits are binding.

Both approaches will be used in the application of the OKA to the load flow calculations which is presented in the following chapter.

CHAPTER IV

THE LOAD FLOW CALCULATION

4.1 INTRODUCTION

Load flow is the solution of the static load flow equations which describe the relationships between the voltages and powers in the interconnected system. For the system engineer, this is a state of the power system where the demand is satisfied by sending power flows, from generation busses to load busses, along the branches of the given power system network.

It is performed in power system planning, operational planning and real time control. Many different load flow methods have been reported in the literature, which demonstrates the enormous amount of effort put into the development of reliable load flow methods [36,37]. But as the size of the power system is getting larger, the problems associated are getting more complicated. This calls for the development of specialized load flow methods which possess a combination of features such as accuracy, reliability, speed, etc. Thus, it has always been difficult to decide which method is the best choice for a given application. The Newton-Raphson method proved to have powerful convergence characteristic [38,39] and after careful development by Tinney & Walker [40], it became widely regarded as the dominant general purpose load flow approach [41]. This however, did not prevent researchers

from trying further to improve the performance of load flow methods [42-44]; on the contrary such attempts continue to appear in the power system literature [45].

The load flow problem is a flow transportation problem in a network of busses and lines. This makes it possible for network flow methods to be applied on the load flow calculation. Several attempts, that have been reported in the literature [46,47], addressed this particular problem where the aim is not to compete as a substitute for existing solution methods but to make the most of the transportation algorithms when applied to problems requiring fast solutions.

In this chapter, the load flow solution will be formulated in a structure amenable to solution by Network Flow methods (NF). The algorithm developed is applied to both dc and ac load flow calculations and the results obtained, for standard test systems, are compared to those obtained when using the FDLF [42].

4.2 THE LOAD FLOW PROBLEM

The primary function of a transmission power system is to maintain the real and the reactive power supply to meet the various load connected to the system. In doing so, the operating state of the system must be kept within specified tolerances, with respect to voltages and frequency, to ensure quality and continuity of the energy

supply. Thus, it is essential, for the continuous evaluation of the current performance of a system, to have the load flow solution available.

The mathematical formulation of the load flow results in a set of nonlinear equations called the static load flow equations, where the load flow problem is of solving these simultaneous equations [48]; for the two bus system of fig 4.1, the static load flow equations can be written as:

$$P_{G_1} - P_{D_1} = \frac{1}{X_L} \left[v_1^2 \sin \omega - v_1 v_2 \sin[\omega - (\theta_1 - \theta_2)] \right] \quad (4.1)$$

$$P_{G_2} - P_{D_2} = \frac{1}{X_L} \left[v_2^2 \sin \omega - v_1 v_2 \sin[\omega + (\theta_1 - \theta_2)] \right] \quad (4.2)$$

$$Q_{G_1} - Q_{D_1} = \frac{1}{X_L} \left[v_1^2 \cos \omega - v_1 v_2 \cos[\omega - (\theta_1 - \theta_2)] - \frac{v_1^2}{X_C} \right] \quad (4.3)$$

$$Q_{G_2} - Q_{D_2} = \frac{1}{X_L} \left[v_2^2 \cos \omega - v_1 v_2 \cos[\omega + (\theta_1 - \theta_2)] - \frac{v_2^2}{X_C} \right] \quad (4.4)$$

where ω is the loss term R/X_L , $\omega \ll 1$ as the losses are usually small, and

$$X_L = 2\pi fL \quad \text{and} \quad X_C = 1/2\pi fC$$

The solution of these non-linear equations can be obtained through an iterative technique. It must satisfy both Kirchoff's laws i.e. the current law, where the algebraic sum of flow at each node equal zero and the voltage law where the algebraic sum of voltages in a loop

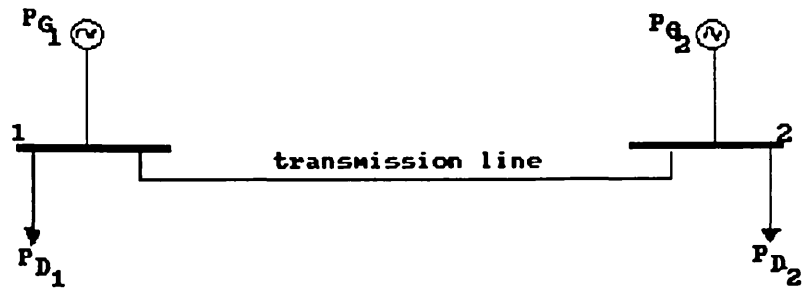


Fig 4.1 A two bus system

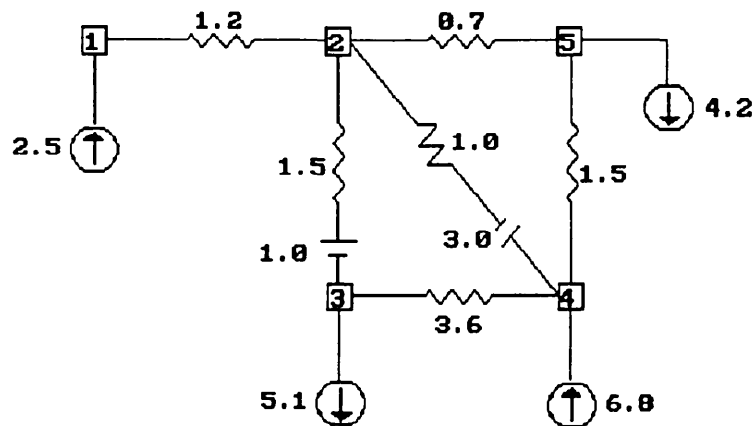


Fig 4.2 A small example system

equal zero.

The load flow solution methods used both the node and the loop frame of reference [49]. The criterion to choose a method is justified, mainly, by the convergence characteristics of this method in addition to the CPU time taken to converge, including the data preparation, compared to other methods.

4.3 THE LOAD FLOW SOLUTION

The AC and DC solutions for the load flow problem will be presented briefly and discussed in turn. This enables a better comparison between the methodology used by the NF methods and the formulation of conventional nodal methods.

4.3.1 THE DC LOAD FLOW

The DC load flow is an approximate solution for the static load flow equations where all line resistances are neglected (as $X \gg R$) and only active power flow is considered. For the two bus system of fig 4.1 the power transmitted along the line is

$$P_{12} = \frac{1}{R^2 + X^2} (R v_1^2 - R v_1 v_2 \cos \theta + X v_1 v_2 \sin \theta) \quad (4.5)$$

by neglecting the resistances, the active power flow becomes

$$P_{12} = \frac{1}{X} (V_1 \cdot V_2 \sin \theta) \quad (4.6)$$

$$\approx B_{12} \theta \quad (4.7)$$

when $V_1, V_2 \approx 1$ and $\theta = \theta_1 - \theta_2$ is small ($\sin \theta \approx \theta$). In a matrix form the DC load flow formulation is

$$[P] = [B].[\theta] \text{ or } [\theta] = [B]^{-1} . [P] = [X].[P] \quad (4.8)$$

The equations are linear and there is no need for iterative techniques. The line flows obtained from a DC load flow compare well on lightly loaded circuits with those obtained with a full AC load flow.

The DC load flow is used for applications where the system engineer can exchange the degree of accuracy with the speed of the solution. In other words, problems like the contingency analysis requiring fast solution methods with satisfactory accuracy, are likely candidates for the DC load flow.

In the next section the DC load flow problem is formulated in a structure amenable to a solution by a NF method. The formulation is based on the fact that the DC load flow problem is a power transportation problem in a network of busses and branches.

4.3.2 THE NF AS A DC LOAD FLOW

In this simplified solution, the problem becomes to determine approximately the real power and the phase angle distribution in an electric system. The solution technique consists of two stages: At the first stage the demand P_{Di} is satisfied by sending flows f_{ij} 's across the network from generating nodes to load nodes. During this stage, only Kirchoff's current law is met and the computation is done by neglecting all impedances and voltage sources of the branches. For the example system of fig 4.2, the first stage solution gives the following line flows (fig 4.3):

$$f_{12} = 2.5, f_{23} = 2.5, f_{24} = 0, f_{25} = 0, f_{34} = -2.6$$

$$\text{and } f_{45} = 4.2$$

The second stage of the solution modifies the above flow distribution so as to satisfy Kirchoff's voltage law. By superposing the circulating loop flows, we get a set of simultaneous linear equations where the problem becomes to solve this set of equations and compute the loop flows; then the line flows are corrected accordingly.

Consider the independent loops set chosen in fig 4.4 (the final solution is independent of the loops determination as the set of loops cannot always be determined uniquely in a network), it gives the following simultaneous linear equations with respect to the loop flows ($\Delta f_1, \Delta f_2$):
for loop 1

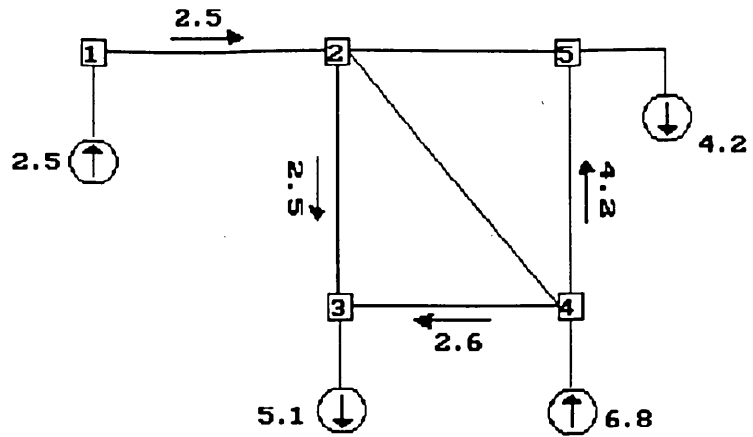


Fig 4.3 First stageline flows

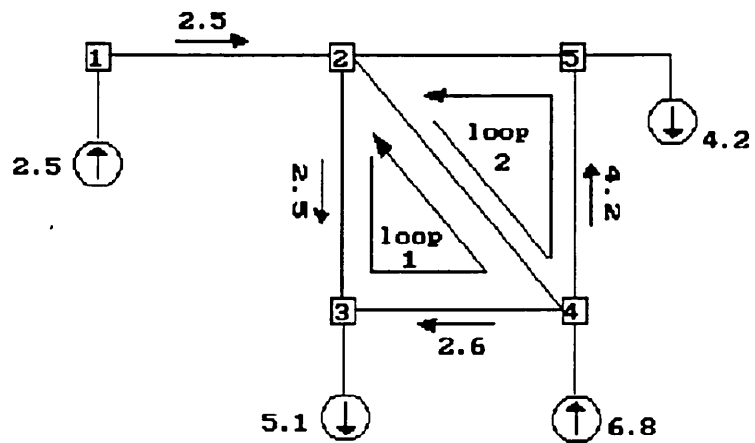


Fig 4.4 The independent loops

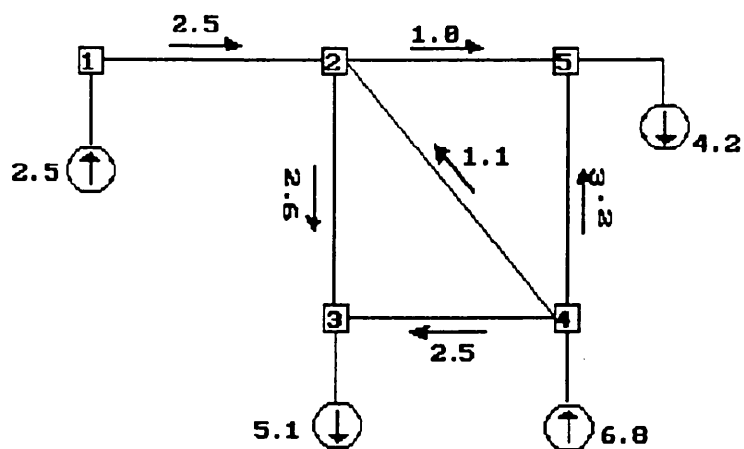


Fig 4.5 Final flow distribution

$$1.5(2.5+\Delta f_1) + 3.6(-2.6+\Delta f_1) + 1.0(\Delta f_1 - \Delta f_2) = -4 \quad (4.9)$$

for loop 2

$$0.7(\Delta f_2) + 1.0(\Delta f_2 - \Delta f_1) + 1.5(4.2+\Delta f_2) = 3 \quad (4.10)$$

In a matrix form this can be stated as:

$$\begin{bmatrix} 6.1 & -1.0 \\ -1.0 & 3.2 \end{bmatrix} \begin{bmatrix} \Delta f_1 \\ \Delta f_2 \end{bmatrix} = \begin{bmatrix} 1.61 \\ -3.3 \end{bmatrix} \quad (4.11)$$

The 2x2 square matrix is the loop connectivity matrix whose diagonal elements are the sum of branches impedances contained in a loop; and the off-diagonal elements are the mutual impedances of neighboring loops. The right hand side matrix of equation 4.11 represents the difference between the voltage sources and the voltage drops due to the first stage's flows i.e. eq 4.11 can be written as

$$Z \Delta f = L^t V - Z f_o \quad (4.12)$$

where Z is the loop connectivity matrix

L^t is the transpose of the branch to loop incidence matrix

V is the voltage source of the branches

f_o is the flow distribution obtained at stage 1.

$$\text{Therefore} \quad \Delta f = (L^t V - Z f_o) Z^{-1} \quad (4.13)$$

and the final line flows becomes

$$f = f_o + \Delta f \quad (4.14)$$

Figure 4.5 shows the final flow distribution.

Using the NF algorithm to solve the flow distribution problem is a fairly simple and straight forward task. The models adopted, for the generators, the load and the transmission lines, are the same as presented in chapter 3. The only difference is that there is no cost term present i.e. the demand is satisfied from generating nodes arbitrarily. In other words, the formulation of the NF, in this case, assumes that the objective function equals zero. Recalling the mnemonic device for the OKA, this means that the arc can only be in out-of-kilter states β_1 or β_2 , or in-kilter state β (fig 4.6 illustrates the new mnemonic device for the algorithm while the valid displacements are shown in fig 4.7).

The algorithm attempts to find the *flow augmentation path* to maintain flows within the limits when satisfying the demand; otherwise no feasible solution exists. This constitutes the first stage of the DC load flow solution. At the second stage, the independent loops are determined and the simultaneous linear equations are solved. The flow distribution obtained from the NF is modified accordingly to give a zero voltage sum around the independent loops of the network. As the real power through any branch is proportional to the difference of phase angles between both ends of the branch, the phase angles can be easily calculated using any arbitrary tree of the network (the phase angle does not depend upon an alternate selection of a tree). Table 4.1 illustrates the correspondence between the NF model and the DC load flow [46].

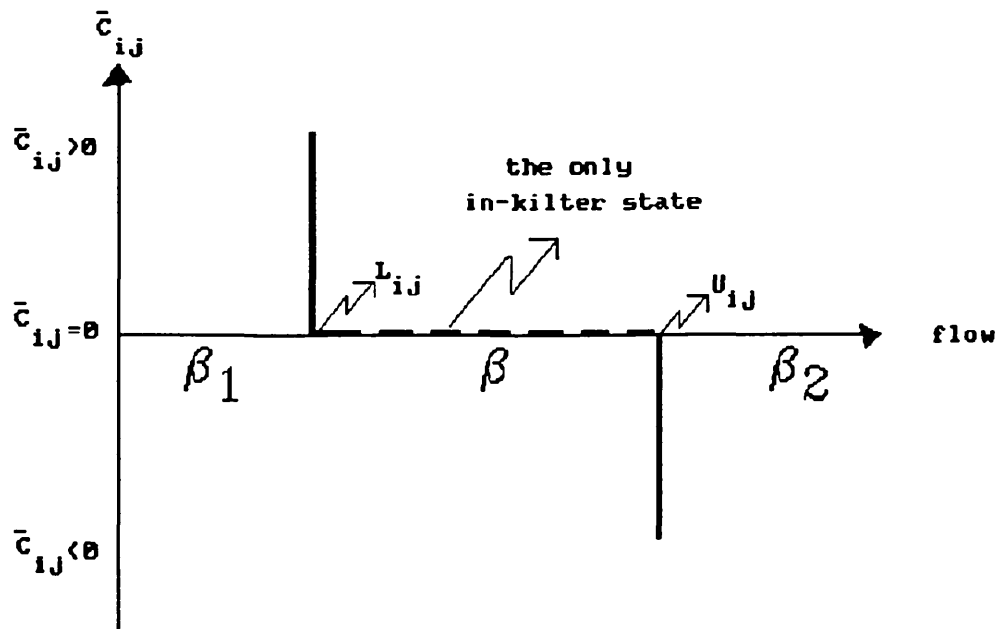


Fig 4.6 New mnemonic device for the OKA

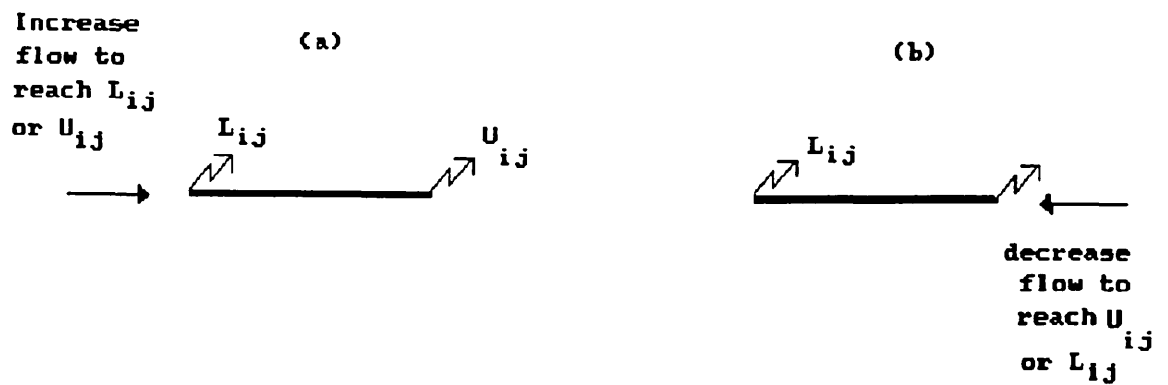


Fig 4.7 New valid displacements

The flow chart of the method is presented in fig 4.8.

NF technique		DC load flow
FLOW	————→	REAL POWER FLOW
POTENTIAL	————→	PHASE ANGLE
IMPEDANCE	————→	REACTANCE OF BRANCH
VOLTAGE SOURCE	————→	PHASE SHIFTER
NODE INJECTION	————→	AMOUNT OF GENERATION OR LOAD

Table 4.1 The correspondence between the NF
and the DC method

The algorithm developed was applied to the IEEE 14-bus (fig 4.9) and 24-bus systems. Table 4.2 shows the active power flows obtained by the NF and DC method, for the 24-bus system.

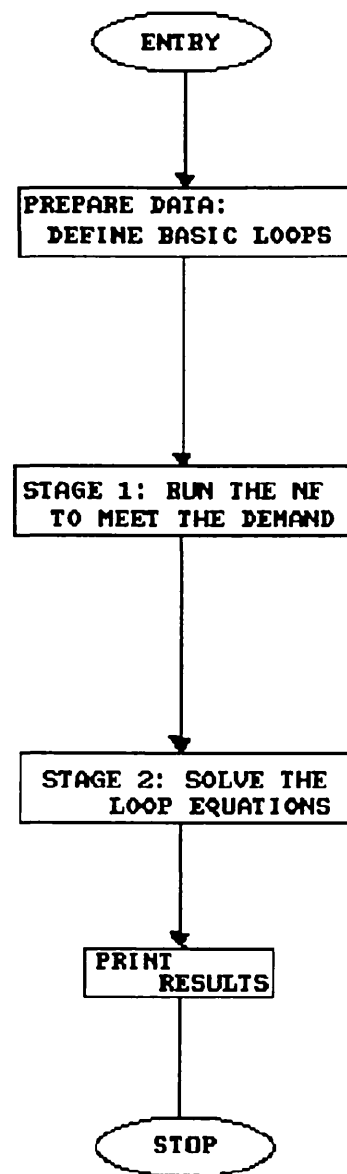


Fig 4.8 The flowchart for the DC load flow

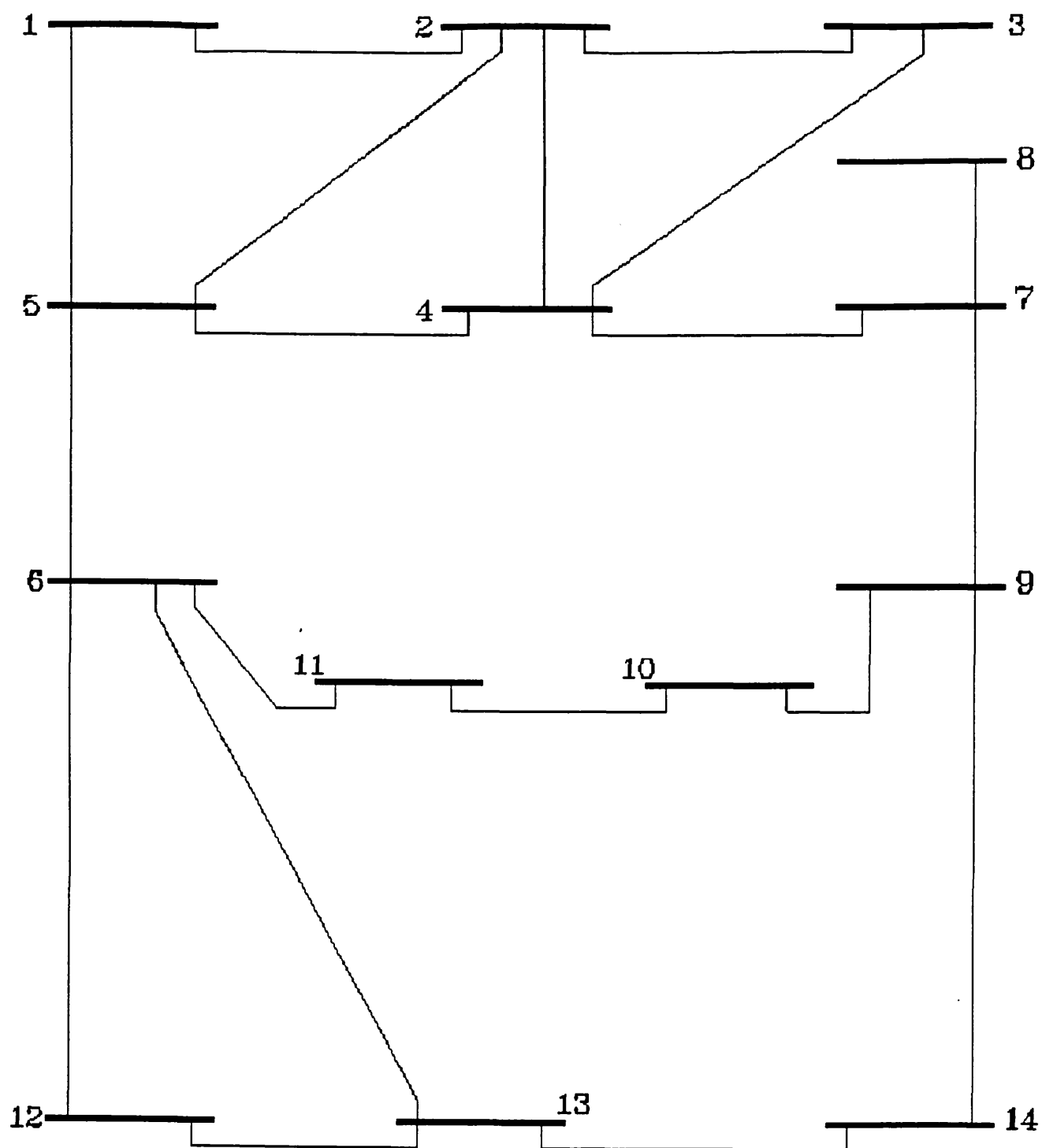


Fig 4.9 The IEEE 14-bus system

NETWORK FLOW				DC LOAD FLOW
LINE	S	T	LINE FLOW	LINE FLOW
1	1	2	0.868	0.914
2	1	3	-0.042	-0.289
3	1	5	0.132	0.335
4	2	4	0.178	0.087
5	2	6	0.120	0.257
6	3	9	-0.125	0.144
7	3	24	-1.716	-2.233
8	4	9	-0.562	-0.653
9	5	10	-0.577	-0.375
10	6	10	-1.239	-1.103
11	7	8	0.750	0.750
12	8	9	0.168	-0.599
13	8	10	-1.128	-0.361
14	9	11	0.011	-1.260
15	9	12	-2.28	-1.597
16	10	11	-2.154	-1.726
17	10	12	-2.74	-2.064
18	11	13	-0.294	-1.100

Table 4.2a The real power flows for the
IEEE 24-bus system

NETWORK FLOW				DC LOAD FLOW
LINE	S	T	LINE FLOW	LINE FLOW
19	11	14	-1.848	-1.885
20	12	13	-1.376	-0.505
21	12	23	-3.644	-3.156
22	13	23	-3.312	-3.247
23	14	16	-3.788	-3.825
24	15	16	-0.320	-0.740
25	15	21	-4.265	-4.363
26	15	24	1.716	2.233
27	16	17	-2.604	-2.507
28	16	19	-0.124	-0.679
29	17	18	-1.362	-1.242
30	17	22	-1.278	-1.265
31	18	21	-0.956	-0.872
32	19	20	-1.934	-2.489
33	20	23	-3.214	-3.769
34	21	22	-1.521	-1.535

Table 4.2b The real power flows for the
IEEE 24-bus system

The NF solution is mainly decided by the flow distribution reached during the first stage of the algorithm, which in turn depends on the node-to-arc ordering of the network i.e. the priority to carry a certain amount of flow is for the arc that comes on the top of the node-to-arc table, and so on (this is only encountered when the cost term is omitted from the formulation; otherwise the priority is given for the arc that contributes most to the cost minimization process). This is the fact responsible for the slight deviation in line flows and phase angles. The computational time (32ms on MicroVAX II) for the NF algorithm is less than the time taken by the conventional DC load flow (an exact comparison should take into consideration the programming language used and any possible code improvement [50]).

The simplified DC solution, although it models only active power, is very useful for the analysis of the effect of overloading of transmission lines and transformers, for reliability evaluation and for system expansion planning, etc. For problems requiring a complete model and full representation of the power system, the reactive power must be included in the formulation; this is discussed in section 4.3.4 that addresses the AC load flow solution.

4.3.3 CONTINGENCY ANALYSIS

Contingency analysis is used to predict the steady state conditions following line or generation outages. Such analysis is performed regularly on a periodic basis to cater for network status changes. It is carried out primarily through repeated load flow solutions for each of a list of a potential component failures. If the loss of a line and/or generator would result in the overload of another line, then the system is said to be vulnerable ; a condition which should be quickly detected for possible corrective rescheduling actions in operation or for system redesign in planning.

The comprehensive contingency analysis techniques must consider the impact of voltage effects as well as line flow effects: this can only be done accurately and comprehensively using a full AC load flow. However, the computation time taken by AC load flows does not permit the consideration of large number of contingencies. Recently, contingency screening methods for voltage security analysis have been reported in the literature which use approximate network solutions [51]; further work is still required for satisfactory results [52]. Meanwhile security analyses are performed with DC load flows. Thus, the approximate DC solution is adopted, as it provides satisfactory speed and accuracy for on-line applications (although DC solutions do not model reactive powers and voltages, Knight [53] suggested that DC methods could give

an approximate assessment of reactive powers and voltage magnitudes).

The NF algorithm developed to solve the DC load flow has been used to study the effect of line outages on the IEEE 24-bus system; Table 4.3 reports on the amount of load shedding required, relative to the loss of a line. The study was carried out on all the lines of the system, excluding line N°11 which if considered would island Bus N°7; as the line limits are very large, they were reduced as follows: 175→100, 400→250 and 500→350.

LINE OUTAGE	LOAD SHEDDING
No	MW
1	69
5	36
6	116
8	67
10	67
11	75
19	102
21	317
22	317
23	358
24	133
25	197
27	337
32	189
33	317

TABLE 4.3 The load shedding caused
by line outages

4.3.4 THE AC LOAD FLOW

The formulation of the AC load flow problem is very well known [48,54] and it will be treated briefly, in this section, to enable a comparison later on with a network flow formulation of the problem.

The variables in the static load flow equations (4.1)-(4.4) can be classified into three categories: the uncontrollable variables which are beyond the control of the system engineer such as the demand variables P_D and Q_D , the state variables that describe the state of the system (the phase angle θ and the voltage magnitude "v") and the control variables P_G and Q_G (the active and reactive generation). In a vector form these variables are defined, for n-bus system, as follows:

$$\begin{array}{ll} \text{the uncontrollable variables} & P = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_n \end{bmatrix} \triangleq \begin{bmatrix} P_{D1} \\ Q_{D1} \\ \vdots \\ P_{Dn} \\ Q_{Dn} \end{bmatrix} \\ \\ \text{the state variables} & X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} \triangleq \begin{bmatrix} \theta_1 \\ v_1 \\ \theta_2 \\ v_2 \\ \vdots \\ \theta_n \\ v_n \end{bmatrix} \end{array}$$

the control variables

$$U = \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_n \end{bmatrix} \stackrel{\Delta}{=} \begin{bmatrix} P_{G_1} \\ Q_{G_1} \\ \vdots \\ P_{G_n} \\ Q_{G_n} \end{bmatrix}$$

The solution procedure consists of specifying :

- (1)- The active and reactive generation but leaving the generation at the slack bus unspecified to satisfy the losses which are not yet known.
- (2)- The voltage magnitude $v_1 = 1$ pu and the phase angle at the slack bus $\theta_1 = 0$.
- (3)- The active and reactive demand which is assumed known at all buses. Then, an iterative procedure will solve the static load flow equations for the unknowns v 's, θ 's, P_{G_1} and Q_{G_1} . Also, the line powers are computed after the solution convergence.

4.3.5 THE NF AS AN AC LOAD FLOW

The AC load flow solution by the NF technique decouples the active power/phase angle part and the reactive power/voltage magnitude part. The algorithm iterates between the two parts of the problem until convergence is reached. The electric power transmitted by a transmission line can be written as:

$$\begin{aligned}
S_{12} &= P_{12} + jQ_{12} = V_1 \cdot I^* = V_1 \left(\frac{V_1^* - V_2^*}{Z^*} \right) \\
&= \frac{v_1^2 - v_1 v_2 \exp j\theta}{R - jX} \quad (\theta = \theta_1 - \theta_2, v \equiv |V|) \quad (4.15)
\end{aligned}$$

When separating the real and imaginary parts, equation (4.15) can be rewritten :

$$P_{12} = \frac{1}{R^2 + X^2} (R v_1^2 - R v_1 v_2 \cos \theta + X v_1 v_2 \sin \theta) \quad (4.16)$$

$$Q_{12} = \frac{1}{R^2 + X^2} (X v_1^2 - X v_1 v_2 \cos \theta - R v_1 v_2 \sin \theta)$$

Since $\cos \theta = 1 - 2\sin^2(\theta/2)$, equation (4.16) becomes

$$P_{12} = \frac{1}{Z^2} \left[X v_1 v_2 \sin \theta + R v_1 (v_1 - v_2) + 2 R v_1 v_2 \sin^2(\theta/2) \right] \quad (4.17)$$

$$J_{12} = \frac{1}{Z^2} \left[X (v_1 - v_2) - R v_2 \sin \theta + 2 X v_2 \sin^2(\theta/2) \right]$$

$$\text{where} \quad J_{12} = Q_{12} / v_1 \quad \text{and} \quad Z^2 = R^2 + X^2$$

Now if $\theta \ll 1$ rad (where $\theta = \theta_1 - \theta_2$) then we can say

$$\sin \theta \cong \theta - \frac{\theta^3}{6}$$

and (4.17) can be restated as:

$$P_{12} = f_{\theta_{12}} + T_{\theta_{12}} \quad (4.18)$$

$$J_{12} = f_{v_{12}} + T_{v_{12}}$$

where

$$f_{\theta_{12}} = (\theta_1 - \theta_2) / Z_{\theta_{12}} \quad (4.19)$$

$$f_{v_{12}} = (v_1 - v_2) / Z_{v_{12}}$$

and

$$Z_{\theta_{12}} = \frac{Z^2}{X \cdot v_1 \cdot v_2}$$

$$\begin{aligned}
Z_{v_{12}} &= Z^2/X \\
\text{also } T_{\theta_{12}} &= v_1(v_1 - v_2) \cdot R/Z^2 + \\
&\quad v_1 v_2 (\theta_1 - \theta_2)^2 \cdot R/2Z^2 - v_1 v_2 (\theta_1 - \theta_2)^3 \cdot X/6Z^2
\end{aligned} \tag{4.20}$$

$$\begin{aligned}
T_{v_{12}} &= -v_2(\theta_1 - \theta_2) \cdot R/Z^2 + \\
&\quad v_2(\theta_1 - \theta_2)^2 \cdot X/2Z^2 + v_2(\theta_1 - \theta_2)^3 \cdot R/6Z^2
\end{aligned} \tag{4.21}$$

It should be noted, however, that Z_θ is not an impedance, yet the terminology has been used to keep the consistency between the θ and the v parts of the formulation. Also the term $\sin^2\theta/2$ has been neglected as $|\theta| \ll 1$.

As is well known, the real power flowing through any branch depends essentially on the phase angle difference between the terminating nodes of the branch, and the reactive power depends on the difference of the voltage magnitude between the pairs of nodes. This way of thinking is the basis of the ac flow method which enables a decoupled active/reactive solution to be obtained.

For the active part: As $|\theta_1 - \theta_2| \ll 1$ and $X \gg R$, The real part of equation (4.18) is mainly determined by the first term f_θ ($T_{\theta_{12}} \ll f_{\theta_{12}}$). Thus, if an initial assumption is made for the voltage magnitudes v 's and the phase angles θ 's, then T_θ 's can be calculated; and if P_i is the original active power injection at bus i , the active power injection at bus i becomes

$$\varphi_i = P_i - \sum T_{\theta_{ij}} \quad \text{for all } ij \text{ connected to } i \quad (4.22)$$

The Network Flow solution evaluates the flow distribution $f_{\theta_{ij}}$'s due to the active power injections φ_i 's. The same procedure presented in section 4.2.2. is used where both Kirchoff's laws are considered. Since the flow distribution is now known, the phase angles can be computed by substituting f_{θ} 's in eq 4.19, and the θ 's values updated to be used in the reactive part of the solution.

For the reactive part: Similarly, the imaginary current injection is calculated from

$$\tau_i = Q_i/v_i - \sum T_{v_{ij}} \quad \text{for all } ij \text{ connected to } i \quad (4.23)$$

which is used as a nodal injection. The NF algorithm computes the imaginary current flow distribution f_v and then the voltage magnitudes are updated from eq 4.19.

The iterative procedure continues until the convergence is reached (within a certain tolerance); the flowchart for the NF algorithm, developed to solve the full AC load flow calculation is shown in fig 4.10. It should be noted, however, that in the reactive part we have used the imaginary component of the current injection instead of the reactive injection; this is because reactive power cannot be treated as an injection. The correspondence of variables and constants between the network flow technique and the ac flow method is illustrated in table 4.4.

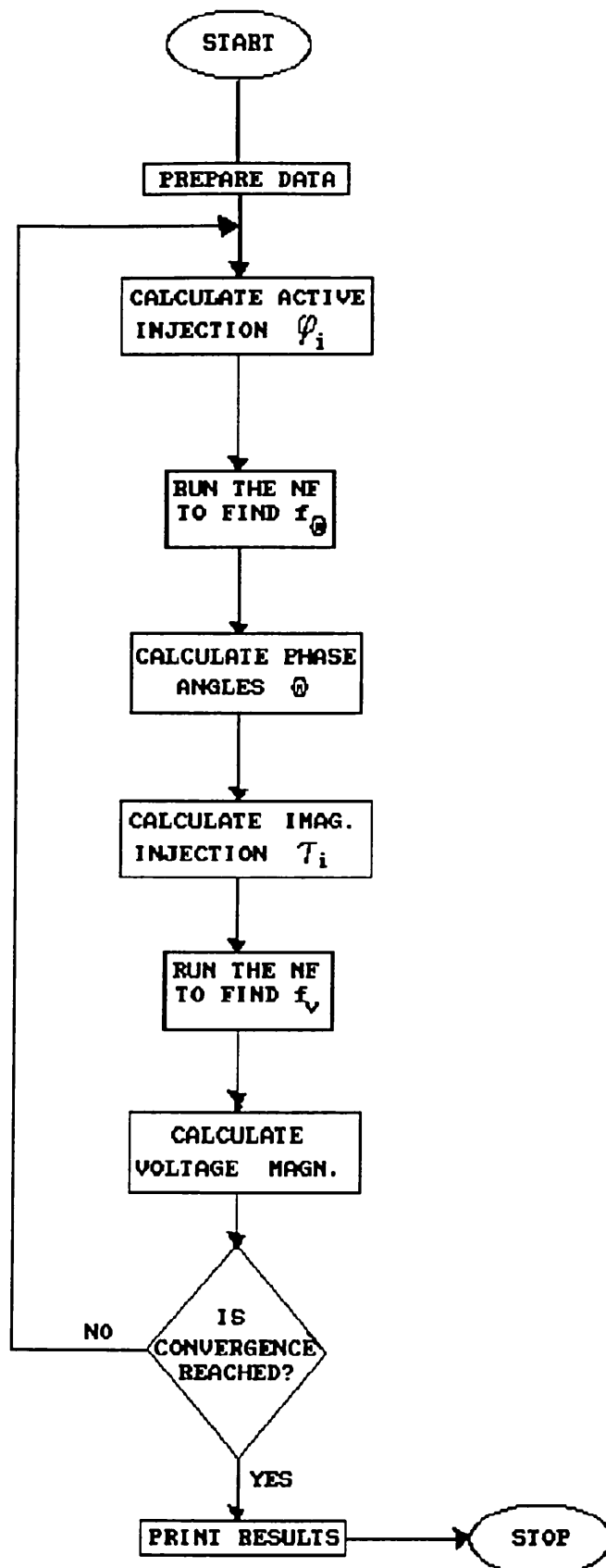


Fig 4.10 The flowchart for the AC load flow

The NF algorithm adopts the same lines and transformer models (equivalent π circuit of fig 4.11) used by conventional methods [54]; while the capacitive line effects are taken into consideration by computing the current injection due to the half-susceptance. On the other hand, when a voltage control bus (PV) is present in the system, the reactive power generation is checked against the limits: if it hits a limit, the reactive power generation is fixed at the violated limit and the bus voltage magnitude is left free to vary (i.e. the bus becomes a PQ bus); otherwise the bus voltage magnitude takes its specified value.

FLOW TECHNIQUE		AC FLOW METHOD	
		<u>θ NETWORK</u>	<u>v NETWORK</u>
FLOW	→	FLOW f_{θ}	→ FLOW f_v
POTENTIAL	→	PHASE ANGLE θ_i	→ V MAGNITUDE v_i
IMPEDANCE	→	IMPEDANCE Z_{θ}	→ IMPEDANCE Z_v
VOLTAGE SOURCE	→	PHASE SHIFTER	→ V MAG OF PV BUS OR TCUL SET
NODE INJECTION	→	MODIFIED REAL COMPONENT OF POWER AT NODE	→ MODIFIED IMAG COMPONENT OF CURRENT

Table 4.4 The correspondence between the network flow and ac method

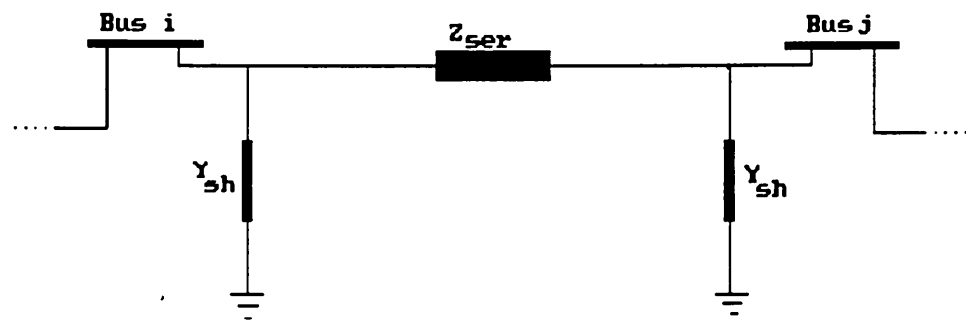


Fig 4.11 Equivalent π circuit

The difference, however, between the NF method and the conventional nodal methods is that the NF algorithm computes the line flows f_{θ} 's and f_v 's first, then the phase angles and the voltage magnitudes, responsible for this flows, are calculated. As for the conventional nodal methods, the line flows are only calculated after convergence is reached. The NF solution, using the loop frame of reference, has its advantages over nodal methods in terms of both computational speed and core memory requirements [49,55], particularly when applied to systems having less number of loops than the number of nodes (Zhang and others [49,55] used a loop based method for automatic contingency selection).

The algorithm developed has been applied to various systems such as the 9-bus system of fig 4.12 and the IEEE 14-bus system of fig 4.9. Table 4.5 illustrates the converged voltage magnitudes and phase angles obtained by both the NF algorithm and the FDLF. The NF solution converged in 6 iterations, which took 77ms on MicroVax II, and found to be faster than the FDLF (90ms). The slight difference in the results is subject to the same comment of section 4.3.2. The first stage flow distribution is influenced by the node-to-arc table where the priority to carry the flow is for the arc on top of the table, and so on; in the second stage, Kirchoff's Voltage Law is satisfied by modifying the first stage's flow. Consequently, it is the node-to-arc table that influences the final flow pattern.

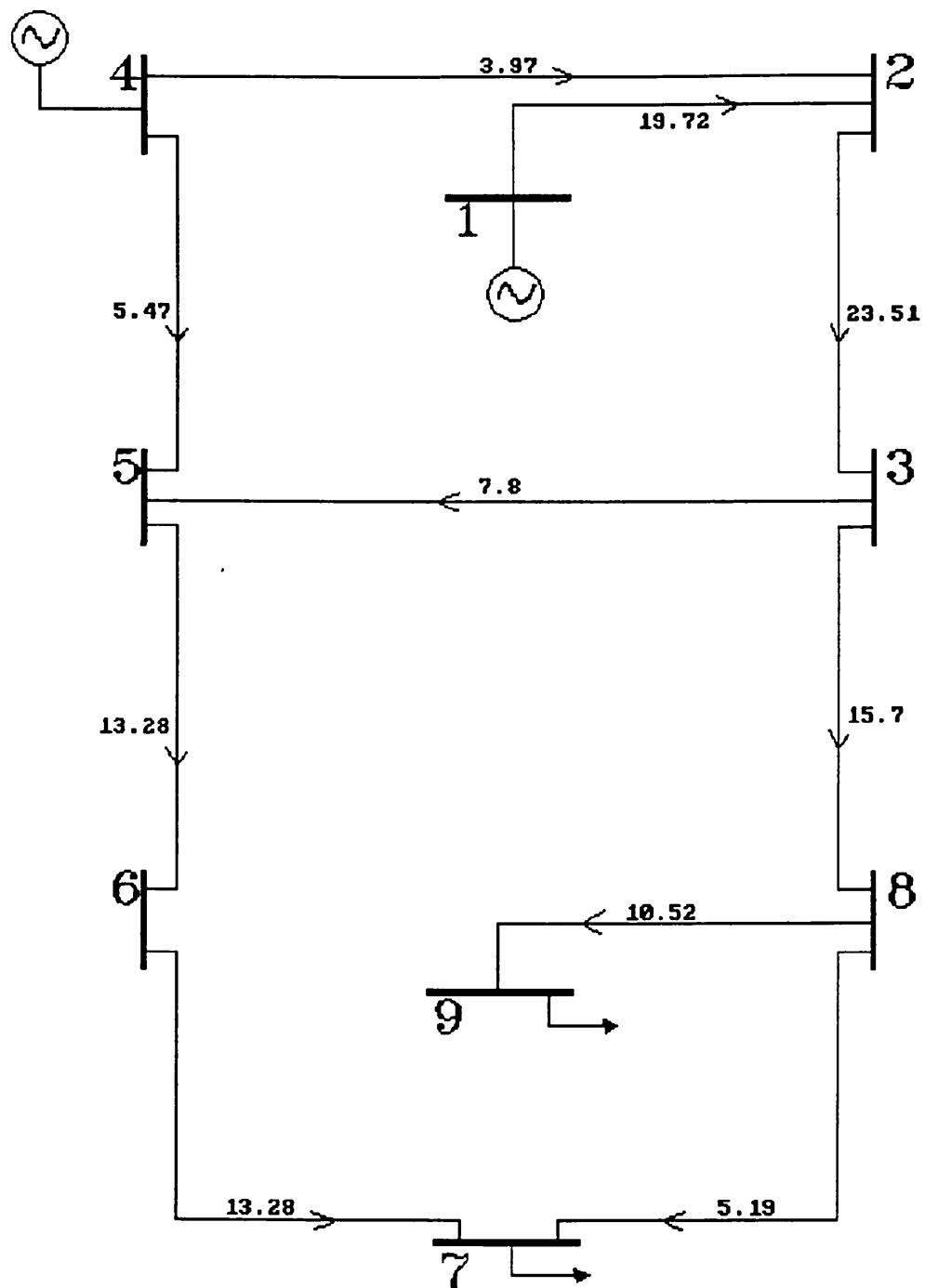


Fig 4.12 Real power flows for the 9-bus system

NF SOLUTION			FDLF SOLUTION	
BUS	CPU TIME=77ms		CPU TIME=90ms	
Nb	$ V _{pu}$	θ°	$ V _{pu}$	θ°
1	1.06	0.0	1.06	0.0
2	1.041	-4.93	1.045	-4.97
3	1.01	-12.83	1.01	-12.71
4	1.00	-10.15	1.01	-10.36
5	1.01	-8.7	1.02	-8.9
6	1.0	-14.78	0.98	-15.0
7	1.0	-13.26	1.01	-14.0
8	1.05	-13.26	1.05	-14.0
9	0.98	-14.96	0.99	-16.0
10	0.98	-15.25	0.98	-16.1
11	0.99	-15.18	0.98	-15.7
12	0.98	-15.87	0.97	-16.0
13	0.98	-15.98	0.966	-16.2
14	0.97	-16.27	0.962	-17.2

TABLE 4.5 The converged voltage magnitudes
and phase angles

The computation time of the NF solution could be greatly improved perhaps up to 10 times, as found by Singh [10], through a better implementation of the OKA. This is discussed in chapter VII.

4.4 ALTERNATIVE AC SOLUTION BY NF METHODS

It was made clear in the previous sections of this chapter that the NF formulation of the DC and AC load flow calculation, does not make use of the optimization feature it possesses: no objective function was considered and consequently the criteria, for the flow distribution, was only to meet the load, without violating the flow constraints.

Another advantage of the NF method is that it is always possible to meet the demand while minimizing a certain objective function (cost of generation, losses, etc) rather than allocating arbitrarily the load to the generating sets in the system. Also, as the NF formulation of the AC load flow problem is divided into two parts (active and reactive), the algorithm is capable of handling two different objective functions; for instance, the active power flow distribution minimizes the total generation costs, while in the reactive part, a security objective function [56-58] could be considered (such as improvement of voltage profile, maximizing reactive reserve, minimum change in reactive power, etc [59]).

Although this means that, unlike other methods, the algorithm produces directly optimal results, it will not be without its cost in computation time. In the author's opinion this is not recommended as it takes from the NF methods their main feature i.e. the speed of the solution.

Practically this has been done and reported in the literature [47].

Barras et al [47] used a NF method to solve the AC load flow problem. The losses, taken as objective function, were minimized subject to the limits imposed on line flows and to the flow conservation constraint (Kirchoff's current law). As expected, the results were as accurate as those obtained when using the Newton-Raphson; as for the CPU time, the NF solution was much slower. Thus, the author believes that the NF algorithm developed to solve the load flow calculations is not suitable for base case solutions but for applications requiring fast solutions such as reliability evaluation and security analysis.

4.5 CONCLUSION

The potential applications of the OKA are problems that can be cast into a network structure and require a large computation time such as the security analysis. This includes the contingency analysis where line outages can be easily modeled by setting the upper and lower limits, of the tripped line, equal to zero. This ensures that no power will be sent over the tripped line. If the line belongs to one basic loop, the loop is removed from the computation. Yet, if it belongs to two neighbouring loops, the elements of these two loops are joined to form one single loop.

The OKA is capable of modeling the transfer of power into or out of a group of generators and lines which is defined as insecure under some conditions, from operational experience or following an off-line security study. The power transfer into or out of the group can be modeled by a single arc where the flow is confined to upper and lower bounds on the arc and consequently the security is enhanced (see section 6.6).

In conclusion, the network flow solution is not suitable for a base case load flow but it appears to have many advantages for problems where the load flow calculation is only part of a larger problem like contingency analysis and security assessment; nevertheless further work is still required on this aspect.

CHAPTER V

RELIABILITY ANALYSIS

5.1 HISTORICAL BACKGROUND

Reliability analysis is an important aspect of power system design. It involves consideration of service reliability requirements of loads to be supplied, defined as Loss-Of-Load-Probability (LOLP) i.e. the probability that the generating/transmitting capacity will be unable to supply the demand [60]. This problem is not at all new to power system engineers as it was defined as early as the 1930's by Dean [61]: "One of the really difficult problems faced by those responsible for planning electric supply systems is that of deciding how far they are justified in increasing the investment on their properties to improve the service reliability" (a planning function).

There are also many variations on the definition of reliability but a widely accepted form is as follows: *Reliability is the probability of a device performing its purpose adequately for the period of time intended under the operating conditions encountered* [62]. In case of a power system, this means the probability of supplying the customer's demand as reliably as possible (an operational planning function).

Reliability requirements of a system was introduced by many researchers in the 30's [63-65]. However, lack of data and limitation of computational facilities restricted

the numerical application of reliability techniques. Then, a major step forward was made by Calabrese [66] and Halperin & Alder [67] who presented a practical model and a practical index to measure the system adequacy using probabilistic methods. Their model was restricted to the adequacy of generating capacity to satisfy the load under the assumption of an adequate transmission system. This has continued until Billinton & Bhavaraju [68,69] applied probabilistic methods to bulk power supply evaluation.

With the rapid evolution in digital computer technology, the use of probabilistic techniques became more popular [70,71] and reliability evaluation methods are under continuous development [72-74]. They provide the system planner with the capability to evaluate the system reliability in terms of two indices:

1- *Loss-Of-Load-Probability (LOLP)*

2- *Expected-Demand-Not-Served (EDNS)*

The first index (LOLP) serves as determining the probability that the capacity of the generating/transmitting capacity remaining in service, following a loss of generation and/or transmission elements, is exceeded by the system load level, while the second index (EDNS) determines the expected value of the demand not served.

The evaluation of these two indices requires the inclusion of an AC load flow model in the analysis. This is an appalling thought, at the present time, as AC methods are computationally demanding. However, the incorporation

of a DC load flow model will sometimes lead to difficulties [75] as it requires the solution of a number of linear programs. Therefore, the research in reliability assessment led many to consider the use of simpler models for the electrical network. For this purpose, NF methods came into use for applications where they could be used to find critical minimal cuts in the network [80].

Reliability assessment is another major promising application for NF methods. They could be used for single and multi-area reliability calculation to identify bottlenecks in the network [76-80]. In this chapter, the method proposed by Sullivan [77] is implemented and the NF algorithm is applied to evaluate the LOLP. The presentation of the basic theory of this probabilistic method follows closely Sullivan's work.

5.2 BASIC PROBABILITY

The word "probability" is used to describe the likelihood of an event to happen or to prove correct (true). Mathematically, it represents a numerical index that can vary between zero, which defines an absolute impossibility, to unity which defines an absolute certainty. Then the scale of probability can be represented by figure 5.1.

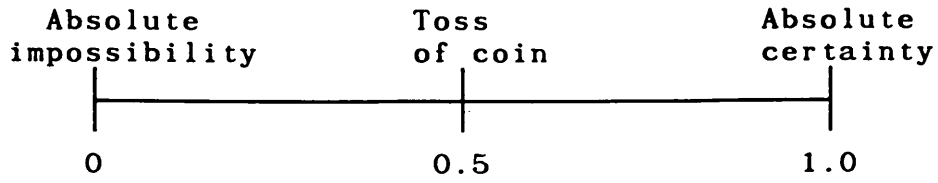


Fig 5.1 Probability scale

It is reasonable to assume that each event has two possible states: one can be described as the favourable or success and the other as the unfavourable or failure. Even if an event has more than two possible states it is often justified to group together those outcomes which can be called success and those which can be described as failures to end up eventually with two states. This enables the probability of success and failure to be written as:

$$\text{Prob(failure)} = \frac{\eta}{\chi + \eta} = q \quad (5.1)$$

$$\text{Prob(success)} = \frac{\chi}{\chi + \eta} = p \quad (5.2)$$

where η = expected failure rate,

χ = expected success rate

and $p+q = 1$.

In power system applications, Prob(failure) is known as the unit Forced Outage Rate (FOR) and transmission elements are described by a simple two-state model (unit up or unit down). A pictorial representation of the concept, using a Venn diagram, is shown in fig 5.2. In probability

theory the two states are said to be complementary if when one does not occur, the other must (i.e. $p+q=1$).

5.3 PROBABILISTIC RELIABILITY ASSESSMENT

We have assumed that each element in the system under study can reside in either of two states: the "0" state with probability q_m , in which it has no capacity and is out of service; or the "1" state with probability p_m , in which it has capacity U_m and is in service. The system then will have 2^E capacity states ξ_i , $i = 1, \dots, 2^E$ (E being the total number of system elements). The 2^E states must be decomposed into states that are acceptable and states that are rejected (or unacceptable). A state will be called acceptable if the maximum flow from the source node to the sink node is equal to the total demand, while rejected states are system capacity states ξ_i for which the demand P_D cannot be met, because of insufficient generation/transmission capacity. Recalling the definition of LOLP in sec 5.1, we can mathematically define it as:

$$LOLP \triangleq \sum f(\xi_i) \quad \xi_i = \text{all unacceptable states} \quad (5.3)$$

$$\text{where } f(\xi_i) = \prod_{m=1}^E f(v_m)$$

$$\text{and } f(v_m) = p_m \text{ if } v_m = 1 \quad (\text{element } m \text{ is in service})$$

$$f(v_m) = q_m \text{ if } v_m = 0 \quad (\text{element } m \text{ is out of service})$$

Therefore, if we consider a three bus system which has two generators (fig 5.3), then the system is said to have five elements. The upper limiting state of the system (denoted

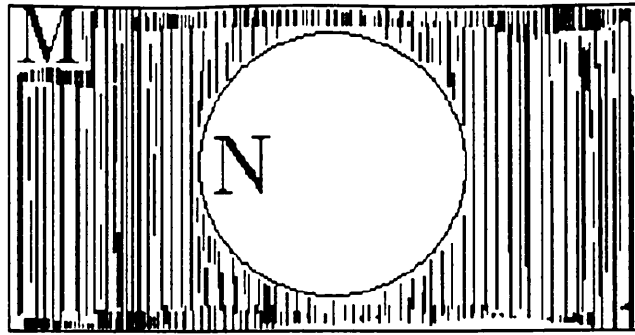


Fig 5.2 Pictorial representation of two state model

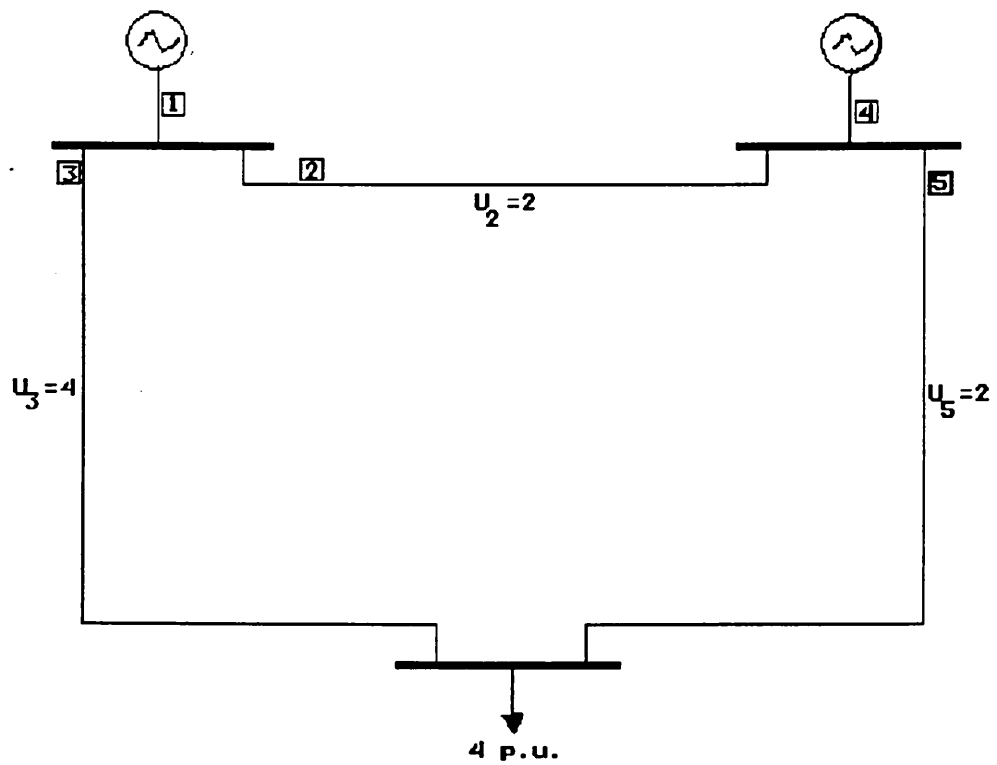


Fig 5.3 A three bus system

$\bar{\xi}$) is when all five elements are in service; hence

$$v_1 = v_2 = v_3 = v_4 = v_5 = 1$$

$$\text{and } f(v_1) = p_1, f(v_2) = p_2, f(v_3) = p_3, f(v_4) = p_4,$$

$$f(v_5) = p_5.$$

$$\text{Thus, } f(\bar{\xi}) = \prod_{m=1}^E f(v_m) = p_1 p_2 p_3 p_4 p_5 \quad (5.4)$$

where $\bar{\xi} = (1,1,1,1,1) \stackrel{\Delta}{=} \text{the upper limiting state.}$

Similarly, $\underline{\xi} = (0,0,0,0,0) \stackrel{\Delta}{=} \text{the lower limiting state.}$

$$\text{and } f(\underline{\xi}) = \prod_{m=1}^E f(v_m) = q_1 q_2 q_3 q_4 q_5 \quad (5.5)$$

In general the system can reside in any state

$$\xi_i = (v_1, v_2, \dots, v_m, \dots, v_E)$$

and the problem becomes that of determining whether these states are acceptable or rejected (unacceptable), as the LOLP is the probability of the rejected states.

5.3.1 STATE CLASSIFICATION

Initially, the power system is in an unclassified state and it is necessary to define the criteria based on which the classification into acceptable and rejected states is done. For this purpose, two capacity states are defined [77]: the first is called the upper critical state, denoted ξ^0 , such that any system state $\xi_i \geq \xi^0$ is acceptable; the second is the lower critical state ξ_0 and any system state $\xi_i < \xi_0$ is rejected. Obviously, there still exist some system states $\xi_0 \leq \xi_i < \xi^0$ unclassified because ξ^0 and ξ_0 fail to ascertain the acceptability of these states. Thus, we end up with three capacity states:

acceptable, rejected and unclassified. This is illustrated, using Venn diagram, in fig 5.4.

From the definition of LOLP we can say that the intermediate LOLP is the probability that the system will reside in subset $\mathcal{X}$; therefore $\text{LOLP} \triangleq f(\mathcal{X})$. Since

$$f(\mathcal{A}) + f(\mathcal{X}) + f(\mathcal{U}) = 1 \quad (5.6)$$

$$f(\mathcal{U}) = 1 - f(\mathcal{A}) - f(\mathcal{X}) \quad (5.7)$$

If $f(\mathcal{U})$ is too small (less than a tolerance factor ϵ), then there is no need for further decomposition. Otherwise, the decomposition process (into $\mathcal{A}$, $\mathcal{X}$ and $\mathcal{U}$) is repeated until the subset $\mathcal{U}$ is empty or its probability $f(\mathcal{U}) \leq \epsilon$. Consequently, all possible system states are inspected and defined whether they are acceptable or not.

5.3.2 DEFINING THE UPPER AND LOWER CRITICAL STATES

The basic idea behind determining the upper critical state ξ^o is a very simple one. The network flow algorithm is used to yield a feasible flow when all elements of the network are in service; then, the upper critical state for each element v_m^o is to be determined, since

$$\xi^o = (v_1^o, v_2^o, \dots, v_E^o)$$

Given the feasible flow pattern obtained, based on all elements being in service, the rules that apply to determine ξ^o are as follows:

$$\begin{aligned} \text{If } f_m = 0, \quad & \text{then } v_m^o = 0 \\ \text{If } 0 \leq f_m \leq U_m, \quad & \text{then } v_m^o = 1 \end{aligned} \quad (5.8)$$

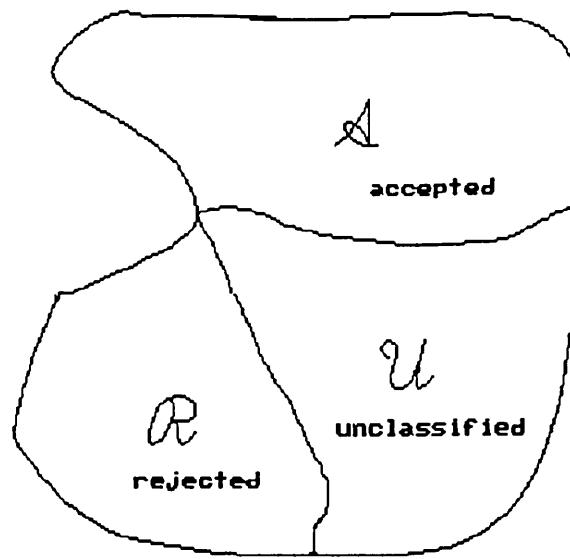


Fig 5.4 Venn diagram representing A , R and U

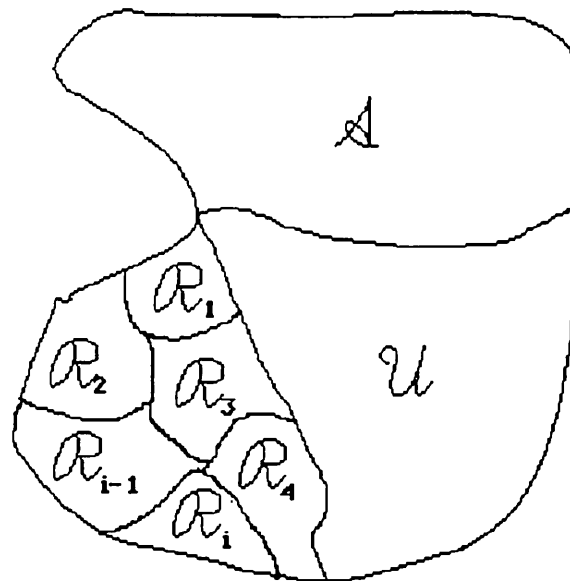


Fig 5.5 Decomposition of R

where f_m and U_m are the flow and the capacity of element m respectively. The probability of the acceptable states is then

$$f(\mathcal{A}_j) = \prod f(v_m \geq v_m^0) \quad (5.9)$$

where

$$\left[\begin{array}{ll} f(v_m \geq v_m^0) = 1.0 & \text{if } v_m^0 = 0 \\ f(v_m \geq v_m^0) = p_m & \text{if } v_m^0 = 1. \end{array} \right.$$

This procedure is easy to implement with the NF algorithm, and $\xi^0 = (v_1^0, v_2^0, \dots, v_E^0)$ can be determined. Clearly, any system state ξ_i that has additional elements in service than in (or equal to) the ξ^0 state should be an acceptable one as it implies that the demand can still be met; in other words, if one (or more) of the "0" state changes to "1" state, the resulting system state is acceptable, while system states ξ_i in which one (or more) of the "1" state changes to "0" are yet unclassified. Therefore

$$\xi_i \text{ is acceptable if } v_m \geq v_m^0 \quad m = 1, 2, \dots, E$$

Consider the five elements system of fig 5.3; when applying the NF algorithm, the following line flows have been obtained

$$f_1 = 2 \text{ p.u.}, f_2 = 2 \text{ p.u.}, f_3 = 0 \text{ p.u.}, f_4 = 2 \text{ p.u.}, f_5 = 2 \text{ p.u.}$$

Thus

$$v_1^0 = 1, v_2^0 = 1, v_3^0 = 0, v_4^0 = 1, v_5^0 = 1$$

and $\xi^0 = (1, 1, 0, 1, 1)$

Now, if for some reason element Nb 5 (for example) goes out of service the system state ξ_i becomes

$$\xi_i = (1, 1, 0, 1, 0)$$

and it is so far unclassified i.e. it is not yet known whether the system is capable of meeting the demand after the loss of element Nb 5. If however element Nb 3 is used to carry some flow, in addition to the elements already involved, its state changes to $v_1 = 1$ and ξ_i becomes

$$\xi_i = (1, 1, 1, 1, 1)$$

which does not affect the ability to satisfy the load.

The procedure to determine the lower critical state ξ_0 is a bit different than that of determining ξ^0 , yet equally easy. Initially, all elements of the network are assumed available i.e. starting with the upper limiting state $\bar{\xi} = (v_1, v_2, \dots, v_m, \dots, v_E)$ all $v_m = 1$, $m = 1, 2, \dots, E$. Then a contingency analysis based procedure is executed as follows:

1. Remove one element m from the network i.e. the element states are thus $v_m = 0$ and $v_i = 1$ $i = 1, 2, \dots, E$, $i \neq m$
2. The NF algorithm is applied to the network in its new state defined in 1 above, to produce a feasible flow pattern
3. Then, the lower critical state v_{m0} for each element is set according to the following rules:

$$\left[\begin{array}{l} \text{if the demand is met, then } v_{m0} = 0 \\ \text{if the demand is not met, } v_{m0} = 1 \end{array} \right. \quad (5.10)$$

4. Increase $m \rightarrow m+1$, go to step 1. The procedure is repeated until all elements are considered; by then ξ_0 is completely known.

The method described above tells us that any system state ξ_i in which $v_m < v_{m0}$ is an unacceptable state because the maximum flow through the system is less than the demand. Hence we have already defined both the upper and lower critical states ξ^0 and ξ_0 , respectively. The algorithm is ready to compute the intermediate LOLP (eq 5.3) which is the probability of all unacceptable system states $\xi_i < \xi_0$.

5.3.3 EVALUATION OF LOLP

The analytical method to calculate the LOLP appears to be ready from the theoretical point of view. However, practically speaking, the computation effort involved is large when one thinks of the number of system states to be inspected (for a 14-bus system with two generators, there exist 2^{16} possible states). Hence, a decomposition process [81] is adopted which allows the decomposition of subset $\mathcal{X}$ into non-overlapping subsets (fig 5.5). In doing so, the calculation of LOLP is performed using properties associated not with each system state $\xi_i \in \mathcal{X}$, but with subsets of $\mathcal{X}$. Obviously, the number of subsets $\mathcal{X}_j$ is far less than the number of states ξ_i ; therefore the computation effort is greatly reduced and the algorithm becomes faster.

As the LOLP is the probability of set $\mathcal{X}$, it can be

written as

$$\text{LOLP} = f(\mathcal{X}) = \sum_j f(\mathcal{X}_j)$$

where $\mathcal{X}_j$ represents a subset of $\mathcal{X}$. These subsets should be mutually exclusive such that no state ξ_i can satisfy the properties of more than one subset:

$$\xi_i \in \mathcal{X}_j \text{ if } v_j < v_{j0} \text{ and } v_m \geq v_{m0} \quad \text{for all } m < j$$

The above rule implies that all those states ξ_i that are unacceptable because of the loss of elements j are grouped in a subset $\mathcal{X}_j$. This subset $\mathcal{X}_j$ can only exist if element j is out of service (i.e. $v_j=0$ which, as stated in sec 5.3, has a probability q_j) and $v_m \geq v_{m0}$ for $m < j$; hence the total probability of $\mathcal{X}_j$ is

$$f(\mathcal{X}_j) = q_j \prod_{m < j} f(v_m \geq v_{m0}) \quad (5.11)$$

where

$$\left[\begin{array}{ll} f(v_m \geq v_{m0}) = 1.0 & \text{if } v_{m0} = 0 \\ f(v_m \geq v_{m0}) = p_m & \text{if } v_{m0} = 1 \end{array} \right.$$

Therefore, the intermediate LOLP becomes the sum of the subset probabilities, $f(\mathcal{X}_j)$, rather than the sum of the state probabilities, $f(\xi_i)$, and great simplification results by avoiding complete enumeration.

It should be noted, however, that the probability of the unclassified states $f(\mathcal{U})$ has been considered negligible; yet if $f(\mathcal{U})$ is not sufficiently small, then further classification of states $\xi_i \in \mathcal{U}$ into acceptable, and rejected states is necessary.

5.3.4 EXHAUSTING THE UNCLASSIFIED SUBSET $\mathcal{U}$

Once the probability of the unclassified states has been found relatively large, subset $\mathcal{U}$ has to be exhausted and $\xi_i \in \mathcal{U}$ should be classified. Figure 5.6 shows a pictorial illustration of subsets $\mathcal{A}$, $\mathcal{X}$ and $\mathcal{U}$ where $\mathcal{U}$ is bounded by ξ^o and ξ_o . Exhausting $\mathcal{U}$, diagrammatically, means that the gap between ξ^o and ξ_o is made narrower or even forced to disappear, as fig 5.7a and 5.7b show respectively. First, subset $\mathcal{U}$ is divided into smaller subsets $\mathcal{U}_j$ according to the following rule

$$\begin{aligned} & \xi_i = (v_1, v_2, \dots, v_m, \dots, v_E) \in \mathcal{U}_j \\ \text{if} & \quad v_{j0} \leq v_j < v_j^o \\ \text{and} & \quad \left[\begin{array}{ll} v_m \geq v_m^o & \text{for } m < j \\ v_m \geq v_{m0} & \text{for } m > j \end{array} \right. \quad \begin{array}{l} m = 1, 2, \dots, E \\ m \neq j \end{array} \end{aligned} \quad (5.12)$$

Secondly, having partitioned subset $\mathcal{U}$ into $\mathcal{U}_j$'s, each in turn will be subjected to the same classification procedure applied initially to the 2^E states. In other words, system states $\xi_i \in \mathcal{U}_j$ are grouped into acceptable, unacceptable and unclassified subsets and the overall method, explained in the previous sections, has to be repeated for each subset $\mathcal{U}_j$ (defining $\bar{\xi}(\mathcal{U}_j)$, $\underline{\xi}(\mathcal{U}_j)$, $\xi^o(\mathcal{U}_j)$ and $\xi_o(\mathcal{U}_j)$). This is further illustrated in the flow charts of the subroutines for the reliability assessment algorithm, shown in figures 5.8, 5.9 and 5.10.

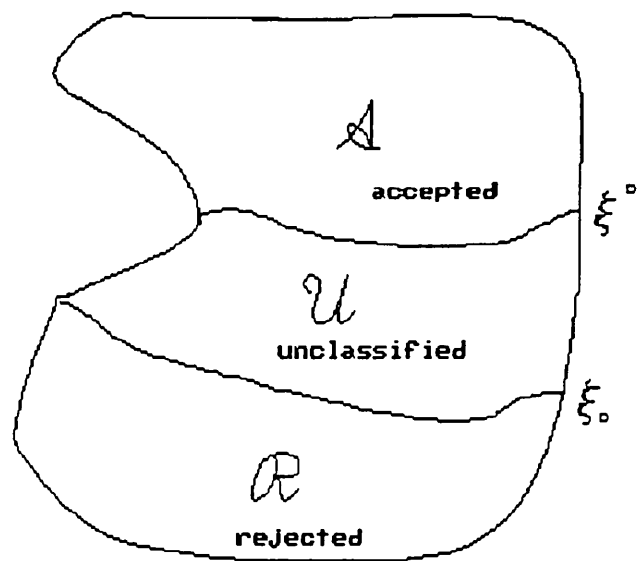


Fig 5.6 A pictorial illustration
of U boundary's

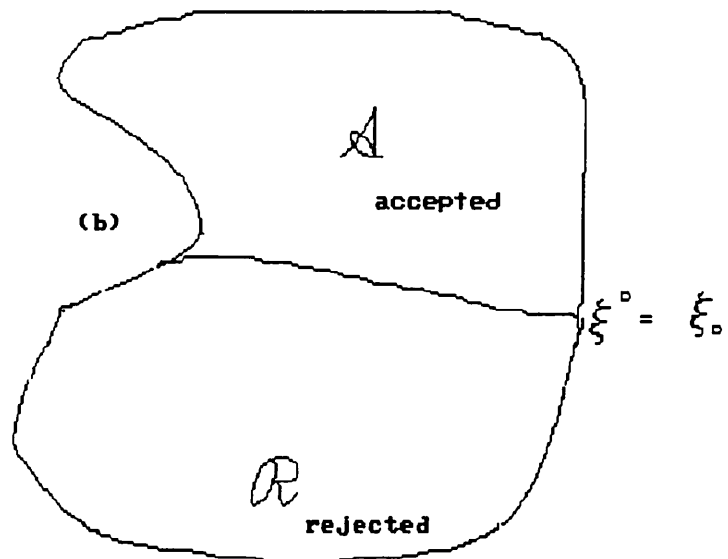
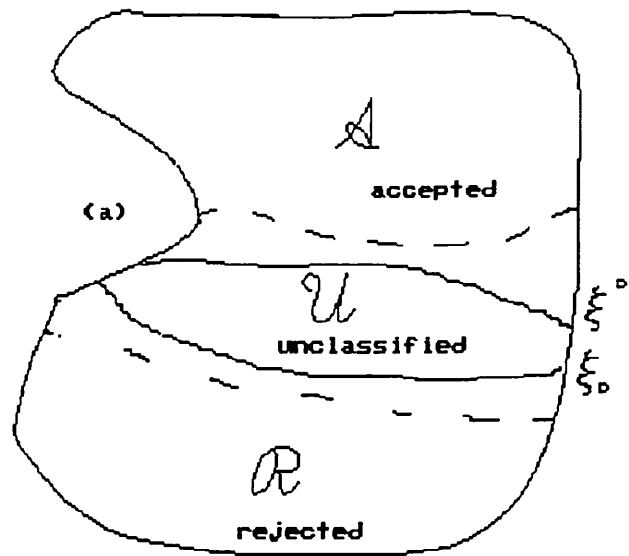


Fig 5.7 Exhausting subset U

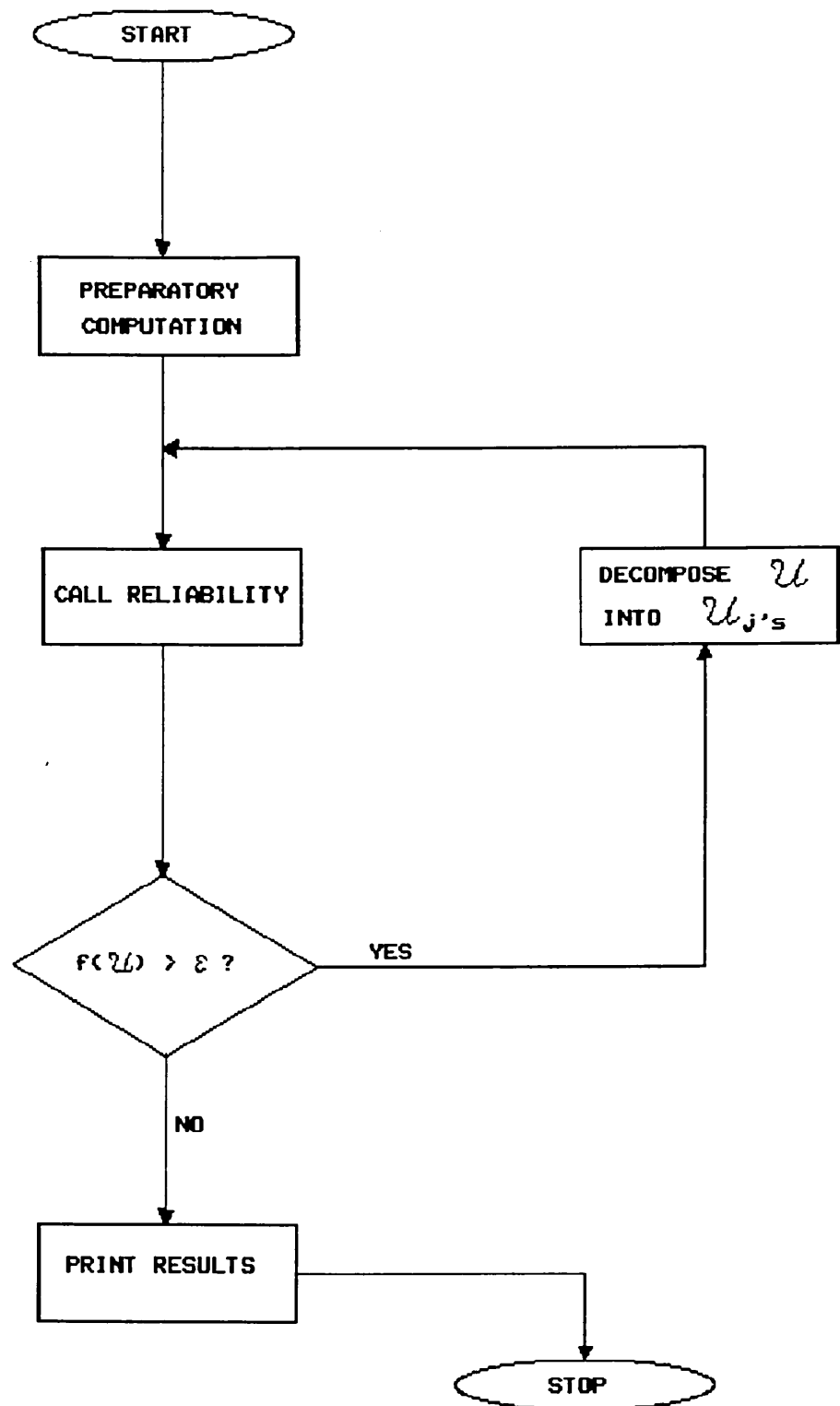


Fig 5.8 Flowchart of the Reliability analysis

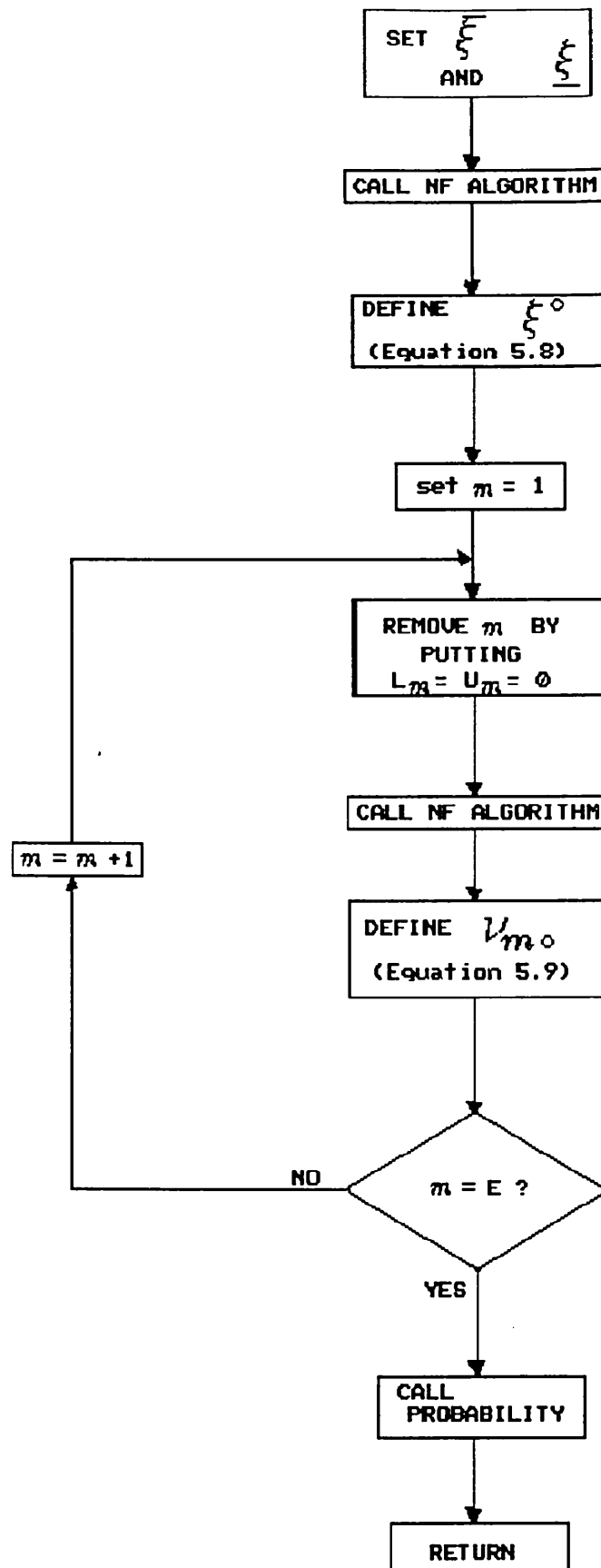


Fig 5.9 Flowchart of subroutine RELIABILITY

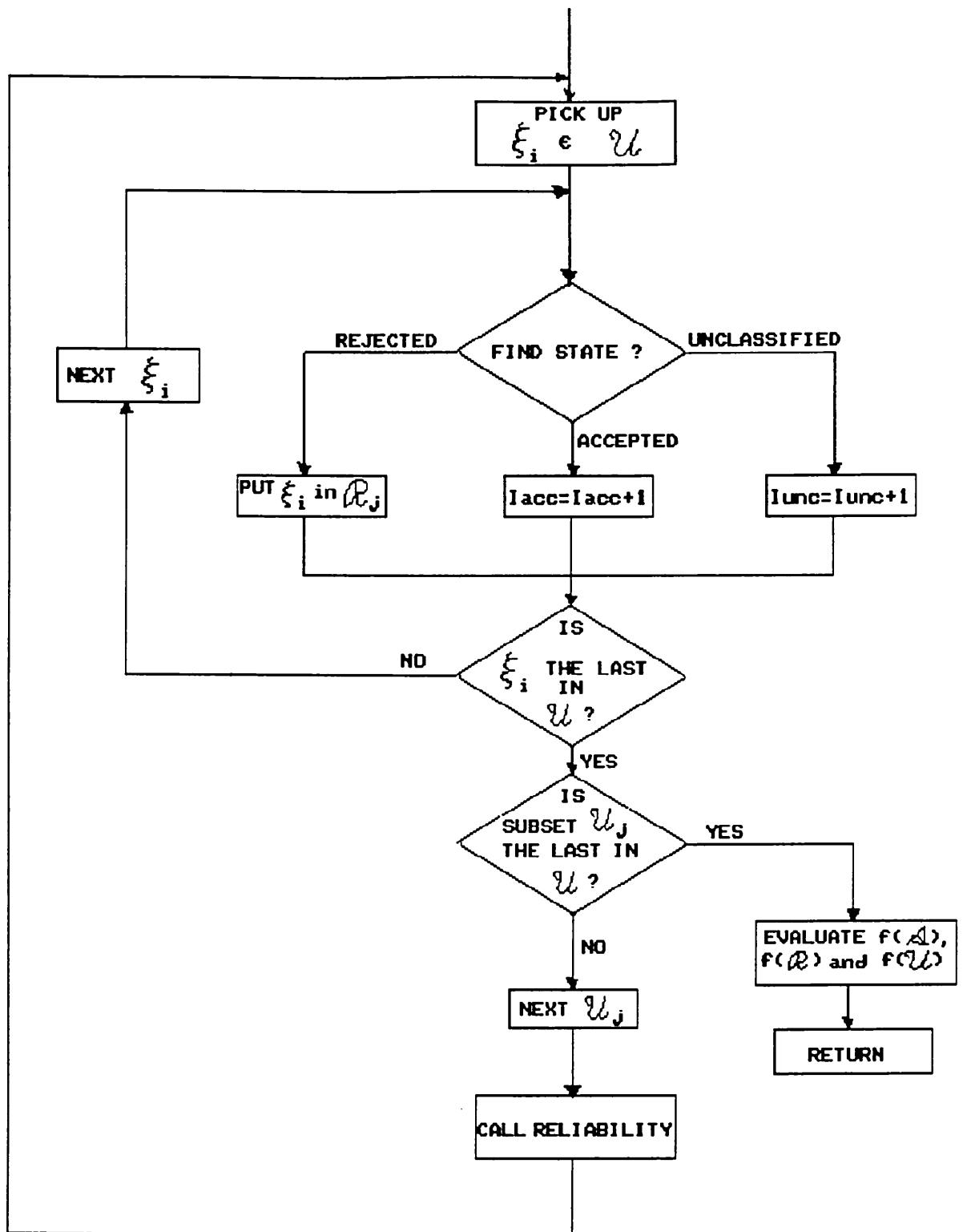


Fig 5.10 Flowchart of subroutine CALCPROB

5.4 RESULTS AND DISCUSSION

The algorithm developed has been applied to a 9-bus system (WSCC9 shown in fig 5.11) whose data are given in tables 5.1a, 5.1b and 5.1c. The upper limiting state for the 9-bus system is

$$\bar{\xi} = (1,1,1,1,1,1,1,1,1,1,1,1,1)$$

and the lower limiting state is

$$\underline{\xi} = (0,0,0,0,0,0,0,0,0,0,0,0,0)$$

Then, the NF algorithm produced the feasible flow pattern listed in table 5.2.

Consequently, the upper critical state is found to be

$$\xi^0 = (1,1,1,1,1,0,1,1,0,1,1,1,1)$$

This implies that any system state ξ_i that involves at least all the elements which have $v_m^0 = 1$, is an acceptable state because the load can still be met. Any system state ξ_i , in which at least one of these elements (which have $v_m^0 = 1$) is out of service, is yet unclassified. Hence, to define ξ_0 , each element m is removed in turn and the NF algorithm is run to find a feasible flow; at the end of the multi NF runs ξ_0 was found as

$$\xi_0 = (1,1,0,0,0,0,0,0,0,0,1,1,0)$$

Once ξ_0 is found, the boundary of the unclassified subset are known (Fig 5.6) and any system state ξ_i in which at least one element of those who have $v_{m0} = 1$ is out of service, is an unacceptable state because it cannot support the demand profile.

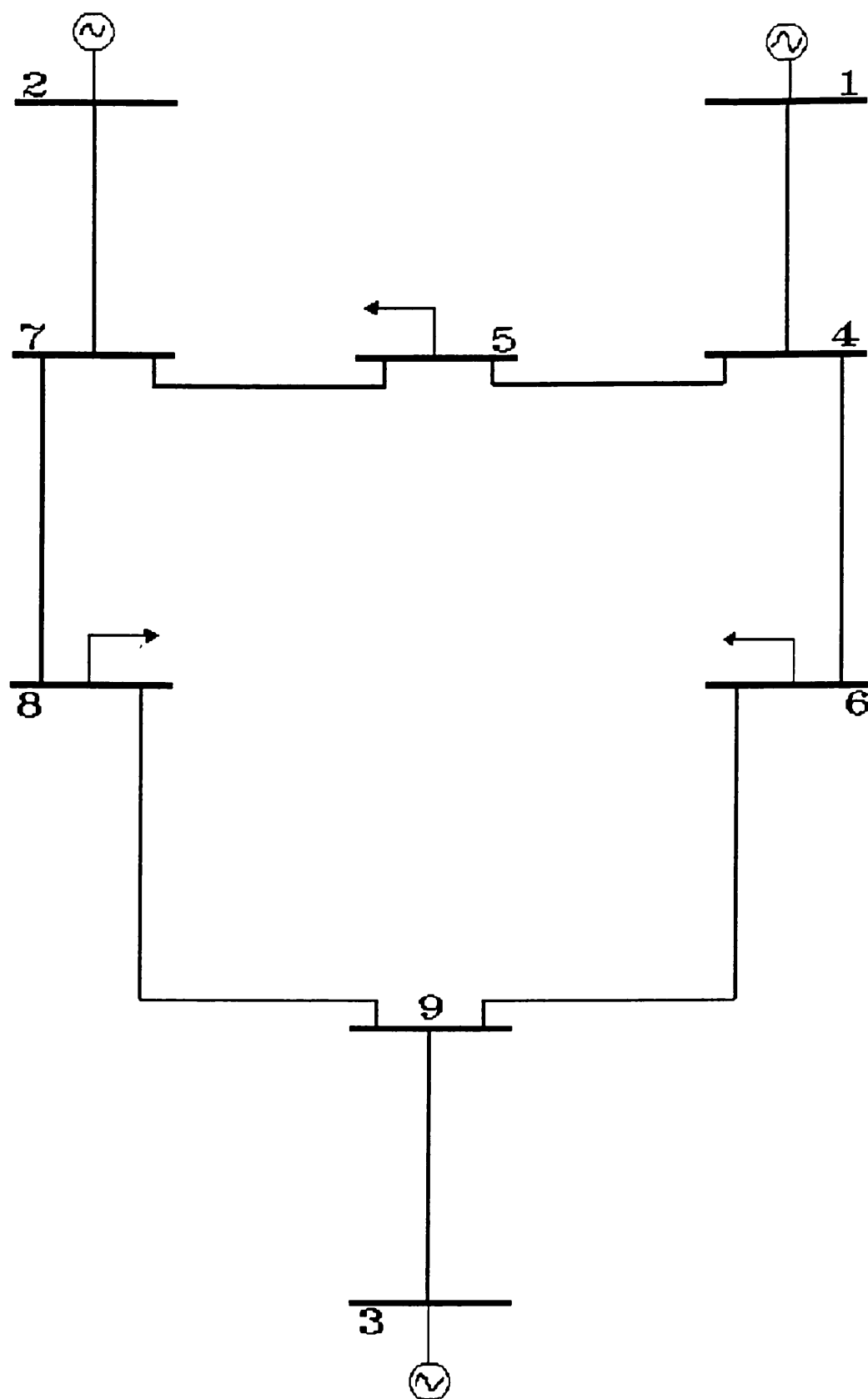


Fig 5.11 The WSCC9 test system

From	To	L	U	p	q
1	4	0	200	0.85	0.15
2	7	0	200	0.9	0.1
3	9	0	200	0.65	0.35
4	6	0	200	0.8	0.2
4	5	0	200	0.7	0.3
5	7	0	200	0.78	0.22
6	9	0	200	0.82	0.18
7	8	0	200	0.75	0.25
8	9	0	200	0.73	0.27

Table 5.1a Transmission lines data

From	To	L	U	p	q
10	1	0		0.95	0.05
10	2	100	163	0.98	0.02
10	3	85	85	0.97	0.03

Table 5.1b Bus generation data

From	To	L	U	p	q
5	11	125	125	0.99	0.01
6	11	90	90	0.99	0.01
8	11	100	100	0.99	0.01

Table 5.1c Demand data

From	To	L	U	LINE FLOW MW
1	4	0	200	130
2	7	0	200	100
3	9	0	200	85
4	6	0	200	5
4	5	0	200	125
5	7	0	200	0
6	9	0	200	-85
7	8	0	200	100
8	9	0	200	0

Table 5.2 The active power flow distribution

Now from eq 5.9 and eq 5.11, the probabilities of the accepted, rejected and unclassified states were respectively:

$$f(\mathcal{A}) = 0.1546533 \rightarrow 4 \text{ states}$$

$$f(\mathcal{R}) = 0.2877856 \rightarrow 3840 \text{ states}$$

therefore

$$f(\mathcal{U}) = 0.5575617 \rightarrow 252 \text{ states}$$

which is not significantly small in this case (the tolerance factor ϵ is taken as 0.05). Thus, subset U is to be exhausted. First, it is partitioned into non-overlapping subsets $\mathcal{U}_j$ (shown in table 5.3); then each in turn is exhausted i.e. the states $\xi_i \in \mathcal{U}_j$ are classified. The new probabilities become

$$f(\mathcal{A}) = 0.2578050 \rightarrow 44 \text{ states} \quad (+40 \text{ states})$$

$$f(\mathcal{R}) = 0.3351383 \rightarrow 3932 \text{ states} \quad (+92 \text{ states})$$

therefore

$$f(\mathcal{U}) = 0.4070568 \rightarrow 120 \text{ states} \quad (-132 \text{ states})$$

UNCLASSIFIED SUBSET	NUMBER OF STATES
$\mathcal{U}_3$	128
$\mathcal{U}_4$	64
$\mathcal{U}_5$	32
$\mathcal{U}_7$	16
$\mathcal{U}_8$	8
$\mathcal{U}_{12}$	4

Table 5.3 The Unclassified subsets $\mathcal{U}_j$'s.

Eventually, after further classification of the remaining 120 states, the probability of the accepted states is

$$f(\mathcal{A}) = 0.321046727 \rightarrow 52 \text{ states} \quad (+8 \text{ states})$$

and the final LOLP is

$$\text{LOLP} = f(\mathcal{X}) = 0.678953272 \rightarrow 4044 \text{ states} \quad (+112 \text{ states})$$

which is

$$\begin{aligned} \text{LOLP} &= f(\mathcal{X}) = \sum f(\mathcal{X}_j) \\ &= f(\mathcal{X}_1) + f(\mathcal{X}_2) + f(\mathcal{X}_{10}) + f(\mathcal{X}_{11}) \end{aligned}$$

clearly showing that for this particular system these elements (1,2,10, and 11) play the most important role in defining its reliability. As these results are dependent on the input data such as the element FOR's and the capacity limits, a change of transmission line capacity (especially line 1 as it connects the slack bus to the rest of the system) might result in alteration of the final LOLP. Thus to reduce the probability of the rejected states, in other

words to enhance the reliability of the system, either additional connection between the slack bus and the rest of the system should be installed; or the capacity of the existing connection need to be raised.

The single area reliability evaluation problem has been solved with a NF method. The algorithm developed is based on successive decomposition of the system states into subsets of acceptable, unacceptable and unclassified states [81]. Although the method has been applied to the evaluation of single area reliability analysis, the general concept is expandable to multi-area reliability assessment [75,78].

It is often necessary, in transmission planning studies, to evaluate the reliability of interconnected areas in a given region rather than the reliability of an isolated area. Also, it is helpful for defining reserve margins, to identify in which areas additional units should be installed and which interconnections need reinforcements. The system under consideration is thus represented by a number of interconnected load-generation areas (N-areas), where the indices to be calculated are the global LOLP and the areas's LOLP [78]. Each interconnection is assumed to be made up of a number of parallel circuits each of which has a transfer capacity and an outage rate.

Although using the NF methods, for reliability evaluation, proved promising [34] in terms of simplicity and computational effort, Wang and others [82] improved

their efficiency and developed a new method which greatly reduced the number of system states to be studied and contributed toward a substantial speed improvement (the CPU time taken by the new algorithm was less than $\frac{1}{241}$ to $\frac{1}{111}$ of that used by a previously developed NF algorithm [83]). In the author's opinion this could be exaggerated, nevertheless, it is worth investigating; the main disadvantage of NF methods, however, is that the feasible flow they give might not be realistic due to their inability to model Kirchoff's Voltage Law (KVL). It would be interesting to see the outcome of implementing Wang et al method [82] with the NF algorithm that satisfies Kirchoff's Voltage Law (section 4.3.4). Finally, the CPU time taken by the reliability evaluation algorithm was 2.61 sec on MicroVax II (the author has no access to a reliability assessment algorithm to compare the results obtained).

5.5 CONCLUSION

A NF algorithm has been developed to assess the reliability of a transmission system in a single area. As graph theory algorithms, they appear to have many advantages over other methods when applied to both single area and multi-area reliability assessment, where the area interchange is constrained by the transfer limits. Because reliability evaluation techniques require an exhausting amount of computational effort and data handling, the advantages of the NF methods lie, mainly, in the speed of

their solutions. Nevertheless, there still exist much work to be done in this area to improve their performance in terms of accuracy, perhaps through the consideration of the voltage effect on line flows.

CHAPTER VI

OTHER POTENTIAL NF APPLICATIONS

6.1 INTRODUCTION

In previous chapters, the theory of the network flow methods and some power system applications have been presented and discussed in detail. Then using the OKA, typical problems were solved and the use of the N.F method was justified by its satisfactory performance compared to some existing methods.

It was shown that N.F methods can be made competitive for certain power system planning and operational planning problems. The major advantages of these methods are the speed of solution and the simplicity of the formulation. This explains the growing interest in the use of N.F and transportation algorithms.

In this chapter, it is intended to highlight other potential uses of the N.F methods, reported in the power system literature, such as the scheduling of hydro power, emergency rescheduling, power interchange and group transfer methods. In all cases, the problems are linear or assumed to be linear. Each of these applications will be addressed and discussed in brief in order to exploit the features and the merits of using the N.F techniques.

6.2 OPERATIONAL PLANNING OF HYDROTHERMAL POWER SYSTEM

The optimal operational planning of a hydrothermal system involves different activities on a timescale of one day up to several years.

Many papers [84-90] have reported recently on the application of network flow models to the operational planning problem of hydrothermal power system. The majority of these papers addressed the short term problem for a large hydrothermal system (the Swedish system). In [84] seasonal operational planning was attempted.

These network flow models, in combination with linear programming techniques [84-86], have gained wide acceptance as an efficient tool for use in optimal hydro generation scheduling. Also a non-linear hydro plant model was developed [87,88] which permits variable reservoir heads. The operational planning problem was described in terms of a non-linear network flow model.

Among these papers, [86] the use of Bender's decomposition method [91], is of particular interest as it decomposes the short-term operational planning problem into master problem and subproblem. The subproblem was formulated in a network structure which makes it possible to use N.F algorithms and provide solutions about 100 times faster than standard methods [5]. A brief summary of the work is presented in the following section.

6.3 SHORT TERM OPERATIONAL PLANNING OF LARGE HYDROTHERMAL SYSTEM

The problem is a large scale mixed integer program. Bender's method [91] is employed to decompose the problem into a master problem, that contains only integer variables and considers the unit commitment of thermal plants; and a subproblem that includes continuous variables and considers the economic dispatch problem.

The objective function is the production cost of thermal plants over the operation horizon. Constraints are divided into thermal, hydro and system constraints. A schematic illustration of Bender's decomposition is shown in fig 6.1. The master problem is an integer program and is solved for the unit commitment schedule of thermal plants which are then fed to the subproblem. The subproblem is a continuous program with fixed thermal commitment schedules. The dual values of the subproblem are fed back to the master problem and introduce a new constraint (Bender's cut) to this problem. This process is continued until the upper and lower bounds available in this method converge.

Bender's method proved to have a slightly higher efficiency than the Lagrangian relaxation technique.

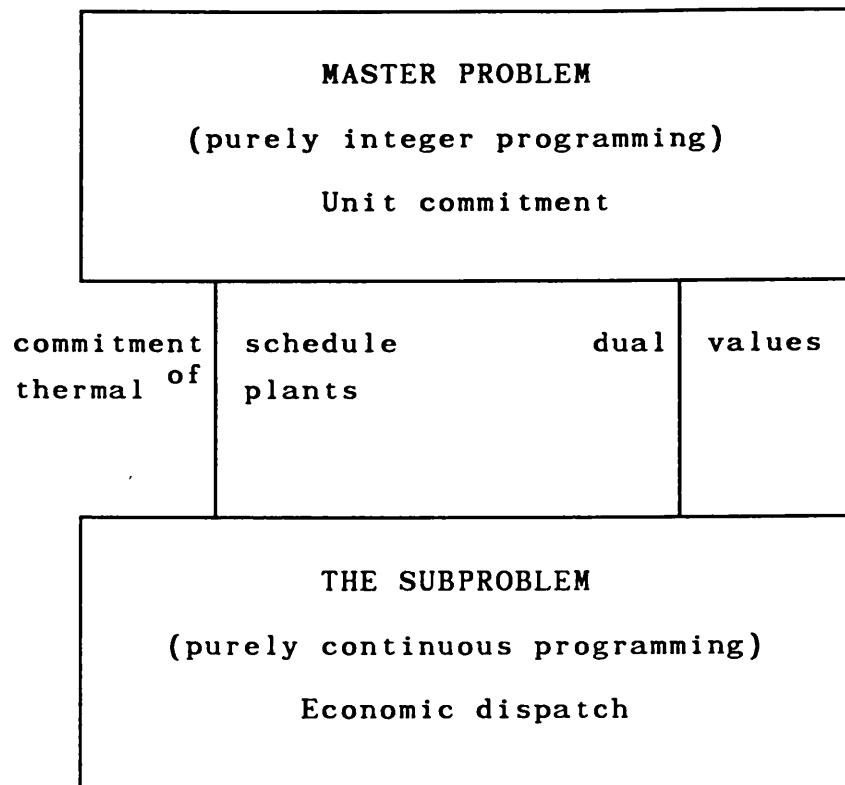


Fig 6.1 Benders' method for short term operational planning.

6.4 GROUP TRANSFER METHOD

To ensure a secure operation of a system the total generation or its distribution must be chosen correctly in relation to the expected demand and the network facilities.

It might occur, however, that current flows exceed circuit capacities causing line outages and consequently disturbing the demand. One of the basic methods that can be used to analyze the effect of excessive current flow is the group transfer method.

A group transfer is the power flow into or out of the identified substations. The constraint imposed on this group limits the power flow from/to the selected group usually after allowing contingencies.

By limiting the transfer of power into or out of the group the insecurity under consideration can be removed. In other words, the difference between the group demand and the group generation is not permitted to exceed a fixed value, I_g for import, E_g for export. Expressing this rule algebraically :

$$\sum D_i - I_g \leq \sum P_g \leq \sum D_i + E_g$$

leads directly to a network flow formulation. The group demand sum can be calculated for the load level of the case being studied. Then the restriction on the generated

power can be modeled by a single arc to enhance transmission security between groups.

The group transfer analysis has been incorporated into the economic dispatch problem and used for on-line application on the CEGB system [92]. The major disadvantage of the method lies in the task of identifying the critical groups, to minimize the computational effort, where there always exists a risk of missing one/some of these groups. From operational experience, or following an off-line security check, a group of generators and lines can be identified as insecure under some conditions (it is difficult to identify potentially critical groups of substations [53]).

6.5 EMERGENCY RESCHEDULING

The problem is to relieve network overloads by active power control. Stott and Marinho [29] state the means to correct this overload as:

generation shifting

phase-shifter control

HVDC link control

emergency start-up

load shedding

All these control actions can be represented analytically as bus injection changes, i.e. as equivalent

generation changes. This enables a network model to be built so that the network flow method is applicable.

In 1986, Hon et al [93] formulated the problem of active power emergency rescheduling as a capacitated transportation problem. Then, the OKA was applied for the alleviation of network overloads or underloads by correcting active power generation or, if necessary, imposing load shedding. The load shedding was simulated by a fictitious generator connected to the load bus. The amount of active power derived as a result of the OKA calculation is equivalent to the amount of load shedding required.

The OKA minimizes the cost of generation increment and the cost of transmission increment subject to the constraints imposed on the generators and the network lines (i.e. to satisfy the flow conservation constraint and to recognize the upper and the lower limits).

The algorithm developed proved to be rapid, (faster than the algorithm proposed in [94]), robust in the elimination of overloads and/or underloads and suitable for on-line uses.

6.6 POWER INTERCHANGE SCHEDULING

In 1982, Doty and McEntire proposed an algorithm [95] based on a network flow method (the minimum cost flow algorithm) to analyze a power brokerage system. The

objective is to reduce the cost of generating electricity by making short term transfers of electric power between utilities.

The problem is formulated in a network structure where the aim is to find the shortest path tree. Then the minimum cost flow algorithm is used to solve the shortest path problem where the costs are taken as lengths. The implementation of the algorithm is a simple and straight forward task. Using the N.F model proved to be satisfactory for the objective of cost minimization and involved the less of mathematical modeling.

6.7 SECURITY CONSTRAINED ECONOMIC DISPATCH

On-line economic dispatch solutions should take into consideration the security constraints dictated by the loss of a transmission line (generator or load). Thus, the objective function of the economic dispatch problem becomes to minimize the cost of power generation subject to meeting the demand when providing a specified amount of spare generation and recognizing the security limits. This is known as the security constrained optimal power flow [96].

Stone [13] proposed a method, using the Out-Of-Kilter algorithm, that yield an optimal secure solution. First, an economic schedule was found with line thermal limits operative; then a series of load flows were run each with

a line out of service. After every outage and load flow, the overload (if applicable) is recorded and when the multiple load flows are complete, new security limits are calculated: half the maximum overload on each line is subtracted from the old thermal limits to give new limits. This was repeated until:

1-No overload was met i.e. the solution is a secure economic schedule of generation

2-Or the security limits cannot be met with the specified demand profile; in other words, a load shedding is to be performed.

The algorithm developed was applied to the 23-bus system of the CEGB network (fig 3.4). The demand pattern of 2643 MW was not met as the transformers 27 and 29 were not able to transport power securely to the low voltage network from the generators at bus 23. However, a total demand of 2300 MW was optimally and securely satisfied, yet on average this was more costly than the optimal dispatch solution that neglected the security constraints. Shen & Laughton [97] compared Stone's results [13] with those obtained using dual linear programming. The line flow distribution of the NF algorithm was slightly different from those obtained by the dual linear programming. This could be due to the fact that the NF algorithm neglects all voltage consideration. Although further development to improve the accuracy of the results is needed, it is believed that the prospect of using the NF

algorithm for the security constrained economic dispatch is promising; for security purposes where the time is decisive factor, if the accuracy is satisfactory, the emphasis should be put on the speed of the solution, rather than the exact values of line flows.

6.8 OPTIMAL CORRECTIVE SWITCHING

Power systems have to operate in a secure state to ensure reliability of electrical power supply and in order to satisfy the demand. This calls for security analysis to study possible preventive or corrective actions capable of enhancing the system security.

In previous sections, some of these preventive and corrective measures have been presented (emergency rescheduling, group transfer and security constrained economic dispatch). One other method to enhance system security is to relief overloads by switching lines in or out of service. The NF methods has been applied to the problem of optimal network switching.

Rossier & Germond [98] developed a method based on the Maximal Flow Minimal Cost algorithm, for relieving overloads by changing the network configuration. The optimal solution meets the various constraints and the resulted power flow distribution is nearly similar to the one obtained by a DC load flow. However, as only active power can be considered, an AC load flow is used to check voltage and short circuit current constraints (by a fast

3-phase short circuit program). The algorithm proceeds iteratively until a solution that respects the security constraints is found. When applied to the IEEE 24-bus system [35], it was found that the overall solution is very fast (≈ 10 s on a Vax 11/780) which makes it applicable for on-line uses.

6.9 TRANSMISSION SYSTEM EXPANSION

Planning the expansion of an existing bulk power system is a complex problem. The objective is to identify the optimal expansion plan subject to the constraints of reliability and security.

This problem has been formulated into a network flow structure by Padiyar & Shanbag [99] who then used the OKA to define the minimum cost solution. The methodology of the problem is as follows: the rights of way and the load demands at specified buses are defined and the objective is to minimize the total investment and operational costs. This formulation allows the inclusion of generation costs, in the objective function, so as to permit an optimal location of generation to be defined. The results obtained agree with those reached using other methods, based on DC and AC load flows, and the OKA solution proved to be superior in terms of simplicity and computation time [99].

As a result, the OKA is believed to be advantageous algorithm for such applications even if the formulation

obeys only Kirchoff's current law, as the detailed accuracy of the results at the planning stage is not a decisive factor in the final solution. For bulk power systems the expansion options could be enormous and impose a heavy computational burden on the system planner; a NF solution is fast, simple and accurate enough for such applications.

6.10 CONCLUSION

In this chapter, it was intended to cast the light upon some potential applications of the N.F methods and to prove that their uses for such power system problems is promising in itself.

The classification of the addressed applications varies between economic oriented and security oriented purposes. This could well be due to the natural analogy between the former and the network flow problems, where a flow of a commodity is transported along a network of nodes and branches so as the total capital cost is kept to a minimum. While the latter (security oriented uses) makes the most of the fast solution, that NF algorithms yield, when retaining a certain desired accuracy. Those two economic and security oriented purposes have been successfully cast into network structure which made a NF solution possible. The merits and the features of the proposed formulation will be addressed in the next chapter.

CHAPTER VII

CONCLUSION

7.1 THESIS CONTRIBUTION

This research has explored the on-line potential applications of NF methods, assessed the practicality of their solutions for real time use and has highlighted possible improvements and suggestions for future consideration. The interest in studying NF methods was stimulated by the surprising range of power system problems that can be cast into a network structure of nodes and branches.

The Out-Of-Kilter Algorithm (OKA) has been applied to the problem of active power economic scheduling where the total generation/transmission cost is minimized, subject to the capacity limits. As demonstrated in Chapter 3, the OKA solution satisfies Kirchoff's Voltage Law when losses are minimized and line flows are not constrained. Alternatively, KVL can be accounted for through the loop equations and consequently the accuracy of the results is maintained. The speed of the algorithm has been improved by checking only the status of the branches connected to load points when initially the transmission lines and the branches representing the generating buses are in-kilter (state α). Another major improvement, made to the OKA code, was the ability to model the flow in both directions using one single arc, thus reducing the total number of routes for the labelling technique without affecting the

path search procedure.

The NF ability to satisfy KVL through the loop equations has enabled the use of the OKA to solve for AC/DC load flows. An active/reactive decoupled solution, using the modified OKA as demonstrated in chapter 4, has been found faster than the FDLF for systems having fewer number of loops than the number of nodes. Yet, it appears that the real advantages lie in applications which require several calls to the load flow solution such as security assessment. For this purpose, a simple contingency analysis study has been performed on the IEEE 24-bus and load shedding caused by line outages has been determined. This was restricted to the active power as load shedding required to deal with reactive power problems was considered beyond the scope of this investigation.

Another further objective was the reliability evaluation of a generation/transmission system. A probabilistic method was implemented to evaluate the LOLP for a small test system (9-bus). Although time constraints did not permit the completion of the planned investigation, the author believes that the aim is partially fulfilled by the development of the reliability assessment algorithm which showed that applying the OKA to reliability studies is very promising.

The Out-Of-kilter Algorithm has been used to solve many operational research problems. However, the only power system applications that have been reported in the

literature are Emergency Rescheduling [93] and Transmission System Expansion [99]. Consequently, this work is unique in demonstrating the use of the OKA to solve the three problems summarized above and in the in depth study of possible power system applications that could benefit from the OKA features such as speed of computation, ability to obtain optimal results, simplicity, etc. Finally, by listing and critically commenting on the power system literature regarding NF applications, this thesis constitutes a survey and bibliography for NF methods.

The following sections address possible improvements to the solutions reached and provide some guidelines for future use. The hope is that more work along the lines suggested will be stimulated.

7.2 IMPROVEMENT IN LABELLING SEQUENCE

Bearing in mind the mechanism of the OKA, it is clear that the most time consuming process in the algorithm is the labelling procedure to determine the augmentation path. Thus it is obvious that any attempt to improve the speed of the algorithm should address the path search technique.

Consider for example the IEEE 24-bus system (Fig 3.5); when trying to solve the DC load flow problem the arc connecting the source node (Nb 25) to generating bus 1 (which models the power generation at bus 1) is supposed to carry an amount of flow which corresponds to the

capacity of the generator. Otherwise, the arc (25,1) is out-of-kilter and the labelling procedure should try to find the path along which the flow can be changed so as to put (25,1) in-kilter. After the first call to the labelling routine a path has been found (Fig 7.1a). However, the flow change incurred is less than the amount required to put (25,1) in-kilter, although it has improved its state; hence another call to the labelling routine is made to define another path (Fig 7.1b). Now each time the labelling routine is executed the node labels are initialized. This means that for each iteration, part of the labelling process is repeated (as illustrated in fig 7.1b). Thus if the labelling is begun from the previously reached stage, then a great saving in computation time could result, especially for large and heavily meshed networks. Implementing these alterations (within the labelling process) is an easy task as long as the overall solution is still considering the same out-of-kilter arc; when the arc is left in-kilter and the algorithm switches to another out-of-kilter arc, the changes in the labelling routine are slightly different. Nevertheless, as shown in fig 7.2a and 7.2b, common labels exist between the two cases e.g. when considering arc (25,1) and arc (25,2): in the second case the labels for nodes 4 and 6 remain the same as reached in the first case (as in each case they are labeled from node Nb 2).

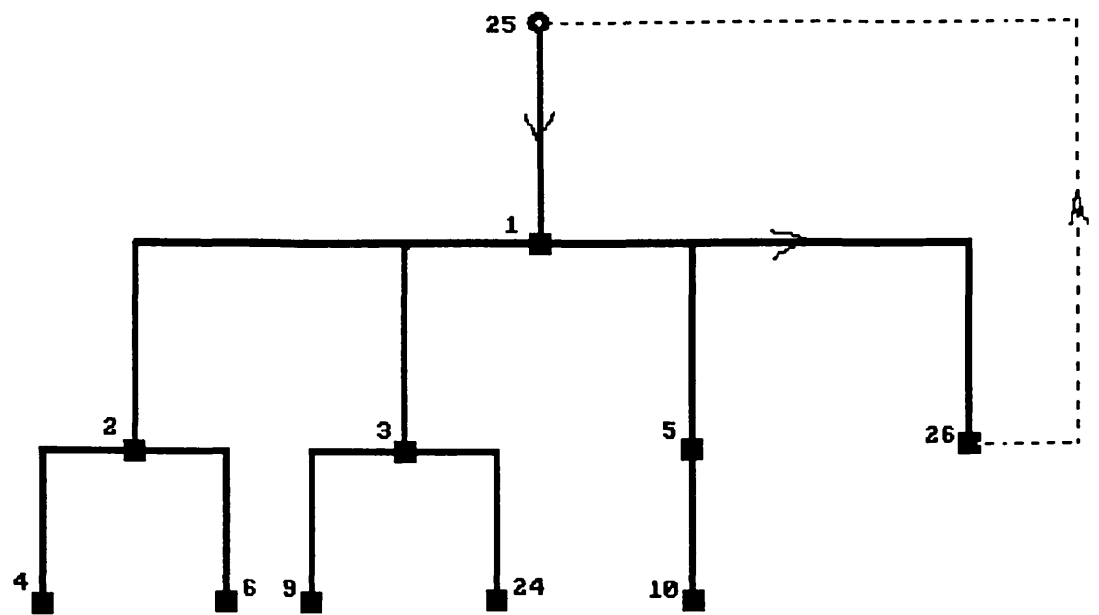


Fig 7.1a The flow augmentation path

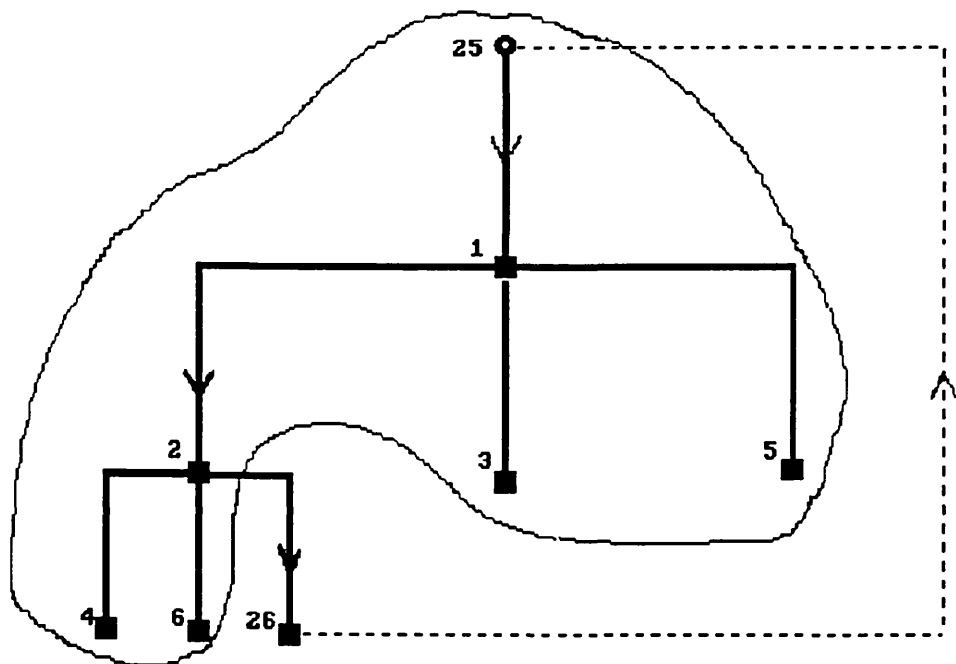


Fig 7.1b The path of the second call

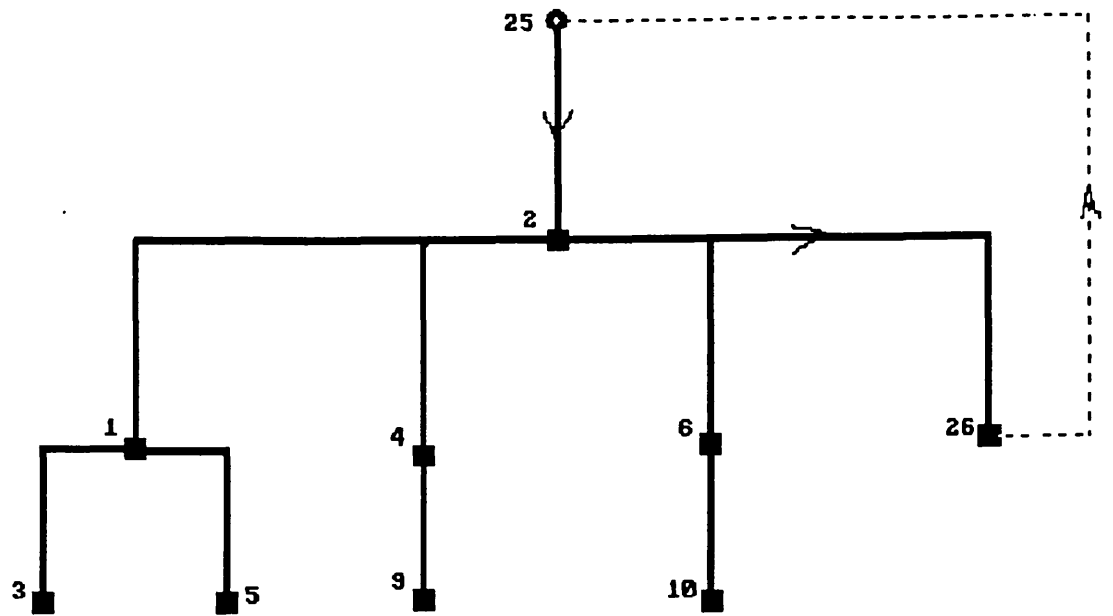


Fig 7.2a The path for different arc

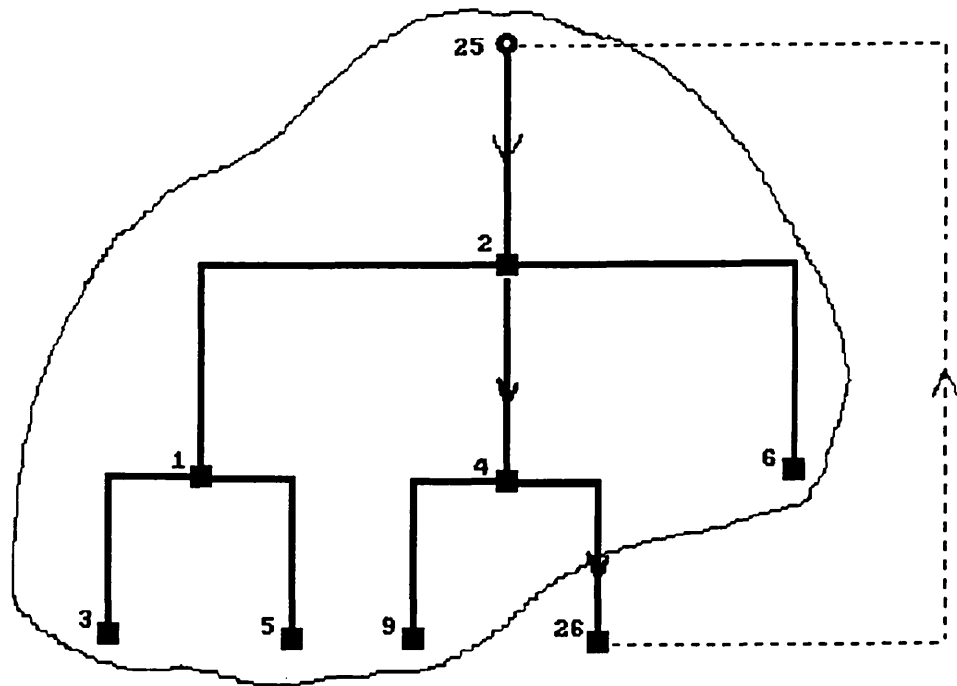


Fig 7.2b The path of the second call

7.3 GUIDELINES FOR FUTURE USES

Considering the power system applications of NF methods, their use is promising even if they obey only Kirchhoff's Current Law. However, much work still needs to be done, in the author's opinion, for the NF applications to be really useful.

For instance, when solving for active power economic dispatch the maximum power transfer from a selected group of substations is often constrained so as to enhance the security of the system.

As the OKA becomes capable of meeting voltage constraints, it is worth investigating the prospect of developing a full OPF algorithm where the total generation cost is minimized and the reactive power reserve is maximized. Should this prove promising, the study could be extended to define a voltage collapse proximity indicator (VCPI) based on [100] which requires a fast and accurate OPF (with multiple OPF solutions).

Other major NF applications, which are probably the most researched areas are:

1. Reliability evaluation requiring extensive computation and involving detailed graph theory (such as the definition of cutsets, trees, etc). The speed and the accuracy of NF solutions can be improved [82]. The author believes that the main hurdle to overcome is to meet the voltage constraints while retaining the competitive speed of the NF computation.

2. Fuel and hydro scheduling, which are by definition transportation problems, that need no undue effort to put into the network flow structure. Perhaps there is not much to be done in this area beside the enormous efforts made mainly by the Swedish authors [84-88].

Bearing in mind the size and nature of power system computations, it is sometimes advisable that large problems be decomposed into many subproblems (using Benders' decomposition for example [91]). This could enable the use of different algorithms, including NF methods, to solve one large problem where there exists an interface between the different parts of the solution. In other words, a large problem would be decomposed into NF-compatible and NF-incompatible parts and it would be ideal if this could be done automatically rather than manually [101]. Although this might be more difficult for the researcher as it means that he has to work on more than one algorithm plus interfaces, the prospects seem interesting because it retains the advantages of NF solutions. For instance, NF algorithms are competitive candidates for power system problems that involve topological analysis such as defining areas for security purposes, depending on certain criteria [102].

The determination of independent loops in Chapter 3, so far, was done manually and included in the input data. Although this task may be somewhat laborious, it has to be done only once unless a change in the system configuration has occurred due to line switching. In this case, only

part of the tree affected by line switching has to be revised and there is no need to alter the whole loop matrix. At a previous phase of this work it was intended to implement an automatic procedure for independent loop determination as given by [103]; unfortunately, the author could not manage to obtain ref. 103 (although he has written to the researchers involved in its development). However, with some modification to the labelling procedure an automatic routine for determining the independent loops could be developed as the prime objective of the labelling process is to define the flow augmentation path, which is in fact a loop (that includes the arc under consideration).

Finally, the privatization of the electric power industry in Britain has stimulated an attempt at Imperial College to develop a method, using a NF algorithm, that defines a wheeling rate for the power transmitted from one utility to another through the facilities of a third party (called the wheeling utility). The problem basically is a transportation one, thus a NF solution should be possible and practical.

7.4 CONCLUSION

The Out-Of-Kilter Algorithm (OKA) has been applied to many power system problems. This research has proved that NF methods, in general, have potential in operational planning where the emphasis is on speed of solution rather than detailed accuracy.

NF methods are special purpose linear programming algorithms developed to solve problems of a transportation nature. Thus their use, unlike general purpose LPs, is restricted to applications that can be formulated into commodity transshipment problem between a network of nodes by branches. For such applications, specialized NF algorithms can reach a solution in execution times much faster than the most advanced general purpose LP; the convergence is always guaranteed, if a solution exists. In addition to their simplicity and the ability to visualize the problem, this explains why researchers have approached such applications using NF methods.

Further work is required on transportation and graph theory methods so that they can be used for real-time uses on practical networks. It is recognized that the larger the considered system, the more significant are the gains to be made by using NF methods.

APPENDIX A

THE COMPUTER IMPLEMENTATION OF THE OKA

A1 Introduction

The out-of-kilter algorithm of Fulkerson [6] has been written in standard Fortran as a group of interrelated subroutines; each of these subroutines is dedicated to tackle part of the overall problem. The algorithm basically consists of two main stages:

1. The labelling technique is used to define a flow augmentation path, along the network, from the origin node to the terminal node. This path search should be done, taking into account the flow across the lines and the possibilities of a flow increase (or decrease), in order to fulfill the aim of putting arcs in-kilter.
2. Once an augmentation path has been determined, the flow along that path is changed by the appropriate permissible amount without sending any arc out-of-kilter or further out-of-kilter i.e. without violating the limits on branches.

To provide a better understanding of the OKA, the computer implementation is presented. This appendix discusses the subroutines involved, each at a time in order of execution. Its flowcharts are also shown. The source prog-

ram listing is given in appendix B where some comments statements are included. The main computer program is taken from [13].

A2 Subroutine KILTAL

This is the main subroutine of the OKA. It controls the call for the subroutines involved. The routine starts by inialising the total number of breakthrough (TBT) and non breakthrough (TNBT). Then, Do-Loop 100 picks up the arcs of the network, in turn, where the arc state is inspected in FINDST. If the arc being considered is in-kilter, FINDST set STAIND equal to zero. The next arc is then inspected; otherwise KILTAL calls subroutine LABELR in an attempt to define a flow augmentation path.

If NEXT equal zero, LABLER failed to find the path and a non breakthrough has been detected. Thus, NODPRI will change the potentials of unlabeled nodes (node price) to improve the state of the considered arc in the next iteration; always after updating the CBAR values ($\bar{C}_{ij}$) in subroutine CPIPI. When no changes in node prices is possible the algorithm shows the message that no feasible solution to this problem exists.

However, if NEXT is not equal zero, the path has been successfully defined and KILTAL transfers the control to FLOCHA which will change the flow along the arcs of the path accordingly and return the control to FINDST, so that

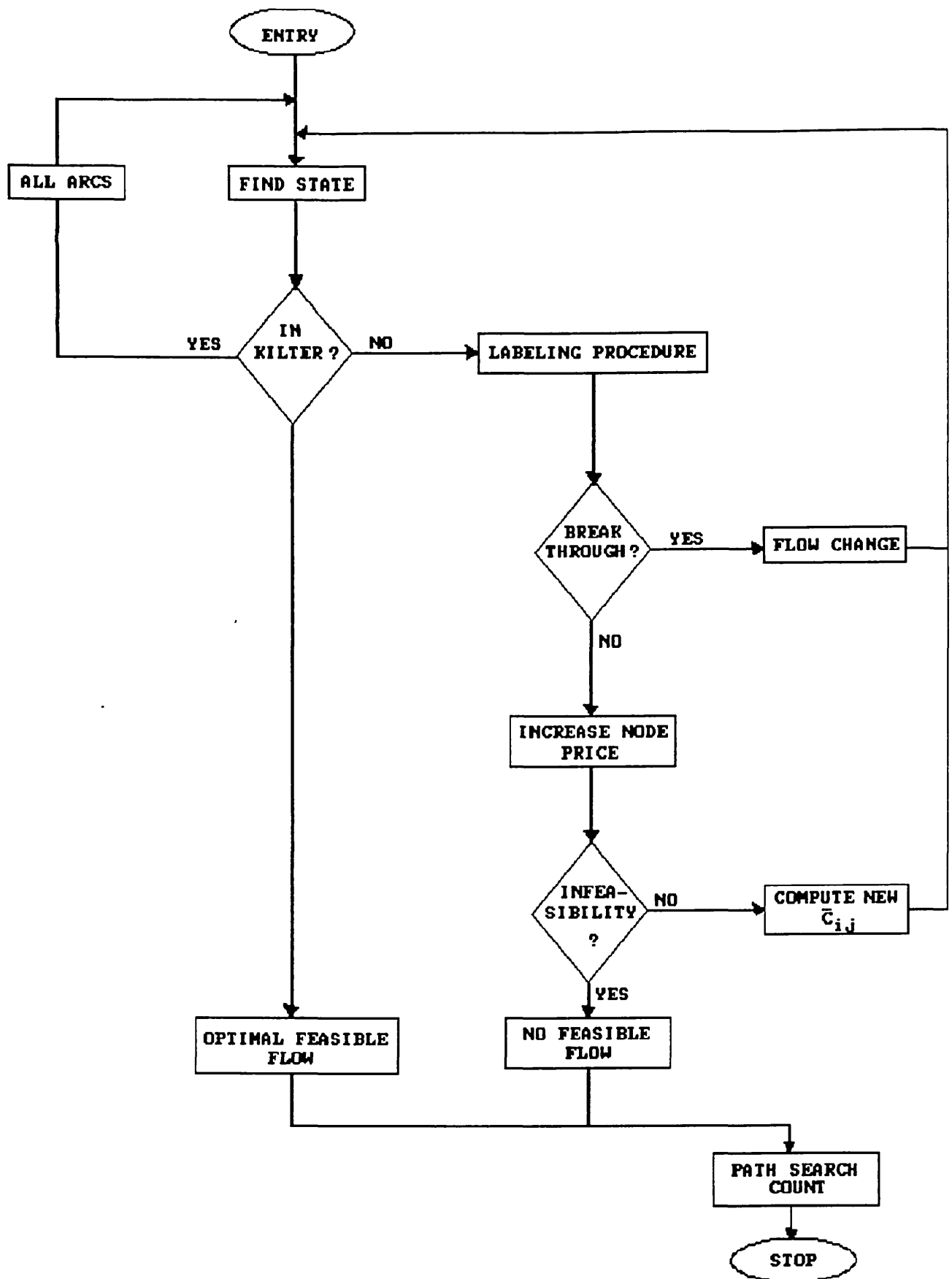


Fig A.1 KILTAL flowchart

the new state of the arc can be inspected. KILTAL finishes when either Do-Loop 100 is satisfied i.e. all arcs are left in-kilter and a feasible optimal solution is reached; or at least one arc is still out-of-kilter and the algorithm cannot put it in-kilter. For the user observation, the algorithm provides the total number of breakthrough, TBT, and non-breakthrough, TNBT. The flowchart for KILTAL is given in fig A.1.

A3 Subroutine NODARC

The idea behind "NODARC" is to tabulate the network connections in a complete and convenient way that permits high speed access to the nodes of a specific arc. This is achieved by tabulating the start node $S(k)$ and the terminal node $T(k)$ of each arc k to form the arc to node table. As all subroutines use this information, it is made accessible through the common blocks.

Also the node-to-arc table is needed for the execution of the labelling procedure; when performing the labelling process manually on a piece of a paper, the arcs connected to a given node are obvious while this is not the case when this task has to be done automatically. Thus a node to signed arc table is constructed in which all arcs connected to a particular node are collected together.

The signed arc array, KS, and the node location array, NODLOC, hold the node-to-signed-arc table. The

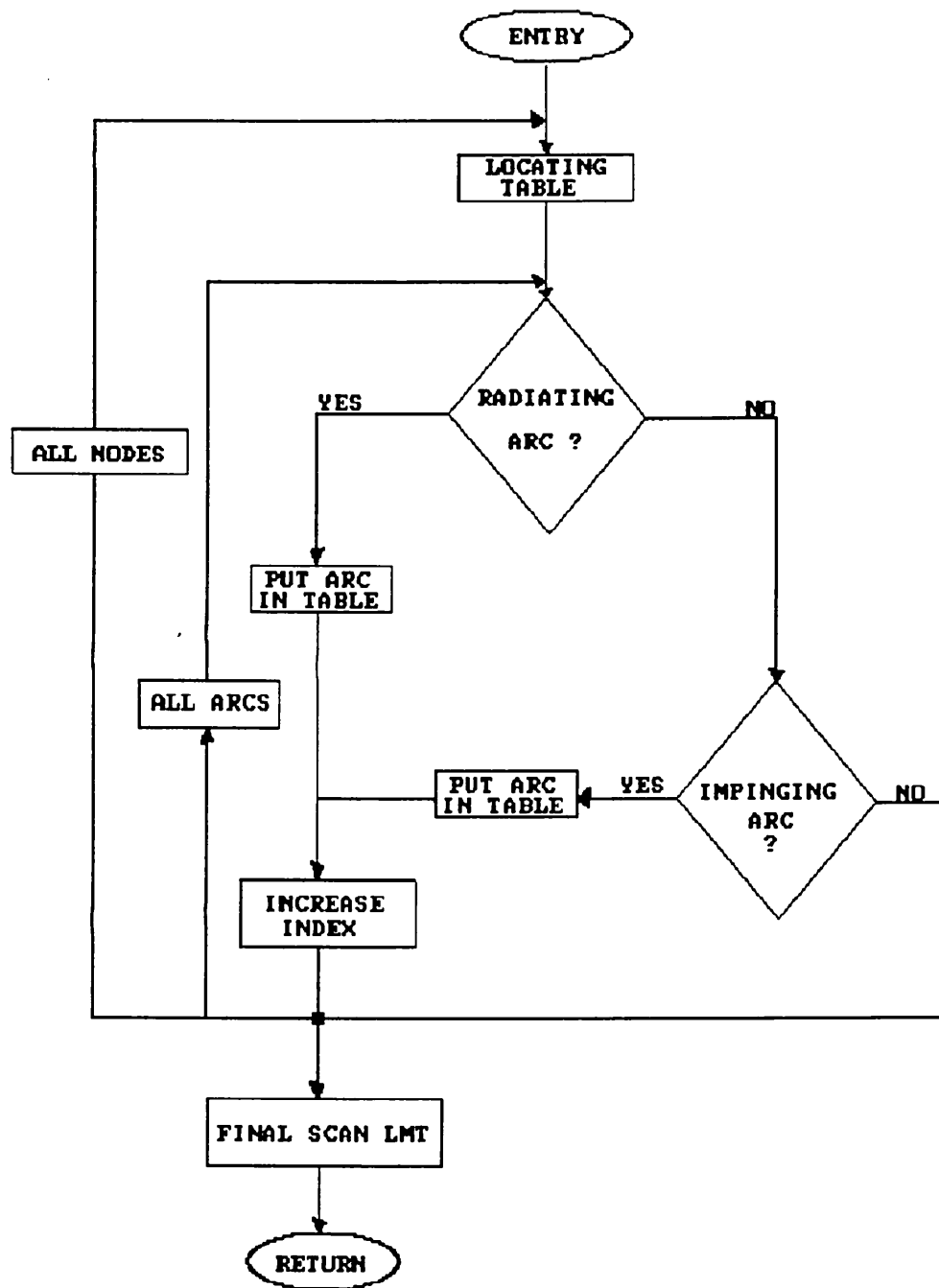


Fig A.2 NODARC flowchart

Do-Loop 200 scans the arcs of the network and builds up array KS which contains the set of arcs KM connected to a node IM. Arcs radiating from IM are given positive sign +KM while those impinging are negatively signed -KM. This is continued until Do-Loop 100 is satisfied which means that the node to arc table has been constructed.

LSA is a count of the total number of arcs added to the table in array KS. NODLOC defines the number of arcs connected to a node IM. The flow chart of "NODARC" is shown in fig A.2. Finally, this subroutine is called for only once at the start of the program execution and needs not to be recalled again, even after lines trip out.

A4 Subroutine CPIPI

This subroutine is called first at the start of the OKA execution to compute the optimising costs $\bar{C}_{ij}$. It consists of a single Do-Loop which ensures that $\bar{C}_{ij}$ values are calculated for all arcs of the network. The variables SKJ and TKJ are node indices for the start node and the terminal node (of a given arc) respectively. CPIPI is called for again following a node price increase that changes the nodes potential values $PI(SKJ)$ and $PI(TKJ)$ i.e. changes in π 's values in subroutine "NODPRI". The flowchart of CPIPI is illustrated in fig A.3.

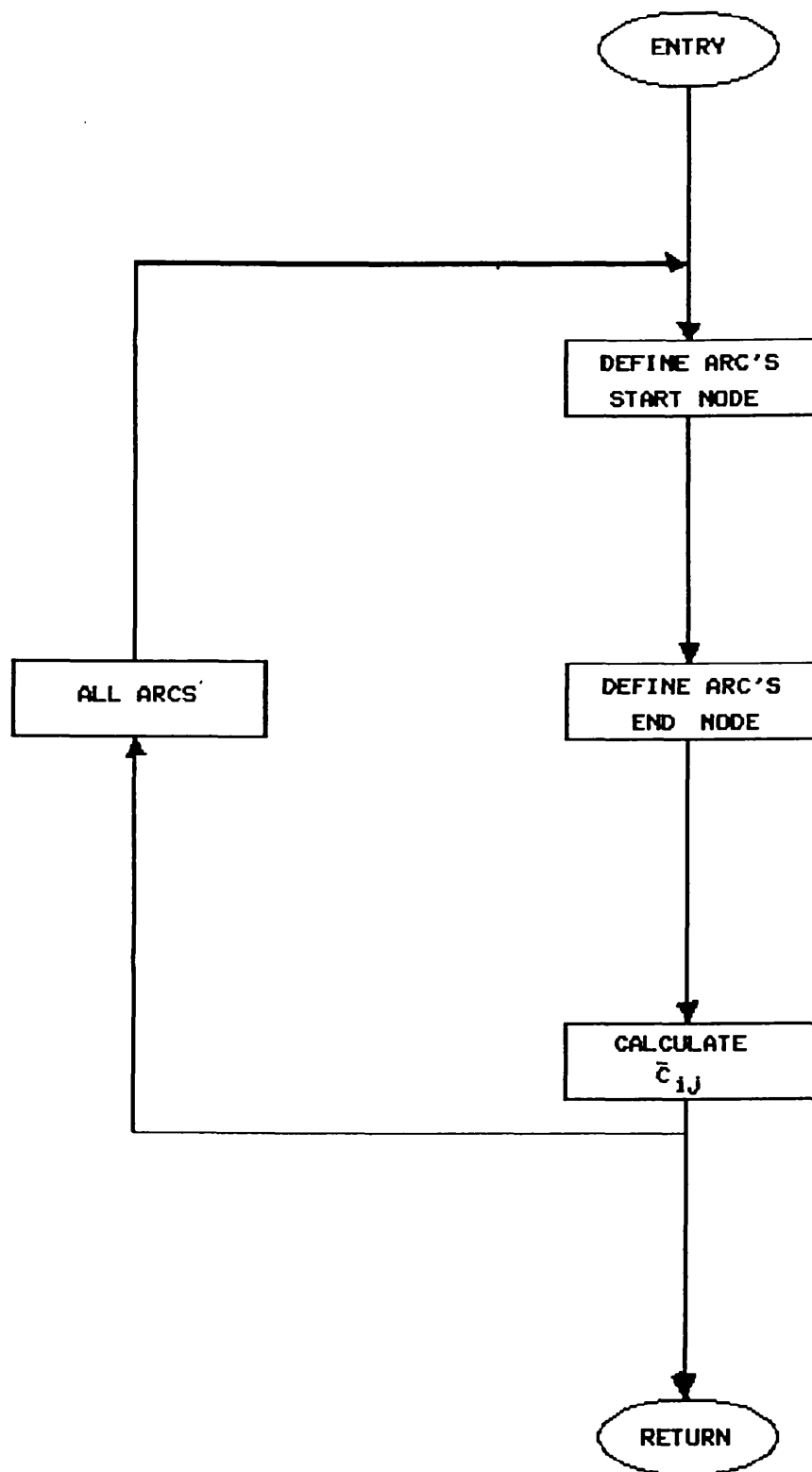


Fig A.3 CPIPI flowchart

A5 Subroutine FINDST

In this routine the state of the arc under inspection is determined. In other words, FINDST checks the state of the arc, using nodes and branches data, against the nine mutually exclusive states listed in table 2.1. The routine starts by computing the arc differences:

$$f_k - L_k \text{ and } U_k - f_k$$

while C_k values are available from CPIPI. Thus FINDST is ready to find the state of arc k .

The five arithmetic IF statements are the logic of the routine; they detect whether arc k is in state ALPHA group, BETTA group or GAMMA group. Then the origin for labelling and the terminal for labelling are determined. FINDST also, defines the permissible flow change, DIF, depending on the arc state.

The alphameric arc state descriptors is assigned to variable "STADES" for possible observation by the user. The removal of STADES, thus, would not affect the routine execution. FINDST flow chart is presented in fig A.4.

A6 Subroutine LABELR

This is the node labelling procedure where the search for a flow augmentation path is carried out. The path across the network should start by the origin for labelling node (OFL) and finish at the terminal for labe-

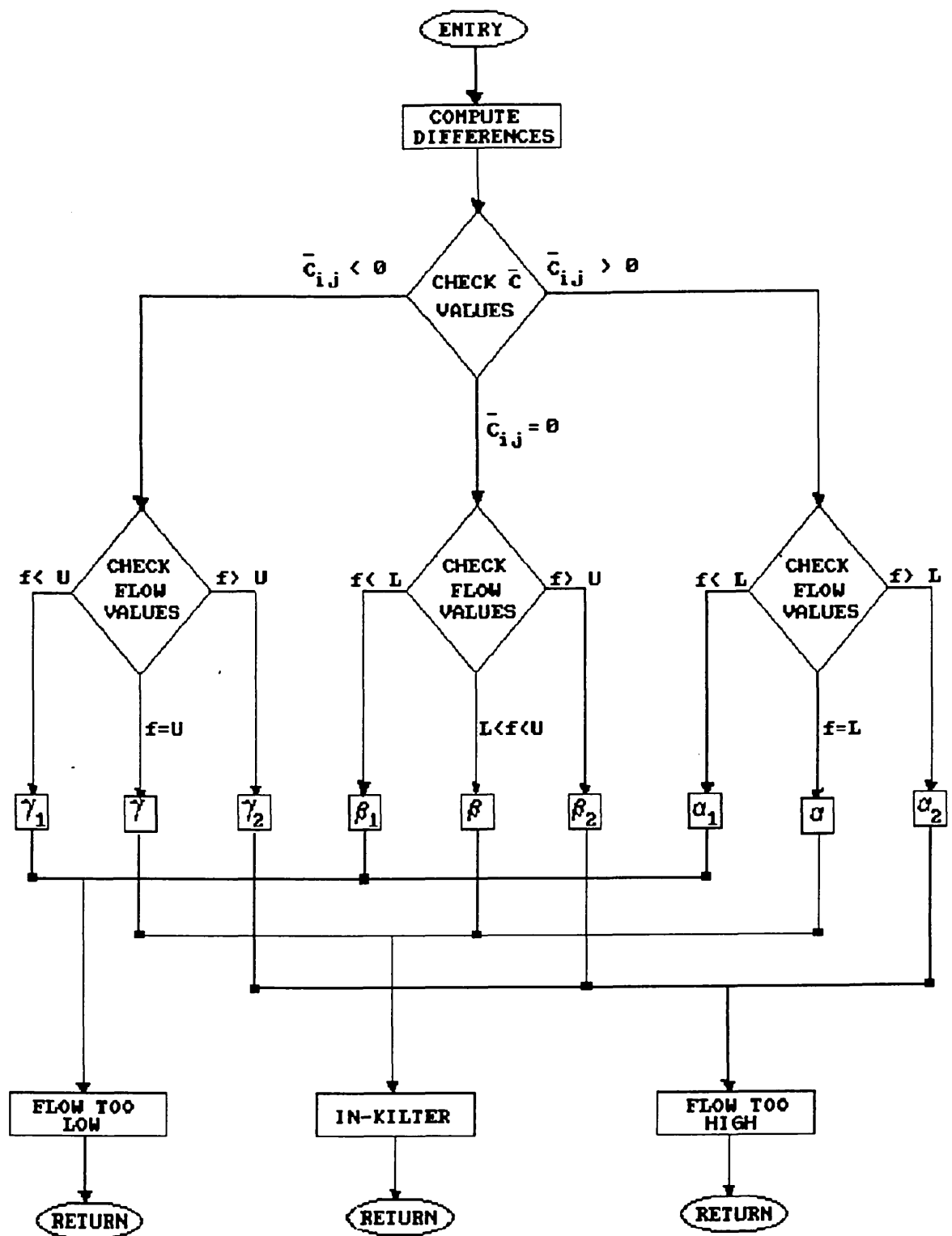


Fig A.4 FINDST flowchart

lling node (TFL); both nodes have been defined in FINDST.

The mechanism of the labelling technique consists of assigning two labels for each node. The first, KLABEL(IL), hold the arc number over which the node, under consideration, was reached. The second, ELABEL(IL), is the maximum permissible flow change up to the node (IL). In doing so, two lists of labelling nodes are maintained; the present stage nodes, from which labelling is being done, and the next stage nodes, which are nodes that have been labeled from the present stage. This information is stored in array NPSTAG.

The routine starts by assigning large numbers to the origin for labelling node (OFL). Then, the outer Do-Loop 100 computes the number of arcs connected to node IL (NARCIL). The inner Do-Loop 100 scans the arcs one at a time. The IF statement determines whether arc KL radiates from or impinges on node IL (in other words, forward or reverse arc), transferring control to statement 2 in the latter case. If far node JL bears a label, it is passed by, and the next arc is inspected. However, if JL is unlabeled, the next IF statements make an attempt to do so according to the labelling rules of subsection 2.4.3. If JL is labelled, the program transfers to statement 9 to assign a flow change label (ELABEL(JL)); if it is not, go to another arc.

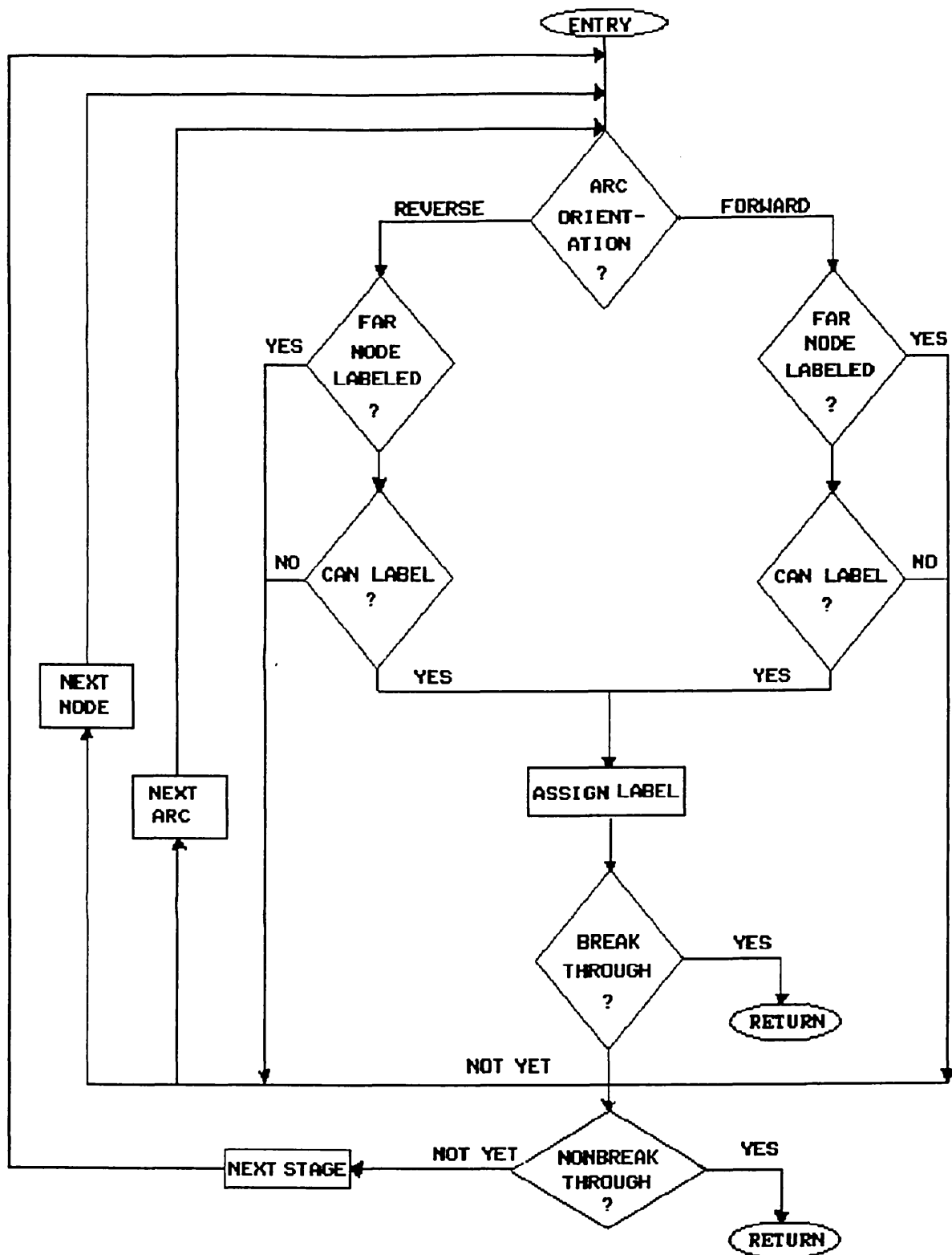


Fig A.5 LABELR flowchart

After labelling JL, the number of labelled node in the next stage (NEXT) increases and the node number JL is stored to be used in the next stage. If NEXT still zero at the end of the labelling technique, the routine reaches a non breakthrough. The other terminating condition (breakthrough) is when JL takes the value of TFL i.e. the terminal for labelling has been reached and a flow augmentation path is defined. Fig A.5 shows the flow chart for the labelling technique.

A7 Subroutine FLOCHA

After a successful determination of the flow augmentation path (breakthrough), comes the task of changing the flow along the arcs involved. Subroutine FLOCHA starts by computing the amount of flow change, EPSILN, which is evaluated by means of the FORTRAN function MIN that takes the value of the smallest of its arguments. This function ensures that the flow change procedure will not force any arc to be sent out-of-kilter, by keeping the flow change equal to the minimum permissible amount taking into consideration all arcs of the network.

The path is, then, backtracked from the terminal for labelling TFL, using the index IA which records the number of the present node, and moving along the tree of the path toward the origin for labelling OFL. The path label, KLABEL(IA), identifies the next arc of the path. If the arc is a forward arc the flow change is carried out within the loop 100 and then IA takes the number of the node on

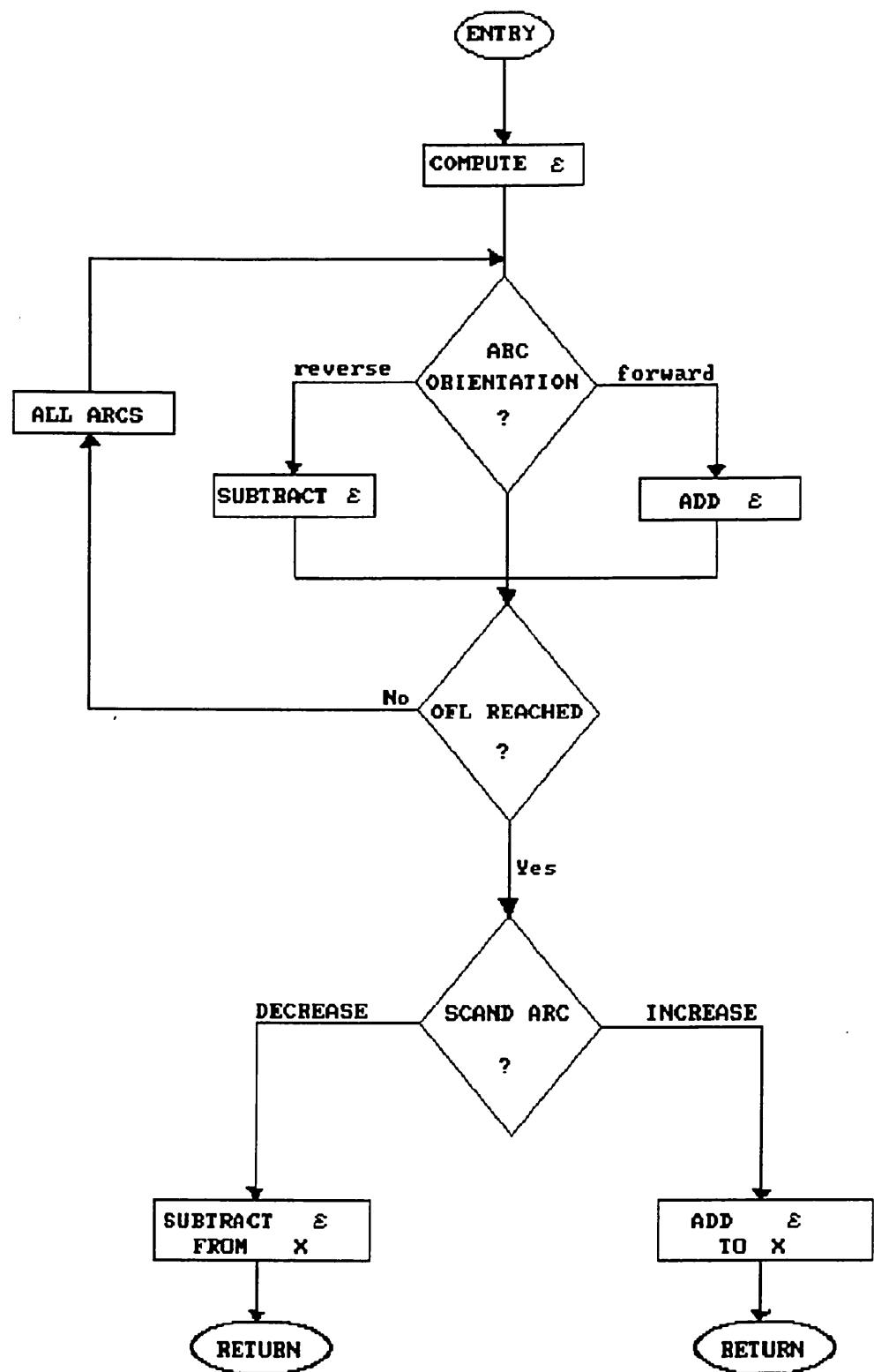


Fig A.6 FLOCHA flowchart

the other end of the arc. Do-Loop 200 is similar to Do-Loop 100 but for reverse arc location.

This procedure continues until IA takes the number of OFL which means that the path has been traced back. The state indicator, STAIN, set by FINDST, informs FLOCHA whether a flow increase or decrease is to be made. Fig A.6 shows the flowchart for FLOCHA.

A8 Subroutine NODPRI

Following a nonbreakthrough occurrence in subroutine LABELR due to the inability to define a flow augmentation path, NODPRI is called to change the node price (node potential) of unlabeled nodes which might enable a breakthrough in the next iteration.

In Do-Loop 100, the network arcs are inspected one at a time and classified whether they belong to set A1 or set A2 (defined in section 2.4.5). The node price increase DELTA is kept equal to the lower absolute value of $\overline{C}_{ij}$, using the MINO function. When Do-Loop 100 is satisfied, DELTA will contain the required node price increase.

NODPRI terminates in one of two ways and the appropriate information are sent to the main routine via the common blocks. This two terminating ways are:

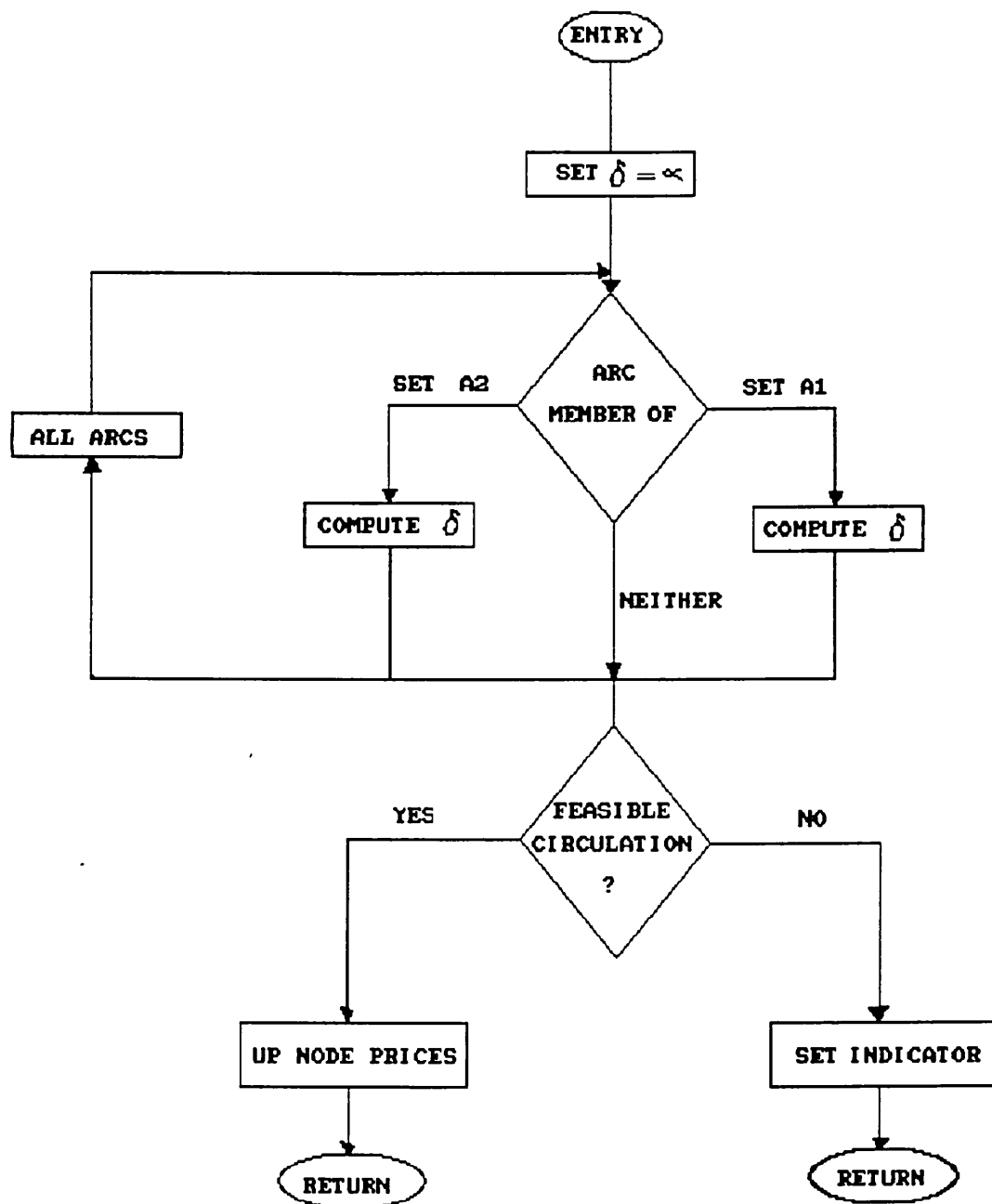


Fig A.7 NODPRI flowchart

1-DELTA has a finite value, which means that a node price increase is possible. Do-Loop 200 scans the nodes and undergoes a node price increase of the unlabeled nodes.

2-DELTA maintains its infinite value, set at the beginning of the subroutine, after the end of Do-Loop 100. This means that sets A1 and A2 are both empty and no-feasible flow terminating condition has been detected. Then FECIRC is set equal to zero, informing subroutine KILTAL to stop due to infeasibility. Fig A.7 shows the flow chart for subroutine NODPRI.

A9 COMMENTS

The performance of the Out-Of-Kilter algorithm could be improved in terms of computational time and memory requirements. This has been reported recently in the literature [10] where the author suggests some modifications of the OKA implementations. The final code is believed to be superior to any previously implemented version of the algorithm. The disadvantage of the code developed as the author put it " the code discriminates among problems of different structures".

APPENDIX B

C " ECONOMIC SCHEDULING OF ELECTRIC POWER SYSTEM "
C USES FULKERSONS OUT-OF-KILTER NETWORK FLOW ALGORITHM.
C ELECTRICAL ENGINEERING DEPARTMENT
C IMPERIAL COLLEGE

CHARACTER*6 STADES

INTEGER U,C,X,CBAR,S,T,PI,ELABEL,DIF,FECIRC,ARCS,
1STAIND,OFL,TFL,TBT,TNBT, STAGES,EPSILN,PRES,DIFL,

2SKJ,TKJ,XML,XMU,UMX,DELTA,SKN,TKN,PRICES,PRESTA

DIMENSION PRTITL(18)

COMMON /BKKIL/ L(1200),U(1200),C(1200),X(1200)

1,CBAR(1200),S(1200),T(1200),PI(400),KLABEL(400)

2,PRESTA(500),ELABEL(400),KS(2400),NODLOC(401),DIF

3,FECIRC,ARCS,NODES,STAIND,STADES,K,NEXT,OFL,TFL

4,INFINY,TBT,TNBT,LDFN,STAGES,DELTA,PRES

C ALPHABETIC STADES

C INPUT AND ASSEMBLY OF PROBLEM DATA

OPEN(5,FILE='INP.DAT;2',STATUS='OLD')

READ(5,901) PRTITL

901 FORMAT(18A4)

READ(5,905) ARCS,NODES

905 FORMAT(2I5)

READ(5,900)(K,S(K),T(K),L(K),U(K),C(K),X(K)

1,K=1,ARCS)

900 FORMAT(7I5)

DO 800 I=1,NODES

```

800 PI(I)=0
      READ(5,905) PRICES
      IF(PRICES.EQ.0) GO TO 8
      READ(5,905)(I,PI(I),II=1,PRICES)
8 CALL NODARC
      CALL CPIPI
      CALL KILTAL
      PRICES=NODES
      WRITE(7,901) PRTITL
      WRITE(7,902) ARCS,NODES
902 FORMAT( 8H ARCS = I4,3X,8HNODES = I4/)
      WRITE(7,903)
903 FORMAT(2X,3HARCS,4X,1HS,4X,1HT,4X,1HL,4X,1HU,4X
1,1HC,4X,1HX)
      WRITE(7,900)(K,S(K),T(K),L(K),U(K),C(K),X(K)
1,K=1,ARCS)
      WRITE(7,905) PRICES
      WRITE(7,904)
904 FORMAT(/5H NODE,3X,2HPI)
      WRITE(7,905) (I,PI(I),I=1,NODES)
      STOP
      END

```

SUBROUTINE KILTAL

```

C      OUT-OF-KILTER ALGORITHM TO FIND MINIMAL-COST NETWORK
C      FLOW FOR UPPER AND LOWER BOUND CAPACITATED NETWORK
      CHARACTER*6 STADES
      INTEGER U,C,X,CBAR,S,T,PI,ELABEL,DIF,FECIRC,ARCS,
1STAIND,OFL,TFL,TBT,TNBT, STAGES,EPSILN,PRES,DIFL,

```

```

2SKJ,TKJ,XML,XMU,UMX,DELTA,SKN,TKN,PRICES,PRESTA
  DIMENSION PRTITL(18)
  COMMON /BKKIL/ L(1200),U(1200),C(1200),X(1200)
1,CBAR(1200),S(1200),T(1200),PI(400),KLABEL(400)
2,PRESTA(500),ELABEL(400),KS(2400),NODLOC(401),DIF
3,FECIRC,ARCS,NODES,STAIN,STADES,K,NEXT,OFL,TFL
4,INFINY,TBT,TNBT,LDFN,STAGES,DELTA,PRES

  TBT=0
  TNBT=0
  FECIRC=-1
C      BRING EACH ARC K INTO KILTER
  DO 100 K=22,ARCS
C      FIND THE STATE OF ARC K
1  CALL FINDST
  IF(KLABEL(OFL).NE.0) GO TO 8
  PRES=1
  PRESTA(1)=OFL
C      IF ARC K IS IN KILTER,NO NEED TO LABEL
8 IF(STAIN.EQ.0) GO TO 100
C      LABELING PROCEDURE SEARCHES FOR A PATH
  CALL LABELR
C      TEST FOR BREAKTHROUGH
  IF(NEXT.EQ.0) GO TO 2
C      BREAKTHROUGH, CHANGE THE ARC FLOWS
  CALL FLOCHA
  TBT=TBT+1
  GO TO 1
C      NONBREAKTHROUGH, INCREASE NODE PRICES

```

```

2 CALL NODPRI
   TNBT=TNBT+1
C       TEST THE FEASIBLE CIRCULATION INDICATOR
   IF(FECIRC.NE.0) GO TO 3
   WRITE(6,4) TBT,TNBT
4 FORMAT('NO FEASIBLE SOLUTION TO THIS PROBLEM
1 EXISTS.BREAKTROUGH',I4,'NONBREAKTHROUGH',I4)
   RETURN
C       COMPUTE NEW OPTIMISING COST, CBAR
3 CALL CPIPI
   GO TO 1
100 CONTINUE
   FECIRC=1
   WRITE(6,5)
5 FORMAT('ALGORITHM TERMINATES WITH FEASIBLE
1OPTIMAL SOLUTION.')
```

6 WRITE(6,7) TBT,TNBT

7 FORMAT('BREAKTHROUGH',I4,3X,'NONBREAKTHROUGH',I4)

RETURN

END

SUBROUTINE LABELR

C **NODE LABELING PROCEDURE**

CHARACTER*6 STADES

INTEGER U,C,X,CBAR,S,T,PI,ELABEL,DIF,FECIRC,ARCS,

1STAIND,OFL,TFL,TBT,TNBT, STAGES,EPSILN,PRES,DIFL,

2SKJ,TKJ,XML,XMU,UMX,DELTA,SKN,TKN,PRICES,PRESTA

COMMON /BKKIL/ L(1200),U(1200),C(1200),X(1200)

1,CBAR(1200),S(1200),T(1200),PI(400),KLABEL(400)

```

2, PRESTA(500), ELABEL(400), KS(2400), NODLOC(401), DIF
3, FECIRC, ARCS, NODES, STAIND, STADES, K, NEXT, OFL, TFL
4, INFINY, TBT, TNBT, LDFN, STAGES, DELTA, PRES
    DIMENSION NPSTAG(400,2)

    INFINY=343597383
    PRES=1
    MP=1
    MN=2
    NPSTAG(1,MP)=OFL
    DO 200 IL=1,NODES
        KLABEL(IL)=0
200 ELABEL(IL)=0
C      LABEL THE ORIGIN FOR LABELING
        KLABEL(OFL)=9999
        ELABEL(OFL)=INFINY
1 NEXT=0
C      SCAN FROM ALL LABELED NODES IN PRESENT STAGE
    DO 100 IPS=1,PRES
        IL=NPSTAG(IPS,MP)
        NARCIL=NODLOC(IL+1)-NODLOC(IL)
C      SCAN ALL ARCS CONNECTED TO NODE IL
    DO 100 ISCAN=1,NARCIL
C      LOCATE A SIGNED ARC
        LSA=NODLOC(IL)+ISCAN-1
        KL=IABS(KS(LSA))
        JL=T(KL)
        ICOST=CBAR(KL)
        IF(JL.EQ.IL.AND.KL.NE.K) THEN

```

```

      IF(X(KL).EQ.0) JL=S(KL)
      ICOST=C(KL)+PI(T(KL))-PI(JL)
      END IF
      IF(KLABEL(JL).NE.0) GO TO 100
C          CASE 3.1.B. STATES BETA,BETA1,GAMMA1
      IF(ICOST)3,3,2
3  IF(X(KL).EQ.U(KL)) GO TO 100
      DIFL=U(KL)-X(KL)
      KLABEL(JL)=KL
      IF(JL.EQ.S(KL)) KLABEL(JL)=-KL
      GO TO 9
2  IF (X(KL).EQ.L(KL)) GO TO 100
      DIFL=L(KL)-X(KL)
      KLABEL(JL)=-KL
9  ELABEL(JL)=MIN(ELABEL(IL),ABS(DIFL))
C          STORE LABELED NODE JL FOR NEXT STAGE
      NEXT=NEXT+1
      NPSTAG(NEXT,MN)=JL
C          TEST FOR BREAKTHROUGH
      IF(JL.EQ.TFL) RETURN
100 CONTINUE
C          TEST FOR NONBREAKTHROUGH
      IF(NEXT.EQ.0) RETURN
      PRES=NEXT
      MD=MP
      MP=MN
      MN=MD
      GO TO 1
      END

```

SUBROUTINE CPIPI

C COMPUTES THE ENTIRE OPTIMISING COST ARRAY, CBAR
CHARACTER*6 STADES
INTEGER U,C,X,CBAR,S,T,PI,ELABEL,DIF,FECIRC,ARCS,
1STAIND,OFL,TFL,TBT,TNBT, STAGES,EPSILN,PRES,DIFL,
2SKJ,TKJ,XML,XMU,UMX,DELTA,SKN,TKN,PRICES,PRESTA
COMMON /BKKIL/ L(1200),U(1200),C(1200),X(1200)
1,CBAR(1200),S(1200),T(1200),PI(400),KLABEL(400)
2,PRESTA(500),ELABEL(400),KS(2400),NODLOC(401),DIF
3,FECIRC,ARCS,NODES,STAIND,STADES,K,NEXT,OFL,TFL
4,INFINY,TBT,TNBT,LDFN,STAGES,DELTA,PRES

DO 100 KJ=1,ARCS
SKJ=S(KJ)
TKJ=T(KJ)
100 CBAR(KJ)=C(KJ)+PI(SKJ)-PI(TKJ)
RETURN
END

SUBROUTINE FINDST

C FINDS THE STATE OF ARC K,AND CHOOSES APPROPRIATE CASE

CHARACTER*6 STADES
INTEGER U,C,X,CBAR,S,T,PI,ELABEL,DIF,FECIRC,ARCS,
1STAIND,OFL,TFL,TBT,TNBT, STAGES,EPSILN,PRES,DIFL,
2SKJ,TKJ,XML,XMU,UMX,DELTA,SKN,TKN,PRICES,PRESTA
COMMON /BKKIL/ L(1200),U(1200),C(1200),X(1200)
1,CBAR(1200),S(1200),T(1200),PI(400),KLABEL(400)
2,PRESTA(500),ELABEL(400),KS(2400),NODLOC(401),DIF

```

3, FECIRC, ARCS, NODES, STAIND, STADES, K, NEXT, OFL, TFL
4, INFINY, TBT, TNBT, LDFN, STAGES, DELTA, PRES
DATA ALF, BET, GAM, ALF1, ALF2, BET1, BET2, GAM1, GAM2
1/'ALPHA', 'BETA', 'GAMMA', 'ALPHA1', 'ALPHA2', 'BETA1'
2, 'BETA2', 'GAMMA1', 'GAMMA2' /

```

C COMPUTE DIFFERENCES FOR ARC K

```
XML=X(K)-L(K)
```

```
XMU=X(K)-U(K)
```

```
LMX=-XML
```

```
UMX=-XMU
```

C STATE-FINDING LOGIC

```
IF(CBAR(K)) 3,2,1
```

```
1 IF(XML) 10,5,22
```

```
2 IF(XMU) 4,5,21
```

```
4 IF(XML) 11,5,5
```

```
3 IF(XMU) 11,5,22
```

```
5 STAIND=0
```

```
RETURN
```

C OUT OF KILTER STATES

C ORIGIN FOR LABELING IS TERMINAL OF ARC K

```
10 DIF=LMX
```

```
GO TO 14
```

```
11 DIF=UMX
```

```
14 OFL=T(K)
```

```
TFL=S(K)
```

```
STAIND=1
```

```
RETURN
```

```

C          ORIGIN FOR LABELING IS START OF ARC K
21 DIF=XML
    GO TO 24
22 DIF=XMU
24 OFL=S(K)
    TFL=T(K)
    STAIND=2
    RETURN
    END

SUBROUTINE NODARC
C    BUILDS NODE TO ARC TABLE FROM ARC TO NODE TABLE
    CHARACTER*6 STADES
    INTEGER U,C,X,CBAR,S,T,PI,ELABEL,DIF,FECIRC,ARCS,
1STAIND,OFL,TFL,TBT,TNBT, STAGES,EPSILN,PRES,DIFL,
2SKJ,TKJ,XML,XMU,UMX,DELTA,SKN,TKN,PRICES,PRESTA
    COMMON /BKKIL/ L(1200),U(1200),C(1200),X(1200)
1,CBAR(1200),S(1200),T(1200),PI(400),KLABEL(400)
2,PRESTA(500),ELABEL(400),KS(2400),NODLOC(401),DIF
3,FECIRC,ARCS,NODES,STAIND,STADES,K,NEXT,OFL,TFL
4,INFINY,TBT,TNBT,LDFN,STAGES,DELTA,PRES

    LSA=1
    NODLOC(1)=1
    DO 100 IM=1,NODES
C          SCANS ARC TO NODE TABLE FROM ARCS
C          CONNECTED TO NODE IM
        NODLOC(IM)=LSA
        DO 200 KM=1,ARCS

```

```

        IF(S(KM).NE.IM.AND.IM.NE.T(KM)) GO TO 200
        KS(LSA)=KM
    3 LSA=LSA+1
200 CONTINUE
100 CONTINUE
        NODLOC(NODES+1)=LSA
        RETURN
    END

```

SUBROUTINE FLOCHA

C CHANGES THE FLOW IN ALL ARCS OF THE CYCLE JUST FOUND

```

        CHARACTER*6 STADES
        INTEGER U,C,X,CBAR,S,T,PI,ELABEL,DIF,FECIRC,ARCS,
1STAIND,OFL,TFL,TBT,TNBT, STAGES,EPSILN,PRES,DIFL,
2SKJ,TKJ,XML,XMU,UMX,DELTA,SKN,TKN,PRICES,PRESTA
        DIMENSION PRTITL(18)
        COMMON /BKKIL/ L(1200),U(1200),C(1200),X(1200)
1,CBAR(1200),S(1200),T(1200),PI(400),KLABEL(400)
2,PRESTA(500),ELABEL(400),KS(2400),NODLOC(401),DIF
3,FECIRC,ARCS,NODES,STAIND,STADES,K,NEXT,OFL,TFL
4,INFINY,TBT,TNBT,LDFN,STAGES,DELTA,PRES

```

C COMPUTE THE FLOW CHANGE. EPSILON

```

        EPSILN=MIN(ELABEL(TFL),DIF)

```

C BACKTRACK FROM THE TERMINAL FOR LABELING

```

        IA=TFL

```

C MAIN PATH TRACING LOOP

```

1 KA=IABS(KLABEL(IA))

```

```

C          TEST IF FORWARD OR REVERSE ARC IN THE
C          KLABEL AT NODE IA
          IF(KLABEL(IA).LT.0) GO TO 2
C          FORWARD ARC LOCATION AND FLOW CHANGE
          X(KA)=X(KA)+EPSILN
C          STEP TO NEXT NODE
          IA=S(KA)
          GO TO 3
C          SUBTRACT EPSILON FROM FLOW IN ARC KA
2 X(KA)=X(KA)-EPSILN
C          STEP TO THE NEXT NODE
          IA=T(KA)
C          TEST IF THE ORIGIN FOR LABELING HAS BEEN REACHED
3 IF(IA.NE.OFL) GO TO 1
C          CHANGE THE FLOW IN THE ARC K,BEING PUT IN KILTER
          IF(STAIND.EQ.2) GO TO 4
          X(K)=X(K)+EPSILN
          RETURN
4 X(K)=X(K)-EPSILN
          RETURN
          END

```

SUBROUTINE NODPRI

```

C          INCREASES THE NODE PRICES OF UNLABELED NODES

          CHARACTER*6 STADES
          INTEGER U,C,X,CBAR,S,T,PI,ELABEL,DIF,FECIRC,ARCS,
1STAIND,OFL,TFL,TBT,TNBT, STAGES,EPSILN,PRES,DIFL,
2SKJ,TKJ,XML,XMU,UMX,DELTA,SKN,TKN,PRICES,PRESTA

```

```

COMMON /BKKIL/ L(1200),U(1200),C(1200),X(1200)
1,CBAR(1200),S(1200),T(1200),PI(400),KLABEL(400)
2,PRESTA(500),ELABEL(400),KS(2400),NODLOC(401),DIF
3,FECIRC,ARCS,NODES,STAIND,STADES,K,NEXT,OFL,TFL
4,INFINY,TBT,TNBT,LDFN,STAGES,DELTA,PRES

C          COMPUTE DELTA,THE NODE PRICE INCREASE

DELTA=INFINY

DO 100 KN=1,ARCS

SKN=S(KN)

TKN=T(KN)

C          FIND IF ARC KN IS A MEMBER OF SET A1

IF((KLABEL(SKN).NE.O).AND.(KLABEL(TKN).EQ.O).AND.
1(CBAR(KN).GT.O).AND.(X(KN).LE.U(KN))) GO TO 1

C          FIND IF ARC KN IS A MEMBER OF SET A2

IF((KLABEL(TKN).NE.O).AND.(KLABEL(SKN).EQ.O).AND.
1(CBAR(KN).LT.O).AND.(X(KN).GE.L(KN))) GO TO 2

GO TO 100

1 DELTA=MIN(CBAR(KN),DELTA)

GO TO 100

2 DELTA=MIN(-CBAR(KN),DELTA)

100 CONTINUE

C          TEST FOR FEASIBLE CIRCULATION

IF(DELTA.EQ.INFINY) GO TO 9

C          INCREASE NODE PRICES OF UNLABELED NODES BY DELTA

DO 200 IN=1,NODES

IF(KLABEL(IN).EQ.O) PI(IN)=PI(IN)+DELTA

200 CONTINUE

RETURN

```

C NO FEASIBLE CIRCULATION

9 FECIRC=0

RETURN

END

APPENDIX C

We prove mathematically in appendix C that if in a resistive network the losses are minimised, the DC load flow distributes itself so that Kirchoff's voltage law is satisfied. It is known that in electric circuits the flows choose the least effort routes [47]. Thus the overall problem is

$$\text{Min } \sum I_k^2 R_k \quad k \in A \quad \text{C1}$$

$$\text{s.t } \sum I_k = B_i \quad \text{all } k \text{ connected to } i \quad \text{C2}$$

$$L_k \leq I_k \leq U_k \quad k \in A \quad \text{C3}$$

where I represents the current through a branch

R is the branch resistance

B is the current balance at each node

The optimality conditions of Kuhn & Tucker [11] can be stated as follows:

$$\sigma_k (I_k - L_k) = 0 \quad \text{C4}$$

$$\mu_k (U_k - I_k) = 0 \quad \text{C5}$$

$$I_k R_k = V_i - V_j + \sigma_k - \mu_k \quad \text{C6}$$

If line flows are not binding i.e. $I_k \neq L_k$ and $I_k \neq U_k$, equations C4 and C5 can only be satisfied if $\sigma_k = 0$ and $\mu_k = 0$. In that case, eq C6 becomes:

$$I_k R_k = V_i - V_j \quad \text{C7}$$

and Ohms's Law is met, when V_i and V_j are interpreted as the potentials at both ends (i and j) of a line k . From

the above demonstration, we have proved that a current respecting Kirchoff's first law and minimising losses, also satisfies Kirchoff's second law if all arcs are not binding i.e. unsaturated.

REFERENCES

1. S.N. Talukdar, F.F. Wu, "Computer-Aided Dispatch for Electric Power Systems", Proc. of the IEEE, Vol. 69, No10, 1981, pp. 1212-1231.
2. B.M. Weedy, "Electric Power Systems", John Wiley & sons, 3rd edition, 1984.
3. B.J. Cory, "Computer Applications in Power Systems", Proc. of the First Symposium on Electric Power Systems in Fast Developing Countries, Kingdom of Saudi Arabia, 1987, pp. 34-42.
4. J. Carpentier, "Towards a secure and optimal automatic operation of power systems", Proc. of the PICA, Montreal, May 1987, pp. 2-37.
5. J.L. Kennington, R.W. Helgason, "Algorithm for network programming", John Wiley & sons, N.Y, 1980.
6. D.R. Fulkerson, "An out-of-kilter method for minimal cost flow problems", JSIAM, Vol. 9, No1, March 1961, pp. 18-27.
7. J.F. Shapiro, "A note on the primal-dual and Out-Of-Kilter Algorithm for network optimization problems", Networks, Vol. 7, 1977, pp. 81-88.
8. R.S. Barr et al, "An improved version of the Out-Of-Kilter method and a comparative study of computer codes", Mathematical Programming 7, 1974, pp. 60-86.
9. H.A. AAshtiani, T.L. Magnanti, "Implementing primal-dual network flow algorithms", Working Paper OR 055-76, Operations Research Center, Massachusetts Inst. of Technology, Cambridge, Mass, 1976.

10. **S. Singh**, "Improved methods for storing and updating information in the Out-Of-Kilter Algorithm", Journal of the Association for Computing Machinery, Vol. 33, No3, 1986, pp. 551-567.
11. **H.W. Kuhn, A.W. Tucker**, "Non-linear Programming", Proc. of the Second Berkeley Symposium on Mathematical statistics and Probability, 1950.
12. **D.T. Philips, A. Garcia-Diaz**, "Fundamentals of Network Analysis", Prentice Hall, International Series in Industrial and Systems Engineering, 1981.
13. **D.G. Stone**, "Economic Scheduling by a Network Flow Method", Ph.D thesis, Imperial College, London, 1971.
14. **H.H. Happ**, "Optimal Power Dispatch - A Comprehensive Survey", IEEE Trans on PAS, Vol. PAS-96, No3, 1977, pp. 841-854.
15. **H.H. Happ et al**, "Large Scale Hydrothermal Unit Commitment - Method and Results", IEEE Trans. on PAS, Vol. PAS-90, No3, 1971, pp. 1373-1384.
16. **M.J. Steinberg, T.H. Smith**, "Economy Loading of Power Plants and Electric Systems", John Wiley & sons, 1943.
17. **E.C.M. Stahl**, "Load Division in Interconnections", Electrical World, March 1, 1930.
18. **M.J. Steinberg, T.H. Smith**, "The Theory of Incremental Rates", Part I, Electrical Engineering, March, 1934.
19. **J. Carpentier**, "Contribution a l'Etude Du Dispatching Economique", Bulletin de la Societe Francaise des Electriciens, March, 1963.

20. R.C. Burchett et al, "Developments in Optimal Power Flow", IEEE Trans. on PAS, Vol. PAS-101, No2, 1982, pp. 406-413.
21. T.H. Lee et al, "A transportation method for economic dispatching", IEEE Trans. on PAS, Vol. PAS-99, No6, 1980, pp. 2373-2385.
22. T.H. Lee et al, "Modified minimum cost flow dispatching method for power system application", IEEE Trans. on PAS, Vol. PAS-100, No2, 1981, pp. 737-745.
23. J.S. Luo et al, "Power system economic dispatch via network approach", IEEE Trans. on PAS, Vol. PAS-103, No6, 1984, pp. 1242-1248.
24. E. Hobson et al, "Network flow linear programming techniques and their application to fuel scheduling and contingency analysis", IEEE Trans. on PAS, Vol. PAS-103, No7, 1984, pp. 1684-1691.
25. M.F. Carvalho et al, "Optimal Active Power Dispatch by Network Flow Approach", IEEE PES Winter Meeting, Paper 88 WM 151-3, New York, 1988.
26. N.H. Dandachi, B.J. Cory, "An Out-Of-Kilter Algorithm for Active Power Economic Dispatch", IFAC Symposium on Power Systems & Power Plant Control, Seoul, 1989.
27. R.J. Ringlee, B.F. Wollenberg, "Overview of Optimization Methods", IEEE Tutorial Course Text 76CH1107-2-PWR, 1976, pp. 5-18.
28. J. Carpentier, "Optimal Power Flows", Electrical Power & Energy Systems, Vol. 1, April 1979, pp. 3-15.

29. B. Stott et al, "Review of Linear Programming Applied to Power System Rescheduling", IEEE, Proc. of PICA Conf., 1979, pp. 142-154.
30. G. Hadley, "Linear Programming", Addison-Wesley, 1962.
31. G.B. Dantzig, "Linear Programming and Extensions", Princeton University Press, 1962.
32. Discussion of ref 21.
33. J.S. Luo et al, "Bus Incremental Costs and Economic Dispatch", IEEE Trans. on Power Systems, Vol. 106 PWRS-1, 1986, pp. 161-167.
34. K.A. Clements et al, "Linear Programming vs Network Flow Applied to Bulk Power Supply Adequacy Assessment, IEEE PES Winter Meeting, Paper A78 062-2, New York, 1978.
35. IEEE Power Engineering Society Reliability Test System Task Force, IEEE Trans. on PAS-98, 1979, pp. 2047-2054.
36. B. Stott, "Review of Load Flow Calculation", Proc. of the IEEE, Vol. 62, No7, 1974, pp. 916-929.
37. D.A. Conner, "Representative bibliography on Load Flow Analysis and Related Topics", IEEE PES Winter Meeting, New York, Paper C73-104-7, 1973.
38. J.E. Van Ness, "Iterative methods for Digital Load Flow Studies", AIEE Trans. (Power App. Sys.), Vol. 78, 1959, pp. 583-588.
39. J.E. Van Ness, J.H. Griffin, "Elimination methods for Load Flow Studies", AIEE Trans. (Power App. Sys.), Vol. 80, 1961, pp. 299-304.

40. W.F. Tinney, J.W. Walker, "Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorization", Proc. of IEEE, Vol. 55, 1967, pp. 1801-1809.
41. W.F. Tinney, C.E. Hart, "Power Flow Solution by Newton's Method", IEEE Trans. on PAS, Vol. PAS-86, 1967, pp. 1449,1456.
42. B. Stott, O. Alsac, "Fast Decoupled Load Flow", IEEE Trans. on PAS, Vol. PAS-93, 1974, pp. 859-869.
43. J. Carpentier, "CRIC: A New Active Reactive Decoupling Process in Load Flows, Optimal power Flows and System Control", IFAC Symposium on Power System & Power Plant Control, Beijing, 1986.
44. D. Rajicic, A. Bose, "A Modification to the Fast Decoupled Power Flow for Network with High R/X Ratios", Proc. of PICA Conf., 1987, pp. 360-363.
45. A. Monticelli et al, "Fast Decoupled Load Flow: Hypothesis, Derivations and Testing", IEEE PES Winter Meeting, New York, Paper 89 WM 172-8 PWRs, 1989.
46. K. Takahashi et al, "Network flow Method Applied to Load Flow Calculation", IEEE Trans on PAS, Vol. PAS-87, No11, 1968, pp. 1939-1949.
47. J. Barras et al, "Network Simplex Method Applied to AC Load Flow Calculation", IEEE Trans on Power systems, Vol. PWRs-2, No1, 1987, pp. 197-203.
48. O.L. Elgerd, "Electric Energy Systems Theory", McGraw and Hill, New York., 1971.
49. B.M. Zhang et al, "A fast loop based algorithm for automatic contingency selection", IFAC, Power Sys-

- tems & Power Plant Control, Beijing, China, 1986, pp. 417-422.
50. D.C. Yu, S.T. Chen, "An Assessment of Programming Languages in Power Engineering Applications", Electric Power Systems Research, 15, 1988, pp. 99-105.
 51. F. Albuyeh et al, "Reactive power Considerations in Automatic Contingency Selection", IEEE Trans. on PAS, Vol. PAS-101, 1982, pp. 107-112.
 52. G.C. Ejebe et al, "Fast Contingency Screening and Evaluation for Voltage Security Analysis", IEEE/PES 1988 Winter Meeting, New York, Paper 88 WM 161-2.
 53. U.G. Knight, "Power Systems Engineering and Mathematics", Pergamon press, 1972.
 54. G. Stagg, A. El-Abiad, "Computer methods in power systems analysis", McGraw and Hill, New York, 1968.
 55. B.M. Zhang et al, "A Noval Loop based Algorithm and its Applications in On-line Security Assessment of Power Systems", Proc. of the 9th PSCC, Portugal, 1987, pp. 800-803.
 56. K.R.C. Mamandur, R.D. Chenoweth, "Optimal control of reactive power flow for improvements in voltage profiles and real power loss minimization", IEEE Trans. on PAS, Vol. PAS-100, No7, 1981, pp. 3185-3193.
 57. E. Hobson, "Network Constrained Reactive Power Control Using Linear Programming", IEEE Trans. on PAS, Vol. PAS-99, No3, 1980, pp. 868-877.
 58. M.A. El-Kady et al, "Assessment of real-time optimal voltage control", IEEE Trans. on Power Systems, May 1986, pp. 98-107.

59. **IEEE Tutorial Course**, "Reactive Power: Basics, Problems and Solutions", Course Text 87 EH0262-6-PWR.
60. **AIEE Committee Report**, "Application of Probability Methods To Generating Capacity Problems", Trans. of AIEE, Vol. 80, 1961, pp. 1165-1177.
61. **S.M. Dean**, "Consideration involved in making system investments for improved service reliability", EEI Bulletin, Vol. 6, 1938, pp. 491-496.
62. **I. Bagowsky**, "Reliability theory and practice", Prentice Hall, 1961.
63. **W.J. Lyman**, "Fundamental consideration in preparing a master system plan", Electrical World, 101, 1933, pp. 788-792.
64. **S.A. Smith**, "Space capacity fixed by probabilities of outage", Electrical World, 103, 1934, pp. 222-225.
65. **P.E. Benner**, "The use of the theory of probability to determine spare capacity", General Electric Review, 37, 1934, pp. 345-348.
66. **G. Calabrese**, "Generating reserve capacity determined by the probability method", Trans. of AIEE, 66, 1947, pp. 1471-1477.
67. **H. Halperin, H.A. Adler**, "Determination of reserve generating capacity", Trans. of AIEE, PAS-77, 1958, pp. 530-544.
68. **R. Billinton, M.P. Bhavaraju**, "Transmission planning using a reliability criterion- Part I- A reliability criterion", IEEE Trans. on PAS, Vol. PAS-89, 1970, pp. 28-34.

69. R. Billinton, M.P. Bhavaraju, "Transmission planning using a reliability criterion- Part II- Transmission planning", IEEE Trans. on PAS, Vol. PAS-90, 1971, pp. 70-78.
70. IEEE Committee Report, "Bibliography on the application of probability methods in power system reliability evaluation, 1971-1977", IEEE Trans. on PAS, Vol. PAS-97, 1978, pp. 2235-2242.
71. R.N. Allan et al, "Bibliography on the application of probability methods in power system reliability evaluation, 1977-1982", IEEE Trans. on PAS, Vol. PAS-103, 1984, PP. 275-282.
72. S.H. Lee, "Reliability evaluation of a flow network" IEEE Trans. on Reliability, Vol. R-29, No 1, 1980, pp. 24-26.
73. R.N. Allan, A.M. Al-Said, "Reliability and production cost of interconnected systems", Proc. of the 9th PSCC, Portugal, 1987, pp. 460-466.
74. S.H. Ahmad, "Simple enumeration of minimal cutsets of Acyclic directed graph", IEEE Trans. on Reliability, Vol. 37, No 5, 1988, pp. 484-487.
75. C.K. Pang, A.J. Wood, "Multi-Area generation system reliability calculations", IEEE Trans. on PAS, Vol. PAS-94, 1975, pp. 508-517.
76. H.Baleriaux et al, "An original method for computing short-fall in power systems", CIGRE Paper 32-09, 1974.
77. R.L. Sullivan, "Power System Planning", McGraw and Hill, 1977.

78. G.C. Oliveira et al, "A direct method for multi-area reliability evaluation", IEEE/PES 1986 Summer Meeting, paper 86 SM 351-1, Mexico City 1986.
79. G.C. Ejebe, "A maximal flow technique for computing shortfall in power systems", ANSTI EE 86, 18th- 20 Aug 1986, Dar Es Salam, Tanzania.
80. M.J. Juricek et al, "Transportation analysis of an electric power distribution system", IEEE/PES Winter Meeting, 1976, Paper A 76, 052-1.
81. P. Douillez, "Optimal Capacity planning of multi-terminal networks", Ph.D. Thesis, Universite Catholique de Louvain, Belgium, 1970.
82. S. Wang et al, "A research on flows in complete stochastic network", Proc. of the 9th PSCC, Portugal 1987, pp. 467-473.
83. X. Wang, Q. Sun, "A network flow approach to reliability studies of multiarea interconnected power systems", Journal of Xi'an Jiaotong Univ., No 4, 1984, pp. 41-52.
84. D. Sjelvgren, T.S. Dillon, "Seasonal planning of a hydrothermal system based on the network flow concept", Proc. of the 7th PSCC, 1981, pp.1191-1199.
85. H. Habibollazade et al, "Optimal short term operation planning of hydrothermal power system - part II: solution techniques", Proc. of the 8th PSCC, 1984, pp. 329-336.
86. H. Habibollazade, J.A. Bubenko, "Application of decomposition technique to short term operation planning of hydrothermal power system", IEEE Trans.

- on Power systems, Vol. PWRS-1, N°1, 1986, pp. 41-47.
87. H. Brannlund et al, "Optimal short term operational planning of a large hydrothermal power system based a non-linear network flow concept", IEEE Trans. on Power systems, Vol. PWRS-1, N°4, 1986, pp. 75-82.
88. H. Brannlund et al, "Optimal generation scheduling of large hydrothermal power system by network programming method", Proc. of the 9th PSCC, 1987, pp. 56-60.
89. A. Merlin et al, "Optimisation of short term scheduling of EDF hydraulic valleys with coupling constraints: the ovide model", Proc. of the 7th PSCC, 1981, pp. 345-352.
90. A. Johannesen et al, "Short term scheduling of large hydroelectric power systems", IFAC symposium on planning and operation of electric energy systems, Rio De Janeiro, 1985, pp. 99-106.
91. A.M. Geoffrion, "Generalized Benders decomposition", Journal of optimization theory and applications, Vol. 10, N°4, 1972, pp. 237-260.
92. P. Pettit, "Digital computer programs for an experimental automatic load-dispatching system", Proc. of IEE, Vol. 115, 1968, pp. 597-605.
93. T.K. Hon et al, "Emergency Rescheduling by a Network Flow Method", Proc. of the 2nd PSMC, Durham, 1986, pp. 113-118.
94. S.M. Chan, E. Yip, "A solution of the transmission limited dispatch problem by sparse linear programming", IEEE Trans. on PAS, Vol. PAS-98, 1979, pp. 1044-1053.

95. K.W. Doty, P.L. McEntire, "An analysis of electric power brokerage systems", IEEE Trans. on PAS, Vol. PAS-101, No2, 1982, pp. 389-396.
96. D.W. Wells, "Method for economic secure loading of a power system", Proc. of IEE, Vol.115, 1968, pp. 1190-1194.
97. C.M. Shen, M.A. Laughton, "Power system load scheduling with security constraints using dual linear programming", Proc. of IEE, Vol. 117, 1970, pp. 2117-2127.
98. C.A. Rossier, A. Germond, "Network topology optimization for power system security enhancement", CIGRE-IFAC Symposium, Florence, 1983, part 206-01, pp. 1-5.
99. K.R. Padyar, R.S. Shanbag, "Comparison of methods for transmission system expansion using network flow and DC load flow models", Electrical Power & Energy Systems, Vol. 10, No1, 1988, pp. 17-24.
100. J. Carpentier et al, "Voltage collapse proximity indicators computed from an optimal power flow", Proc. of the 8th PSCC, Helsinki, 1984, pp. 671-678.
101. N. Dandachi, Private communication
102. M. Winokur, B.J. Cory, "Identification of strong and weak areas for emergency state control", IEEE Trans. on PAS, Vol. PAS-103, No6, 1984, pp. 1486-1492.
103. B.M. Zhang et al, "A practical algorithm for finding a set of independent loops with quasi minimum incidence degree", Proc. of CAD Symposium of the CSEE, Changsha, China, 1985.