ECONOMIC SCHEDULING OF ELECTRIC POWER GENERATION BY A NETWORK FLOW METHOD

David Gerald Stone

A thesis submitted for the degree of Doctor of Philosophy in Electrical Engineering

May 1971

Department of Computing and Control
Imperial College of Science and Technology
University of London

CONTENTS

	Page
ABSTRACT	1
ACKNOWLEDGEMENTS	2
LIST OF SYMBOLS	3
Chapter 1 INTRODUCTION	6
1.1 Power system operation	7
1.1.1 Load prediction	7
1.1.2 Fuel supply	8
1.1.3 Generating unit performance evaluation	9
1.1.4 Security requirements	9
1.1.5 Power transfers	11
1.1.6 Switching	11
1.1.7 Commitment of generating units	12
1.1.8 Frequency control	12
1.2 Approaches to the economic operation problem	14
1.3 A complex variable formulation of the economic scheduling problem	18
1.4 Selecting a mathematical programming method	20
1.5 The network flow method	22
1.6 Contribution of the thesis	24
Chapter 2 A NETWORK FLOW FORMULATION OF THE ECONOMIC SCHEDULING PROBLEM	26
2.1 Branch model	26

2.2 Generator model 33

2.3 Modelling demands and imports 40

2.4 The complete model 42

2.4.1 The return arc 42

2.4.2 Power balance equations 42

2.4.3 Cost form 43

2.4.4 A small example 44

Chapter 3 COMPUTATIONAL EXPERIMENTS IN ECONOMIC SCHEDULING 47

3.1 Purpose and method 47

3.2 Test power systems summary 48

3.3 Computer programs description 49

3.3.1 MULTAR, Multiple arc out-of-kilter network flow routine 49

3.3.2 CMLFP, Conversational mode load flow program 50

3.3.3. NFESP, Network flow economic scheduling program 51

3.4 Optimal economic dispatch for complex power system models 51

3.5 Applying the network flow schedule to the complex model 53

3.6 Computational results 60

3.6.1 British test systems 60

3.6.2 American test systems 74

Chapter 4 FURTHER INVESTIGATIONS 90
4.1 Cost of load prediction errors 90
4.2 Optimizing the voltage profile 91
4.3 Cost of generator data inaccuracies 92
4.4 Including estimated losses with demands 93
4.5 Comparison of branch power flows 94
4.6 Generator model additions 95
4.7 Transmission security 96

Chapter 5 CONCLUSIONS 98

REFERENCES 105

Appendix 1 MULTAR, Multiple arc out-of-kilter network flow routine 115
Appendix 2 CMLFP, Conversational mode load flow program 172
Appendix 3 PROGRAM DESCRIPTION: NFESP, NETWORK FLOW ECONOMIC SCHEDULING PROGRAM 239
Appendix 4 TEST POWER SYSTEMS 261

ABSTRACT

The power system economic load dispatching task is formulated as a minimal-cost capacitated network flow problem. Then D.R. Fulkerson's 'Out-of-Kilter' method is used to solve the power system network flow model for the economic schedule of generation. Multiple-arc branch models form piecewise linear approximations to the quadratic cost of transporting power. Generator cost of production functions are easily and accurately modelled by one network arc per range of constant incremental cost. In a series of computational experiments on test systems of various sizes the validity, speed, and accuracy of the method was confirmed. Documentation of the FORTRAN computer programs written for the experiments is given. These are: the network flow solution algorithm, a conversational mode load flow program, and an economic scheduling program. A large power system with 177 generators was solved in 3 minutes of computing time on the IBM7094. Test data and results for smaller systems are included in the thesis.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor J.H. Westcott, for his guidance, support, and encouragement. Thanks are also due the Central Electricity Generating Board who provided financial sponsorship of the work in the form of a research contract. I am grateful also for various helpful discussions with E.D. Farmer of the Central Electricity Research Laboratories; our technical correspondent for the research contract. L.J. Sweett, Planning Department, CEGB Headquarters, kindly made available test systems data, and shared with me some of his operational experience.

LIST OF SYMBOLS

a_h	quadratic cost coefficient, branch h
a_m	cost coefficient in the quadratic term, generator m
b_m	cost coefficient in the linear term, generator m
c_i	reactive power demanded at bus i
c_k	arc cost
c_m	no-load cost, generator m
d_i	real power demanded at bus i
f_h	cost of transportation, branch h
f_m	cost of generation, generator m
f_T	total cost of generation
g_i	real power generated at bus i
g_m	real power output, generator m
g_m^{max}	maximum real power output, generator m
g_m^{min}	minimum real power output, generator m
g_m^s	upper breakpoint, segment s, generator m
h	branch index
i	bus index, node index
j	bus index, node index
k	arc index
ℓ_h	power losses in branch h
ℓ_k	lower bound, arc k
ℓ_s	system transmission losses
m	generator index
n_{brn}	number of power system branches
n_{bus}	number of power system busses

n_h	tapping ratio, transformer h
n_h^{max}	maximum tapping ratio, transformer h
n_h^{min}	minimum tapping ratio, transformer h
p_h	real power flow in branch h
p_h^{fwd}	forward power flow limit, branch h
p_h^{rvs}	reverse power flow limit, branch h
p_h^s	upper breakpoint, segment s, branch h
p_i	real power impressed on the network at bus i
q_i	reactive power impressed on the network at bus i
r_i	reactive power generated at bus i
r_m	reactive power output, generator m
r_m^{max}	maximum reactive power output, generator m
r_m^{min}	minimum reactive power output, generator m
s	segment index
s_f	scale factor for arc costs
t	top segment
u_k	upper bound, arc k
v_i	voltage magnitude, bus i
v_i^{max}	maximum voltage magnitude, bus i
v_i^{min}	minimum voltage magnitude, bus i
x_k	flow, arc k
z	total flow cost
A	set of arcs in the network
A_h	set of arcs modelling branch h
A_h^{act}	set of active arcs, branch h
A_h^{fwd}	set of arcs modelling forward flow, branch h

A_h^{rvs}	set of arcs modelling reverse flow, branch h
A_i^+	set of arcs radiating from node i
A_i^-	set of arcs impinging upon node i
A_m	set of arcs modelling generator m
A_m^{act}	set of active arcs, generator m
B_i	set of branches connected to bus i
B_s	set of branches in the power system
D_g	set of demands in group g
D_s	set of system demands
E_g	group export power
G_g	set of generators in group g
G_i	set of generators supplying bus i
G_r	set of generators operating 'in range'
G_s	set of generators supplying the power system
I_g	group import power
S_i	apparent power impressed on the network at bus i
V_i	voltage, bus i
Y_{ij}	mutual admittance between bus i and bus j
α	source node
δ_h^{max}	maximum torque angle, branch h
λ^{avg}	system average incremental cost of generation
λ_i	incremental cost of generation at bus i
λ_m	incremental cost of generation, generator m
λ_m^s	incremental cost of generation, segment s, generator m
Θ_i	voltage angle, bus i
ω	sink node

Chapter 1 INTRODUCTION

Economic scheduling of generation to meet the system load is an important task in the daily operation of electric power systems. It aims at dispatching the load to the various generating units in such a way as to minimize the total cost of supply. Regarding the complete power system as an industrial plant; the scheduling task can be seen to be one of process optimization.

Even prior to the advent of high-speed digital computers, electric power utilities were achieving the optimal operation of their systems by the best manual techniques then available.(32) Then with the wide-spread use of digital computing machinery in the power supply industry came the application of these machines to the economic load dispatching problem. Familiarity with the growing computer power at their disposal prompted researchers to propose new formulations of the economic scheduling problem, and to attempt their solution. Some of these approaches are classified in section 1.2. The goal of these efforts is to find an expression of the problem that is as exact as necessary, and yet computationally practicable. This thesis presents a new method of economic scheduling that meets that goal.

1.1 Power system operation

The day-to-day running of an electric power system is a complex operation. For convenience it has been subdivided into a number of traditional activities, one of which is that of economic scheduling, or load dispatching. The purpose of this section is to describe these operational tasks, and their interaction with the economic scheduling problem. Thus, a basis for discussions coming later in the thesis will have been established, and the assumptions inherent in the economic scheduling problem will be placed in evidence.

1.1.1 Load prediction

Calculation of an economic schedule of generation preceeds the actual event by a matter of minutes, or in some cases even days, as with an off-line predictive study. Instead of recorded real power demand levels, predicted values for those demands are supplied to the scheduling program, and used in calculating target set points for the generators. That is, a predictive economic generation schedule is computed. The closeness of the predicted demands to the actual demands experienced in the event, is a crucial factor in achieving economy in system operation.

If the load is to be predicted 24 hours in advance, the prediction error will be greater than that for a one

hour prediction. In the former case the prediction error may be large enough to render the use of an extremely accurate scheduling technique inappropriate. If the prediction period is short, say fifteen minutes, an accurate scheduling method is warranted, because the prediction itself will be accurate. However, it is unlikely in this case that sufficient computing time will be available to solve the problem to a high degree of accuracy. Therefore, it can be seen that a balance must be struck in selecting an approach for economic scheduling.

1.1.2 Fuel supply

The transportation network over which fuel supplies reach the generating stations may be as complex as the electrical power transmission system itself. A global optimization of the fuel transportation and power supply systems taken together is necessary for the longer-range planning of power supplies.(27,28) However, for short-term economic scheduling of electric power, these two networks are assumed to be separted by the relative inflexibility of fuel transport facilities. The global long-term optimization gives the cost of fuel at each generating station. This cost is made available to the economic scheduling program for its short-term calculations.

1.1.3 Generating unit performance evaluation

By monitoring the performance of each set, up-to-date heat rates are obtained for the range of output power of the generator. Combining this function with the fuel cost gives the operational cost curve to be used in the economic scheduling task. With the majority of British sets, the cost curve is a straight line, and its slope, the incremental cost of generation, defines the position of the unit in the list called the Merit-Order Table 'B'.

1.1.4 Security requirements

The task of operating a system so as to obtain the highest degree of security of supply consistent with good cost economics is difficult to specify. It is linked to the economic scheduling problem and the plant ordering problem, as well as a number of other operational activities. A complete, exact formulation and solution of the security checking problem lies beyond the state of the art.(58) In practice, a number of approximate methods of predictive security checking have evolved to help ensure reliability of supply.

The transmission network is normally designed to permit economic transfer of large blocks of power from one geographical location to another, without stability problems. Another function of the network is to enhance the security of supply by making a sufficient number of interconnections between the supply and demand points.

An operative definition of power system security in common use is the following: the system must continue to supply its demand to specification, despite the loss of any one item of plant. Of central concern is the loss of a large generator or an important transmission line, so we confine the discussion to these two contingencies.

When a generator is lost, the system frequency drops and governor action on the remaining generators increases their output to meet the demand. A number of considerations apply here. First, the absolute spinning spare margin in megawatts must be available to meet the loss, that is, all sets must not be operating at or near full load. Second, sufficient spare capacity must be present to allow the generator outputs to rise quickly enough. The spare available from each set is a function of its loading rate, and will often be less than the difference between its present power output and its capacity. This available spare may be reduced further during periods of sharply rising demand.

The effect of these requirements is that the spinning reserve must often be distributed among a number of different sets. As a consequence, it can happen that the economic schedule of generation must be altered to provide the spinning spare capacity necessary for secure system operation.(57,6)

Losing a transmission line may cause other lines to

overload beyond their thermal limits, threatening further outages and a system collapse. To secure the transmission system against such an eventuality the geographical pattern of generated power must be modified to reduce the line flows. Thus the system load cannot safely be dispatched to the most economic plant, and the generation schedule must be adjusted.

1.1.5 Power transfers

Previously scheduled interchanges of power with outside areas are accounted for in the economic scheduling task as fixed imports or exports. Hydro generation and pumped storage pumping and generation schedules are also given as input data for the economic scheduling of the thermal generating sets.

1.1.6 Switching

The switched configuration of the transmission network changes frequently for a variety of reasons: voltage control, reduction of fault levels, and equipment maintenance and construction are some of them. Therefore, before an economic scheduling study can be undertaken, the transmission plant in service must be specified, as well as its connection.

1.1.7 Commitment of generating units

The act of selecting which sets to run is called unit commitment, or economic ordering of generation. During the 24 hour load cycle, variations in the load magnitude are great enough to make the shutting down and starting up of generating sets an economic necessity. Depending on the thermal condition of the unit, a period of between one hour and several hours is required to bring a generator on line and up to full power output. The unit commitment problem is a complicated one, and is at present handled in a heuristic manner. A load prediction several hours in advance will dictate the amount of generation ordered, and generation economics and system security assessment will indicate which generators should be synchronized to the system to meet the load.

The list of committed units is thus an input to the economic scheduling task. If the ordering function has been done properly, the scheduling program will have no difficulty in developing a secure, economic schedule of generation with the units that have been ordered.

1.1.8 Frequency control

Governor action in regulating the system frequency against short term random load fluctuations takes the regulating generators away from their economic set points. Until a new economic schedule is computed, based on an

updated load prediction, the generators meet the demand in an uneconomic way.

An economic schedule of generation may leave but one or two units operating 'in range' and available for regulating the frequency. In this case, the loading rate constraints on the sets may not permit them to meet rapid changes in demand. Consequently, the economic schedule must be changed to allow additional units to assist in the task of system frequency regulation.

It may be desirable to spread the regulating task among several units in order to reduce maintenance costs.(11)

At a given point in time, although the generators are meeting load changes satisfactorily, the economic schedule must be modified to develop a spare regulating capacity against expected large increases in load. This situation is usual prior to the morning load rise.

The above considerations aim, in effect, at transforming what is essentially a dynamic problem into a sequence of static ones. This traditional approach is followed in expressing the problem in a form suitable for a static economic scheduling method.

If we recall the discussion on system security, it can be observed that spinning spare capacity serves the dual function of frequency regulation and security maintenance. Writing an overall system power balance

equation:

$$\sum_{m \varepsilon G_s} g_m = \sum_{i \varepsilon D_s} d_i + \ell_s \qquad \text{MW}$$

helps clarify this point. Thus a sharp rise in demand, or the loss of a generator are seen to produce the same imbalance in total system power, and the effect on the remaining generators is quite similar. G_s is the set of generators currently supplying the system, D_s the set of demand busses, and ℓ_s the system losses.

1.2 Approaches to the economic operation problem

Having accounted for the interaction of such operational tasks as load prediction and security assessment with the problem of economic operation; we are now free to consider the latter question in detail. A complete, exact treatment of the dynamic economic power system operation problem is not as yet amenable to solution. Consequently, approaches to the problem taken by past workers have decomposed it in a variety of ways, treating certain sub-problems deemed to be of central importance and utility. By characterizing some of their work, the method of the present thesis is described in scope, and related to past efforts.

The central problem of economic operation can be characterized by four dichotomies:

1. Instantaneous, static, vs. extended period, dynamic
2. No electrical network, vs. a network representation
3. Unit commitment given, vs. unit commitment task to be undertaken
4. Economic scheduling given, vs. economic scheduling task to be done

In each of these, the second member represents an additional complication to the formulation and an additional computational burden in solution.

The binary nature of this characterization suggests its graphical display in a form similar to a Karnaugh Map. The four dichotomies are ranged on the four sides of the display, as shown in Figure 1-1. If a value 0 is assigned to the first member of each dichotomy and a value 1 to the second; and these values summed for each combination on the map, a rough index of difficulty for each sub-problem has been introduced. The index of difficulty ranges from 0, no problem; to 4, the complete problem. Each square in the map represents a unique combination of problem components. A square, or sub-problem is referenced as one would a matrix element: (i, j) ith row and jth column.

In the various squares are listed the reference numbers of papers with the corresponding combination of problem elements. Sub-problems in the right hand column correspond to solutions already in hand. At the other

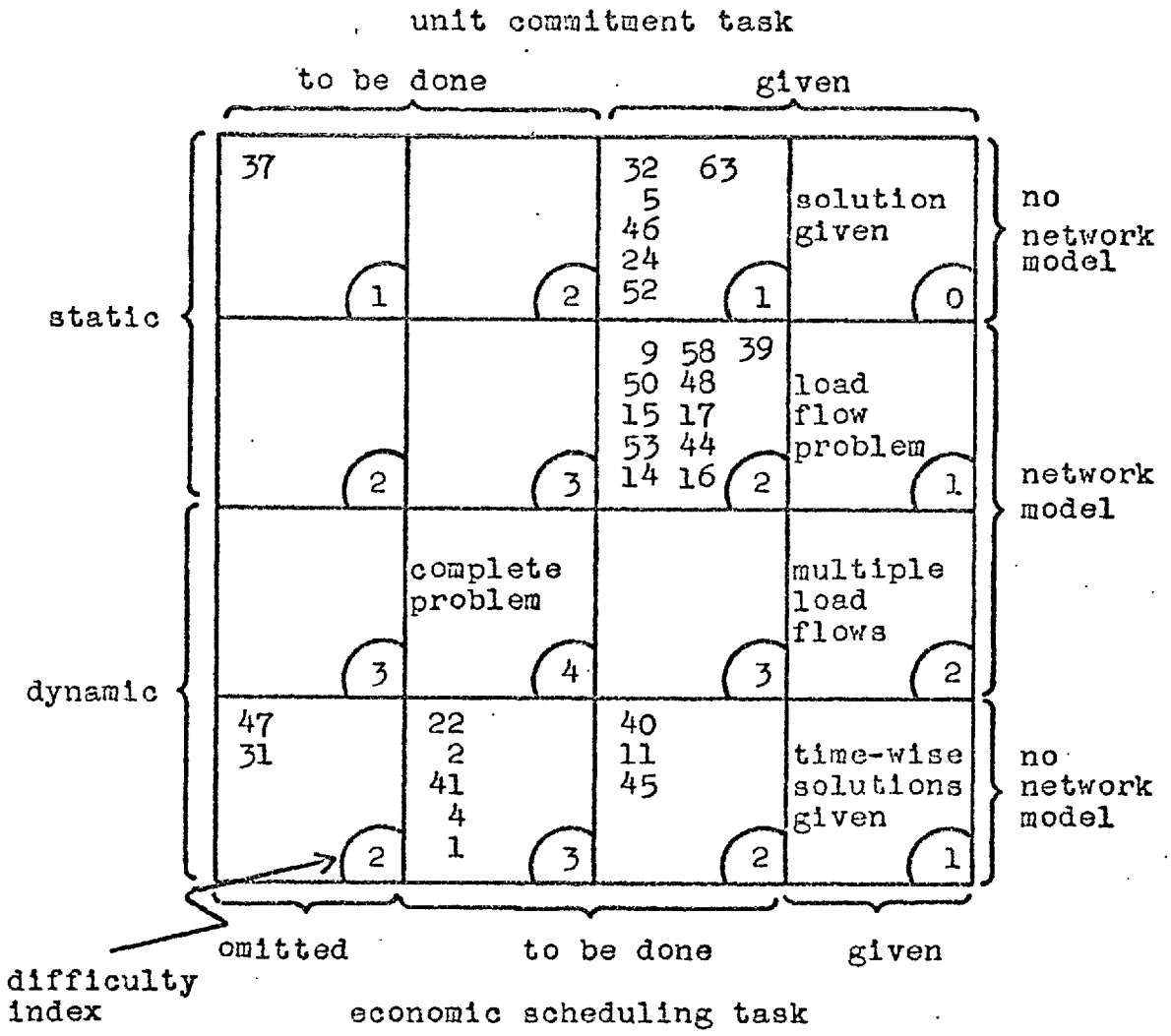


Figure 1-1 Sub-problems in economic power system operation displayed as a Karnaugh Map

extreme, square (3,2) represents the complete problem, the absence of entries indicating a lack of formal attempts to solve it.

Sub-problem (2,3) has received the greater part of research attention in recent years. The present thesis belongs in this category. Knowing the units committed to operate, an economic schedule of generation is sought that will supply a given instantaneous demand pattern. In addition, some form of network model is included in the formulation.

The categorization given here is a crude one, and research contributions vary considerably in the details of their formulation. For example, three types of transmission network model may be distinguished:

1. A network flow model giving an arbitrary, conservative flow of real power.
2. A D.C. approximation neglecting transmission losses and defining a linear model of the network, which gives a flow of real power that obeys Ohm's Law.
3. A complete complex variable network model based on pi-equivalent circuits for the transmission lines.

These are of sharply increasing complexity, and the index of difficulty could be modified accordingly.

Our classification scheme and Figure 1-1 has presented an overview of the relevant research, helped to define the scope of this thesis, and shown it to be in the main-

stream of present research in economic power system operation. Figure 1-1 also indicates the direction of future work, which is expected to pass through sub-problems of increasing difficulty, until the complete problem (3,2) can be attempted.

1.3 A complex variable formulation of the economic scheduling problem

A formulation of the load dispatching problem, square (2,3) which includes a complete complex variable model of the transmission network may be regarded as a canonical one. It is provided here to serve as an introduction, to permit comparison with the network flow approach, and to form a basis for the check solutions in Chapters 3 and 4.

Minimize the total cost, f_T, of supplying the system load:

$$\min\{f_T\} \quad , \quad f_T = \sum_{m \epsilon G_s} f_m(g_m)$$

where G_s is the set of generators now supplying the system, f_m is the cost of generation function, and g_m is the generated power sent out by generator m. The minimization is subject to constraint by the network laws, plant capacities, and statutory limits, as given below.

1. The real and reactive power balance equations must be satisfied at each bus:

$$\left.\begin{aligned} g_i &= d_i + p_i \\ r_i &= c_i + q_i \end{aligned}\right\} \quad i = 1,2,\ldots,n_{bus}$$

where $g_i = \sum_{m \in G_i} g_m$ is the real power generated at bus i, G_i being the set of generators connected to bus i. These generators also produce r_i MVAr of reactive power. d_i MW of real power and c_i MVAr of reactive power are demanded at bus i. The apparent power impressed on the network at bus i is $S_i = p_i + jq_i$ MVA.

2. The network equilibrium equations resulting from Ohm's and Kirchhoff's Laws must also be satisfied:

$$S_i = V_i \sum_{j=1}^{n_{bus}} Y^*_{ij} V^*_j \qquad i = 1,2,\ldots,n_{bus}$$

where $V_i = v_i e^{j\theta_i}$ is the complex voltage at bus i, v_i being its magnitude, and θ_i its angle. Y_{ij} is the mutual admittance between bus i and bus j.

3. The capability of a typical generator to produce apparent power is a complicated function defined by several different technological limits. For economic scheduling purposes the following simple inequalities adequately limit the output power:

$$\left.\begin{aligned} g_m^{min} \leq g_m \leq g_m^{max} \\ r_m^{min} \leq r_m \leq r_m^{max} \end{aligned}\right\} \quad m \in G_s$$

The reactive limits should include the effect of the generator transformer.

4. Narrow limits on the bus voltage magnitude are normally

required in order to ensure quality of supply to the customer.

$$v_i^{min} \leq v_i \leq v_i^{max} \qquad i = 1,2,\ldots,n_{bus}$$

5. A variety of factors, as discussed later in section 2.1, combine to require the placing of a limit on the flow of power in each branch. A convenient expression of this requirement is in terms of the branch torque angle, $\theta_i - \theta_j$:

$$|\theta_i - \theta_j| \leq \delta_h^{max} \qquad h = 1,2,\ldots,n_{brn}$$

where branch h connects bus i with bus j.

6. Transformer tapping ratios must take discrete values corresponding to tap positions

$$n_h = n_h^{min}, n_h^2, n_h^3, \ldots, n_h^{max}$$

but relaxing this requirement to

$$n_h^{min} \leq n_h \leq n_h^{max}$$

for computational purposes seldom renders the results unacceptable.

1.4 Selecting a mathematical programming method

The increasing capabilities of digital computing machinery has spurred remarkable advances in the theory and application of mathematical programming. A broad spectrum of problem formulations and accompanying optimization techniques has grown up. The new ideas have been applied to a wider circle of industrial problems, among them that of economic operation of electric power systems.

Several power system models have been proposed for the load dispatching task. Linear programming, nonlinear optimization, and dynamic programming methods have been used in the works referenced in squares (1,3) and (2,3) of Figure 1-1. None of these approaches has as yet emerged pre-eminent, and the search continues for an efficient and accurate method of calculating instantaneous economic schedules of generation. It is of interest at this point to outline some of the factors influencing the author's search for a new technique of load dispatching.

The primary goal was to be able to schedule generation on a national scale, dispatching the load to hundreds of units. Speed of solution was an essential requirement if this task were to be possible. Low solution times dictated simplicity in the system model and eliminated the complex variable network model from consideration. Furthermore, the economic interaction between real and reactive power schedules was expected to be relatively unimportant. There is not much flexibility remaining in the reactive schedule after the voltage control requirements have been met.

It was also observed that the input-output cost of generation curves are most accurately represented by one or more linear segments.(46) This fact in itself suggested the use of linear programming. However, an additional simplification that demanded attention was the important

fact that but a single commodity, real power, was involved in the optimization.

General linear programming formulations typically undertake the optimization of a system of equations whose variables represent many different commodities or types of activity. Transportation models are specialized linear programming formulations where a single commodity is to be transported from a set of sources to a set of sinks. A transportation problem can be solved much more rapidly by a transportation algorithm than by a general linear programming procedure.

The economic scheduling problem does not fall naturally into formulation as a transportation problem because the transmission network introduces an additional degree of complexity not easily accounted for.

Following these observations, an intermediate approach was sought, lying in complexity between the transportation problem and the general linear programming problem. Such a formulation was found: the static, single-commodity network flow model.

1.5 The network flow method

In 1961, D.R. Fulkerson presented an algorithm for the solution of minimal-cost network flow problems which he called the 'Out-of-Kilter Method'. (See Appendix 1, reference 3) It is the most general of a family of

algorithms that solve static, single-commodity network flow problems. Internally, the procedure operates in quite a different manner than a general linear programming routine. A node labelling technique is used in finding the optimal solution, and the only arithmetic operations carried out in doing so are addition and subtraction. As a consequence, this specialized solution method tailored to the network flow formulation can realize important savings in computing effort.

Appendix 1 contains a complete description of Fulkerson's algorithm, as well as definitions of network flow terms. In the next chapter, Figure 2-5 shows the network flow model of a small power system.

It is a relatively straightforward task to construct a network flow model for the economic scheduling problem. A second advantage of the approach is that the network flow algorithm has a well-defined termination property, as does the general linear programming routine. The solution procedure indicates that an optimal feasible solution has been reached; or that no feasible solution to the problem exists. Thirdly, the network flow solution will always be a global optimum solution as a result of the network flow formulation being a linear programming special case. In contrast, a nonlinear model may have its solution method fail to converge, or converge to an uneconomic local optimum.

1.6 Contribution of the thesis

A new method of economic scheduling with important advantages is proposed and demonstrated in this work. There is a direct equivalence between the economic scheduling problem formulation and its network flow model. Thus we have achieved a mathematical programming formulation and solution, without recourse to heuristic techniques as implemented in an informal digital computer program. Consequently, the solution technique is well-behaved, guaranteeing a global optimal solution where one exists.

Choosing the network flow method was the result of a deliberate policy to build an economic scheduling model of minimum structure consistent with obtaining accurate real power schedules. With this choice, a good trade-off was also achieved between complexity of power system representation and computer solution time. Hence the goal of being able to calculate economic schedules for very large power systems could be met.

The expectations for the network flow approach were tested in an extensive series of computational experiments. Chapter 3 describes checks that were made on the basic accuracy and speed of the method. Chapter 4 examines some of the errors and assumptions inherent in economic scheduling models, and shows that the coarse structure of the network flow model is not inconsistent with these.

Finally, the conclusions of Chapter 5 draw together the results of the previous chapters, and are a confirmation of the validity of this new approach to the economic scheduling of electric power generation.

Chapter 2 A NETWORK FLOW FORMULATION OF THE ECONOMIC SCHEDULING PROBLEM

Having specified the nature of the power system economic scheduling problem in the previous chapter, the next step is to express the same problem in mathematical programming terms. This chapter develops the network flow formulation in detail; at the same time relating the model to the physical elements of the power system.

Both the formulation and its solution make use of a number of elementary ideas from graph theory and the theory of network flows. The greater part of these are presented in Appendix 1, sub-section 3.1, the remaining ideas being introduced here as required. It is fortunate that the network flow model, because of its simplicity, may be readily understood and applied. However, its technique of solution, the 'Out-of-Kilter Method' is a complicated one. Sub-section 3.4 of Appendix 1 describes the Out-of-Kilter algorithm.

2.1 Branch model

A power system transmission network connects the various busses of the system, allowing the transportation of electric power between localities. The individual series branch connecting two busses may be a transmission line, a cable, a transformer, or a series reactor. Although these equipment types are dissimilar, each with their own properties, they all can be modelled in the same way for

the purpose of economic scheduling. The series branch network flow model is created for each branch independently, without reference to the other branches of the network.

In modelling the branch, the foremost concern is to correctly cost the transport of real power over that branch. In addition, we would like to place a limit on the branch power flow.

Figure 2-1 pictures the general series branch h, showing a real power flow of p_h MW flowing from bus i to bus j. Although there is no physical mechanism hard-limiting the flow of power in a branch, a number of practical considerations lead power system operators to impose limits on the permissable flow. Some of these are as follows.

Transmission line steady-current thermal limits, expressed in Amperes or MVA, as well as transformer and reactor MVA ratings may not be exceeded. A network security line outage study may specify real power limits for a branch. Similarly, stability studies may place a restriction on the angular displacement of two busses connected by a long transmission line.

The power system voltage is usually regulated to within quite narrow limits. Under such constant-voltage operation, power flow limits given in MVA, degrees, or Amperes can be expressed as a limit on the real power flow, in MW. The error of approximation in doing so is inconsequential when related imponderables such as security and stability

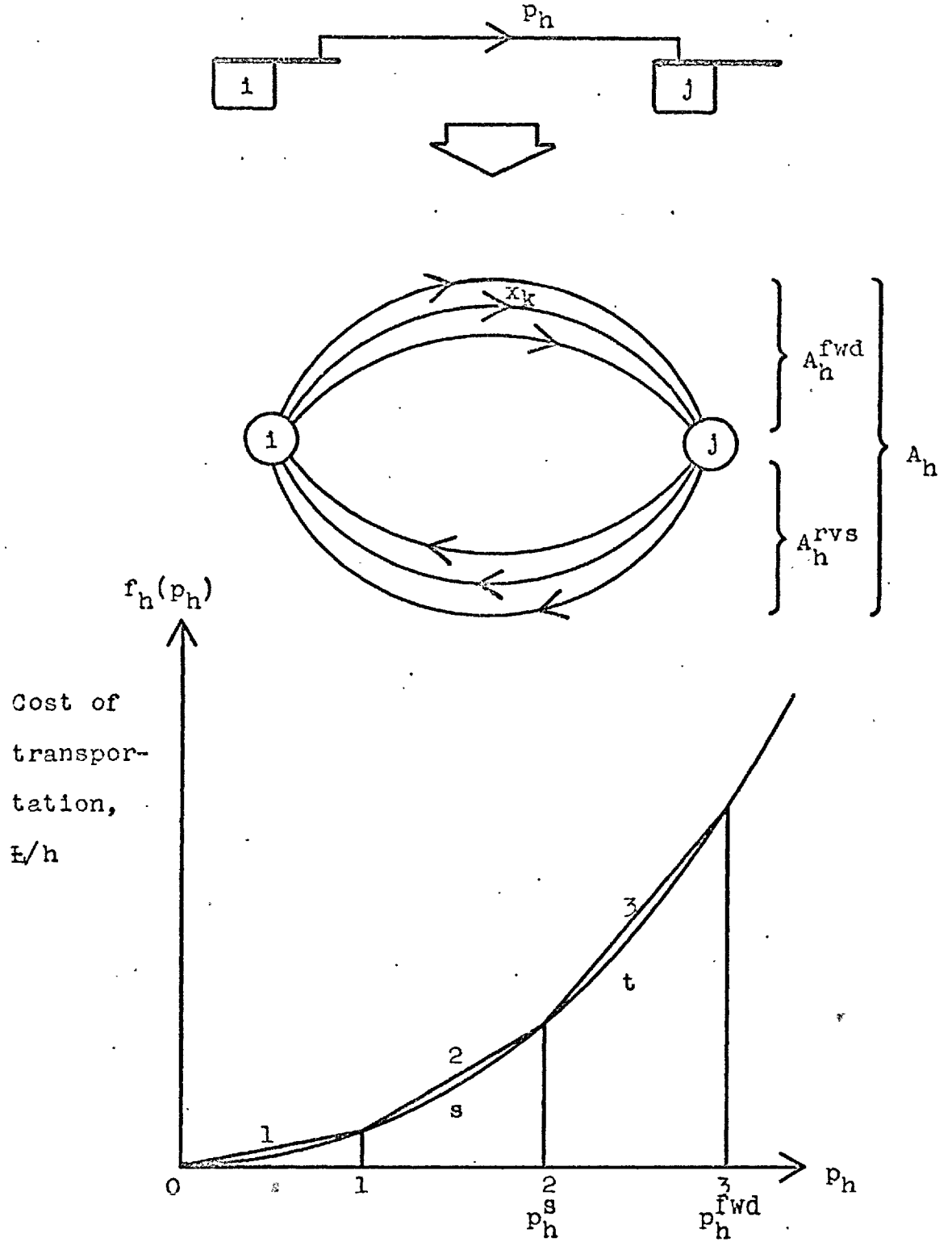


Figure 2-1 Branch model

definitions, wind and air temperature are considered. Moreover, the coarse structure of the network flow transmission system model makes fine attention to this detail superfluous.

Taking the most restrictive of the branch limits as the operative one, a set of inequalities may be written, one for each branch in the system:

$$p_h^{rvs} \leq p_h \leq p_h^{fwd}$$

Since the power flow may occur in the reverse direction, taking a negative value, p_h^{rvs} is a negative quantity. p_h^{rvs} may differ in magnitude from p_h^{fwd} if, for example, it happened to be a security limit while p_h^{fwd} was a thermal limit.

The cost of using branch h to transport power is the cost of the power losses in the branch. To start, an expression for the power lost as a function of the real power flow is required.

The quadratic loss function for the general branch h can be written

$$\ell_h = a_h p_h^2$$

where ℓ_h is the MW power loss in transmitting p_h MW. For each branch, the three-phase full-load power loss is either tabulated, calculable from standards, or can be calculated from the branch parameters. Substitution into the quadratic gives a value for a_h.

Let λ^{avg} be the average incremental cost of generation

in £/MWh. In this formulation, it is a single constant. In application, λ^{avg} can be adjusted for the total demand level, or assigned different values in different parts of the network, to reflect local economics of power generation.

If f_h is the branch cost of transport in £/h, then

$$f_h = \lambda^{avg} \ell_h = \lambda^{avg} a_h p_h^2$$

So in each branch, there is a quadratic cost function associated with the flow of real power through the branch.

Now the network flow model of branch h may be constructed. The branch quadratic transportation cost function can be handled by a convex piecewise linear approximation similar to that described on page 155 of reference 4 in Appendix 1. Figure 2-1 illustrates how the branch power flow is modelled by the sum of arc flows:

$$p_h = \sum_{k \in A_h^{fwd}} x_k - \sum_{k \in A_h^{rvs}} x_k$$

A_h^{fwd} is the set of arcs with orientation (i, j) modelling the forward flow of power in branch h. Three arcs are shown in the figure, but any number may be chosen. Since arc flows are non-negative variables, provision must be made for branch power flows reversed from the reference direction. This is done by including in the branch model a set of arcs A_h^{rvs} with orientation (j,i) symmetric to the forward set A_h^{fwd}.

To allow a zero flow, the lower bounds on all arcs A_h,

of the branch model must be zero

$$\ell_k = 0 \qquad k \varepsilon A_h$$

The quadratic cost function of Figure 2-1, and the discussion below are with respect to the forward flow alone, as the reverse flow model is identical.

Each arc, k, models a segment, s, of the piecewise linear approximate cost curve. Let the upper breakpoint of the segment s be p_h^s. Then

$$u_k = p_h^s - p_h^{s-1}$$

For segment 1, $p_h^0 = 0$, and $u_k = p_h^1$. For the top segment, t, $p_h^t = p_h^{fwd}$, and $u_k = p_h^{fwd} - p_h^{t-1}$. There is no difficulty in setting p_h^{fwd} and the breakpoints to integer values in megawatts. Then, the arc upper bounds will take on integer values, to satisfy that requirement of the network flow formulation.

The arc cost, c_k, is a scaled value of the slope of the corresponding segment, s:

$$c_k = \frac{f_h(p_h^s) - f_h(p_h^{s-1})}{p_h^s - p_h^{s-1}} \cdot s_f$$

Without multiplication by the scale factor s_f the arc costs would be decimal fractions, and they must have integer values for the network flow algorithm.

The multiple-arc network flow model of branch h is completed by the addition of the reverse arcs. It remains to check the validity of the branch model.

Choosing a scale factor $s_f = 10^6$ ensures that none of the arc costs are zero. In the power systems studied, even the lowest segment on a low-cost branch such as a transformer has a corresponding arc cost of a two, or three-digit integer. This confines the arc flows to the forward set A_h^{fwd}, or the reverse set A_h^{rvs}, the flows in the opposing set being zero. That property follows from the minimizing nature of the solution algorithm. For, suppose the converse were the case and flows were set up in two opposing arcs. But the two flows can always be replaced by a single flow equal to their difference and costing less to maintain. Thus only one set of flows can be active in a given solution, and we can confine our attention to the forward, or to the reverse set of arcs in the branch model.

Suppose a real power flow, p_h, had a value placing it in segment s. Then, because the solution algorithm minimizes the cost of arc flow, it will bring up the flow in the arc corresponding to segment 1 first, as that arc has the lowest cost. With that arc flow on its upper bound, the flow in the arc corresponding to segment 2 will be activated, and so on in numerical sequence, since the arc costs are monatonically increasing and form a convex function. In the end, the flows in the arcs corresponding to segments 1, 2, ..., s-1 will be established at their upper bounds, with the flow in the arc corresponding to segment s making up the final difference.

With the correctness of the network flow branch model verified, the last step is to show how the branch power flow is reconstructed from the arc flows. Rather than evaluating the expression for p_h previously set down, we can use the properties just described to indicate a computationally more efficient, yet equivalent method:

$$p_h = \pm \sum_{k \in A_h^{act}} x_k$$

A_h^{act} is the set of active arcs. More simply, the sum is taken over only the active arcs, the remainder being ignored, and is given a sign appropriate to the direction of flow.

2.2 Generator model

Generating sets impress power on the network for transmission to the demand busses. More than one set may be connected to any given generating bus. Each set is modelled individually, having limits and a cost placed on its power production.

The performance chart, or capability diagram of a typical generator, m, is drawn in Figure 2-2, with real power plotted on the vertical axis. Net real power, in megawatts sent out to the system, g_m, is the quantity to be represented in the model. It is the power output at the stator terminals minus the power required by the auxilliary

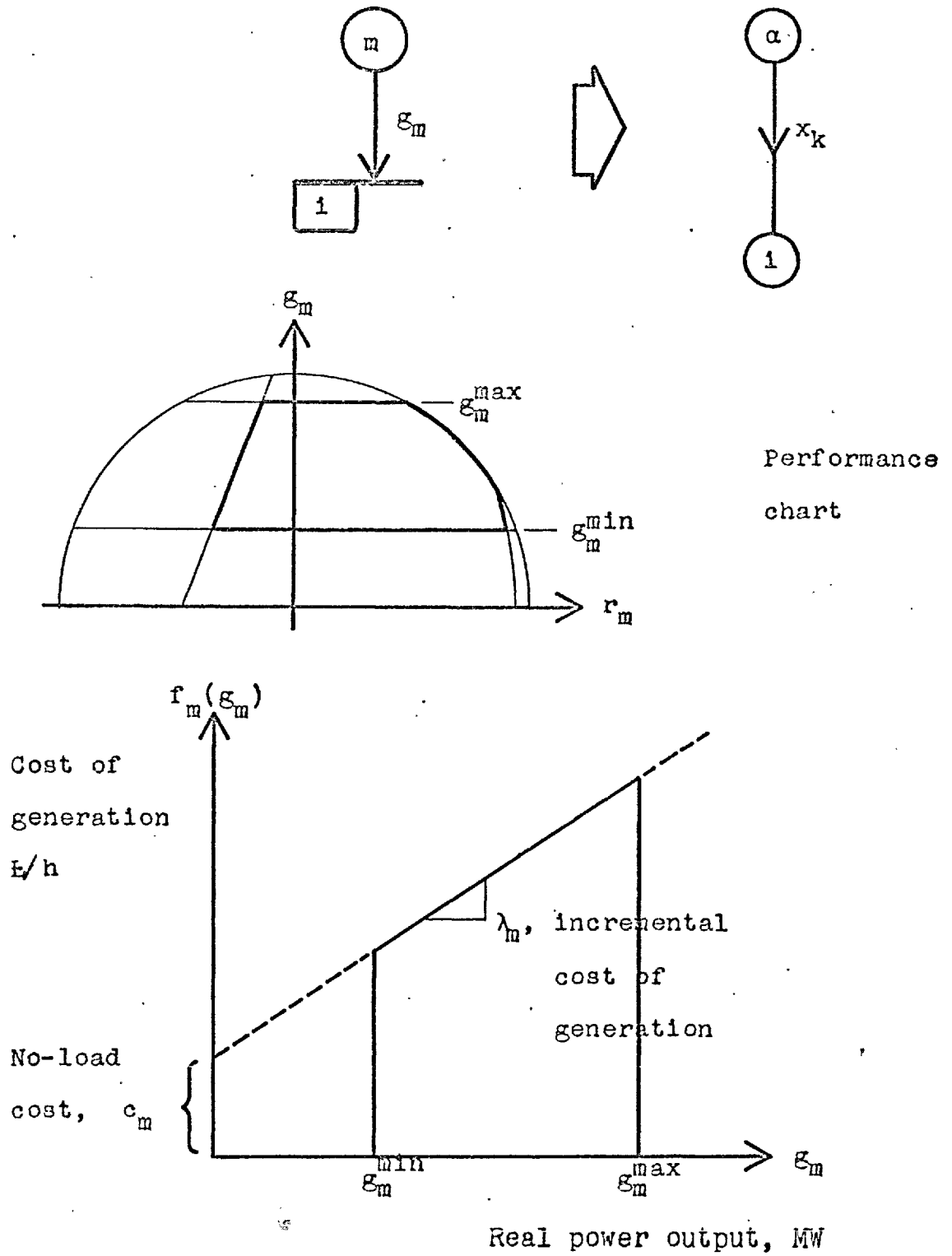


Figure 2-2 Generator model, Britain

equipment. g_m is limited by the designed capacity of the boiler/turbine combination to produce real power: g_m^{max}. In the model, the assumption is made that the system will be operated so that there are no important interactions between real and reactive power generation. That is, the alternator MVA limit, or any other limit, will not be active on any set to reduce its maximum real power generation.

Besides a maximum power capability, each generating unit has a lower limit, g_m^{min}, on its power output. This can be dictated, for example, by the inability of the boiler to run on low load, or by overheating in the steam turbine low pressure stage.

Thus the range of operation of each generator can be expressed as:

$$g_m^{min} \leq g_m \leq g_m^{max}$$

All the modern British turbogenerators are designed to reach maximum efficiency at full load, and their generation cost curves are straight lines. So for economic scheduling purposes, each generator has a single incremental cost of generation, λ_m, over its range of operation. The cost of generation is

$$f_m = \lambda_m g_m + c_m$$

£/h. c_m is the fixed, no-load cost of generation. The Merit Order 'B' Table is a list of sets ordered by their λ_m with the most efficient ones coming at the top of the

table, and having low values of incremental cost.

Recall that in the economic scheduling problem, sets are pre-selected and are not free to come on and off the system. Then the sum of the no-load costs, c_m, will contribute a fixed additive constant to the total system generating cost. The optimal schedule is independent of this total no-load constant cost. Therefore, only the variable part

$$f_m = \lambda_m g_m$$

of the cost of generation need be included in the formulation.

The network flow model of the British single-cost set takes a particularly simple form, that of a single arc, $k = (\alpha, i)$. Terminal node i corresponds to the power system generator bus i, while start node α is the network flow model source node supplying all the generator models. Rounding generator power limits to the nearest megawatt gives the arc flow bounds:

$$\ell_k = g_m^{min}, \qquad u_k = g_m^{max}$$

λ_m, the generator incremental cost, has a significant fractional part, and so must be scaled up. The cost scaling factor, s_f, is the same value as used for the branch cost.

$$c_k = \lambda_m s_f$$

After the network flow solution has been calculated, the generator output power is

$$g_m = x_k$$

In the United States, the design of generating units is such that their cost curves resemble the one shown in Figure 2-3.(12,25) The turbines have more than one steam admission valve, and this allows them to run efficiently at less than full load. A partly-opened valve gives rise to throttling losses. The effect of the valve throttling losses is to cause the cost curve to 'loop' upwards inefficiently at the points where a new valve is opened.

The classical American scheduling method (32) approximates the generator cost curve by a quadratic function

$$f_m = a_m g_m^2 + b_m g_m + c_m$$

Dropping the constant term as before leaves the variable portion of the function as

$$f_m = a_m g_m^2 + b_m g_m$$

computational examples of which are presented in Appendix 4.

A better approximation to the multi-loop cost curve is a segmented straight line function, with breakpoints corresponding to the valve loading points.(46) With this representation the incremental cost of generation, instead of varying smoothly, is stepped, taking on discrete values. The piecewise linear approximation is superior to the quadratic approximation because it tends to load most of the generators up to the efficient setting just below the valve opening point. In this way only the units required for regulation will incur valve throttling losses.

With these considerations in mind, the network flow

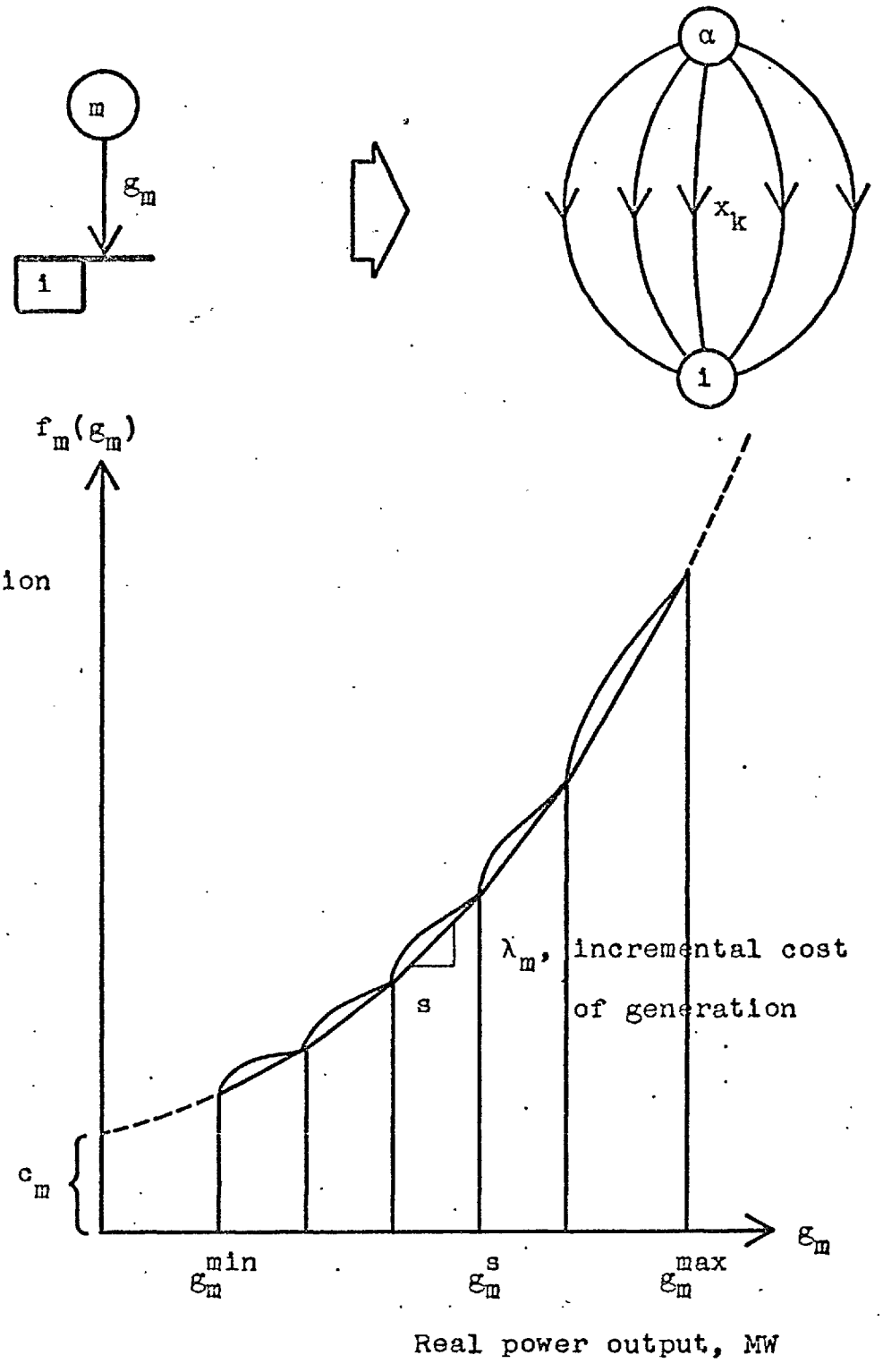


Figure 2-3 Generator model, United States

model for a multiple-valve generating unit suggests itself immediately: model each segment of the cost curve with an arc. A unit with five segments in its cost curve for example, is completely described by a group of five parallel arcs, all emanating from the source node α, and terminating on the generating bus node i.

The arc modelling the bottom segment has bounds

$$\ell_k = g_m^{min}, \qquad u_k = g_m^1$$

where g_m^1 is the first breakpoint. Subsequent segments are modelled by arcs with bounds

$$\ell_k = 0, \qquad u_k = g_m^s - g_m^{s-1} \quad \text{segment s}$$

For the top segment, g_m^s is g_m^{max}.

The slope of segment s is the incremental cost of generation in that range. Multiplying by the scale factor s_f gives the cost for the arc modelling segment s.

$$c_k = \frac{f_m(g_m^s) - f_m(g_m^{s-1})}{g_m^s - g_m^{s-1}} \cdot s_f = \lambda_m^s s_f$$

This completes the construction of the generator model; the next step is to demonstrate its validity.

If the segment incremental costs are monatonically increasing, a condition that obtains in practice, the network flow model is valid. The first arc maintains a flow at least as great as the minimum generation g_m^{min}. As the generation increases, the arc flows become active in sequence in the same manner as they do for the branch model.

Reconstructing the generator power output from the

arc flows

$$g_m = \sum_{k \epsilon A_m} x_k$$

where A_m is the set of arcs in the generator model. Again, the inactive arcs may be passed over when forming the sum.

2.3 Modelling demands and imports

At bus i, the power transfers to areas outside the system, and to lower voltage networks not included in the model can be netted to give the demand in megawatts, d_i. For a predictive economic scheduling study, d_i is the value resulting from a load forecasting program.

The network flow model of a positive demand is simply a single arc connecting bus i to a node ω, called the sink node. As its name implies, ω is added to the network flow model to remove flow from the network. It does not correspond to a bus in the power system. The arc lower and upper bounds are both set equal to the nearest integer value of d_i:

$$\ell_k = u_k = d_i$$

This fixes the arc flow x_k at a single value. There is no point in costing the flow, so that

$$c_k = 0$$

If the demand is negative, indicating a net import, an arc of opposite orientation (ω, i) is used to model the power flow. ℓ_k and u_k are then set equal to the absolute

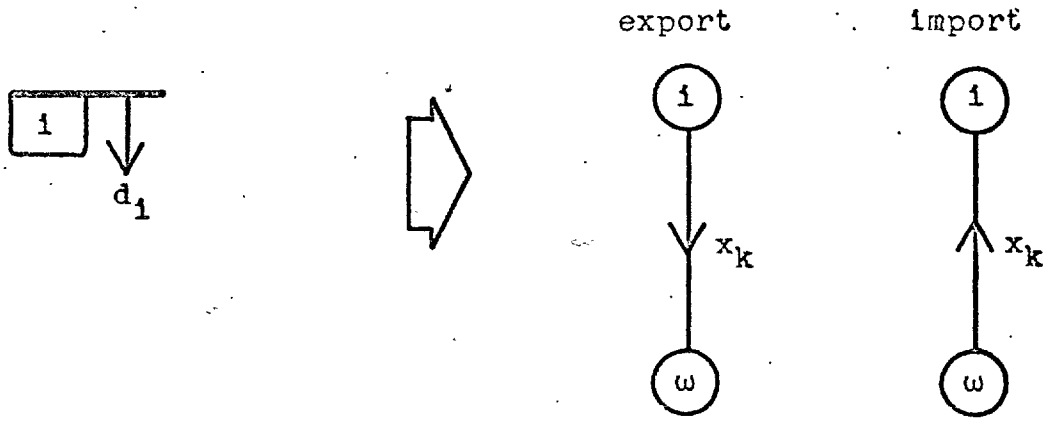


Figure 2-4 Demand model

value of d_i.

After the flow solution has been found for the network, the sign of d_i can be retrieved by examining the orientation of the arc k that models it.

$d_i = \pm x_k$

The demand model is drawn in Figure 2-4.

2.4 The complete model

2.4.1 The return arc

Flow is conserved at each node of the network, a fact expressed by the conservation equations, section 3.3, Appendix 1. The return arc, $k = (\omega,\alpha)$ satisfies this requirement at the source and sink nodes by the simple expedient of joining them. If we choose

$$\ell_k = 0, \qquad u_k \geq \sum_{m \in G_s} g_m^{max}, \qquad c_k = 0$$

for the return arc, it will not affect the solution. The precise value selected for u_k is unimportant.

2.4.2 Power balance equations

At each bus in the system there exists a balance between the power entering and the power leaving. That is, the power generated must equal the power demand plus the power transported elsewhere. Algebraically this requirement is expressed by a set of equations, one for each bus in the system:

$$\sum_{m \in G_i} g_m = d_i + \sum_{h \in B_i} p_h \qquad i = 1,2,\ldots,n_{bus}$$

G_i is the set of generators feeding bus i, and B_i is the set of branches connected to bus i. We must check that the network flow model of the system satisfies the power balance equations.

Substituting the network flow equivalent expressions for the three types of model element:

$$\sum_{m \in G_i} \sum_{k \in A_m} x_k = \pm x_k + \sum_{h \in B_i} \left(\sum_{k \in A_h^{fwd}} x_k - \sum_{k \in A_h^{rvs}} x_k \right)$$

$$i = 1,2,\ldots,n_{bus}$$

These arc flows can be collected into two sets, those impinging on the node representing bus i, A_i^-, and those radiating from it, A_i^+. For example if a demand term were an import, it would bear a minus sign in an above equation, and would be included in the set A_i^-. Rearranging,

$$\sum_{k \in A_i^-} x_k = \sum_{k \in A_i^+} x_k \qquad i = 1,2,\ldots,n_{bus}$$

The equations thus written are seen to be but a subset of the network flow conservation equations described in Appendix 1. Since the solution algorithm maintains conservation of flow at all nodes of the model, the power balance equations are therefore automatically satisfied.

2.4.3 Cost form

$$z = \sum_{k \in A} c_k x_k$$

is the linear cost form minimized by the network flow algorithm. The demand arcs and the return arc were given cost coefficients of zero and so may be omitted from the cost sum. Many arcs in the branch and generator models will be inactive, contributing nothing to the cost of flow. Rewriting z as the sum of terms making a positive contribution to the flow cost:

$$z = \sum_{h \in B_s} \sum_{k \in A_h^{act}} c_k x_k + \sum_{m \in G_s} \sum_{k \in A_m^{act}} c_k x_k$$

z is expressed in two sums, the first being the power transportation cost, and the second the cost of power generation.

The relative magnitude of the two parts of z is such that generation costs dominate the solution, just as in operational practice. However, the multiplier λ^{avg} applied in costing the losses, results in transportation cost coefficients c_k of magnitude sufficient to influence the solution to the proper degree. When it is more economical to transport power than generate it locally, the cost coefficient values will permit the solution algorithm to arrive at that result.

2.4.4 A small example

The five-bus test system GHKS5 of Appendix 4 serves here to illustrate the overall structure of a power system network flow model. Models for larger systems follow the

same pattern. Branch cost curves are modelled with three segments and generator cost curves with five. The network diagram is drawn to overlay the power system diagram of Figure 3-2. Although it appears cluttered and intricate, its tabular realization in a digital computer is not. Note the connection of the source and sink nodes by the return arc. As there is no demand at bus 3, no demand arc is required to be in the model at node 3.

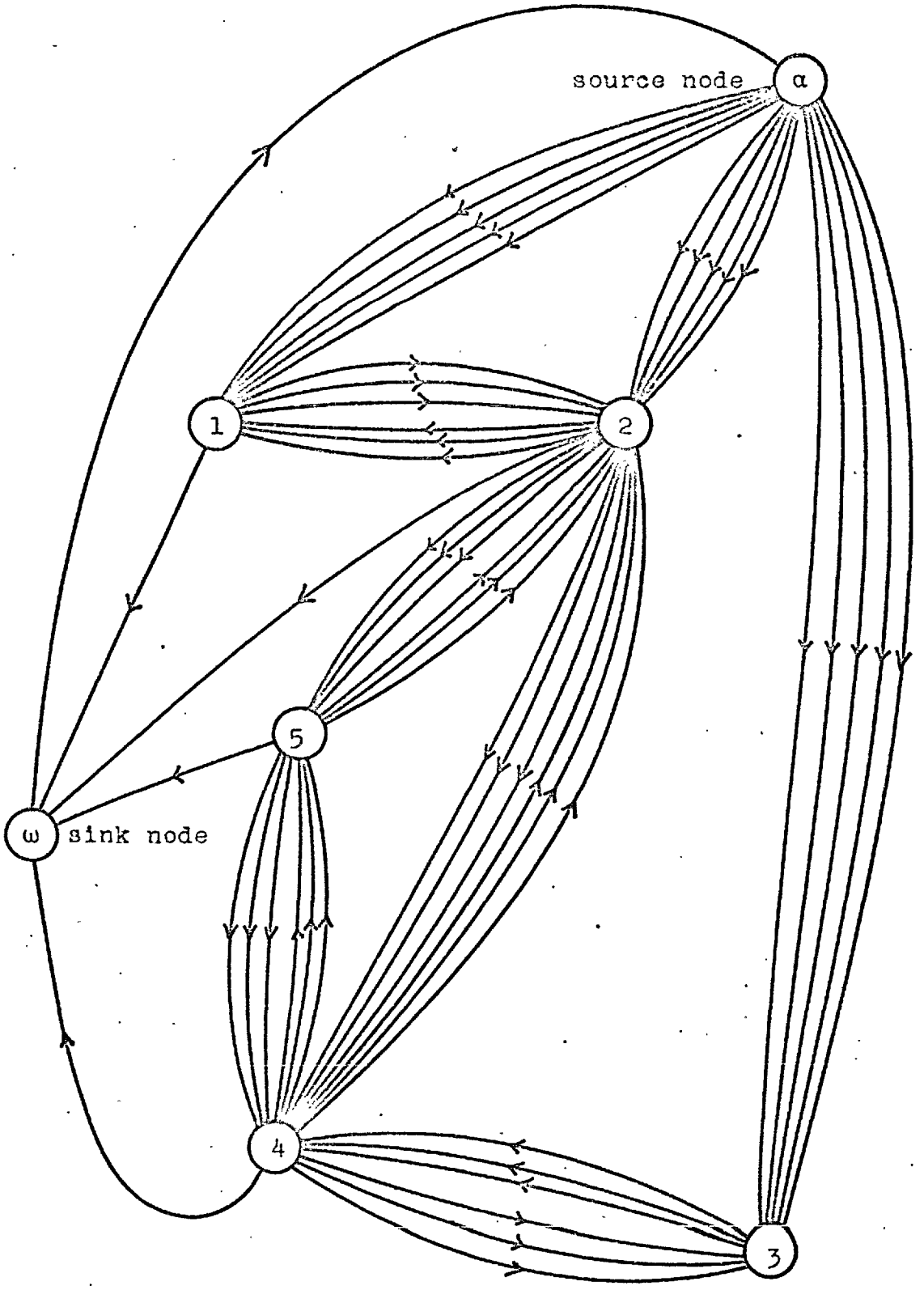


Figure 2-5 Network flow model of GHKS5

Chapter 3 COMPUTATIONAL EXPERIMENTS IN ECONOMIC SCHEDULING

3.1 Purpose and method

The network flow approach to economic scheduling was chosen because it held the promise of low solution times for large power systems. Demonstration of this expectation as a fact required a series of computational exercises on test systems of various sizes. Section 3.2 gives a brief summary of these systems.

The first concern was to verify the correctness of the problem formulation and of the computer programs. As economic schedules for the smaller examples were computed by the network flow method, they were manually checked for the slightest deviation from optimality, constraint satisfaction, or flow conservation. No such anomalies were found. The computer programs used in the experiments are introduced and discussed in section 3.3, and their documentation is provided in the Appendicies 1, 2, and 3.

One of the chief objectives of the experiments was to record the solution time for a large test system. A timing subroutine working from the computer's internal clock calculated running times to a fraction of a second. The largest system, SG149, was used for this test, the result of which is included in Table 3-4.

A more difficult undertaking was the investigation of the accuracy of the economic schedules produced by the network flow method. Complete complex variable models of

the five smallest of the seven test systems were constructed and used in these checks. It was assumed that conclusions reached for the representative, medium-sized test systems MB23, KS28, and BCA56 would be valid for the two largest systems, GR142, and SG149. Checks on the schedule accuracies for GR142 and SG149 were not attempted.

To solve the complex variable model, a conversational mode load flow program was written. This general-purpose program was employed in finding optimal economic schedules, in checking network flow economic schedules, and in the investigations of Chapter 4. It is described in Section 3.3.2, and documented fully in Appendix 2.

3.2 Test power systems summary

Four of the seven test systems were abstracted from CEGB data, and three were taken from the American literature on load dispatching. The following table permits comparison of the test system sizes.

Table 3-1 System sizes

System	Branches	Generators	Demands	Busses
GHKS5	5	3	4	5
SQ19	24	4	10	19
KS28	35	10	24	28
MB23	30	24	23	23
BCA56	84	43	45	56
GR142	175	60	79	142
SG149	185	177	96	149

All seven systems are described in some detail in Appendix 4. That appendix also includes data for the first four systems; comprising complete branch data, generator data, and real power demand patterns. The remaining data, defining the complex model operating conditions for these four systems, may be found with the optimal solutions in section 3.4. In the tables and figures to follow, spb stands for segments per branch, and spg denotes segments per generator.

3.3 Computer programs description

3.3.1 MULTAR, Multiple arc out-of-kilter network flow routine

With many possible areas of application in industrial problems, Fulkerson's network flow algorithm is a quite general one. Consequently, this valuable characteristic was retained in its implementation, the MULTAR routine being written as a general-purpose mathematical programming module, rather than specialized for the particular application in hand. Appendix 1 is a complete documentation of MULTAR.

RS OKF1 (See Appendix 1, section 16) is a previous implementation of the Out-of-Kilter Method. It is an intricate, optimized program, approximately twice as fast in execution as MULTAR. However, it could not be converted to run under the IBSYS monitor of the IBM7094, and moreover,

would have been difficult to imbed in the power system economic scheduling program. The trials using RS OKF1 were valuable nevertheless, since a reliable indication of the time improvement to be gained through optimized computer programming was obtained.

3.3.2 CMLFP, Conversational mode load flow program

This program was written for the small PDP-9 digital computer, allowing close interaction between the operator and the power system model via the KSR33 teletype. In this way, the advantages of both the network analyzer and the digital computer were realized in one instrument. The program was written entirely in standard FORTRAN.

The electric power system complex variable model assembled by CMLFP defines the relation between the steady state system complex voltages and powers. CMLFP uses the well-known method of Glimn and Stagg (Reference 4, Appendix 2) to solve the network equilibrium equations. It has no automatic features, relying on the operator to adjust system conditions for constraint satisfaction or optimization. A version of CMLFP was prepared for the IBM7094 in order to produce results on diagrams, and to solve the larger test system, BCA56. The branch data format of CMLFP is compatible with NFESP, the network flow economic scheduling program.

3.3.3 NFESP, Network flow economic scheduling program

NFESP was the main instrument by which the ideas of the thesis were checked. In outline, it constructs the network flow model, calls upon MULTAR to solve it, and tabulates the solution in power system terms. The program is described in Appendix 3. Full documentation is not given, firstly because of the program size (951 text lines) and secondly because it is experimental and not likely to be of direct use in its present form. The program sections that are reproduced are key ones: they illustrate the implementation of the theory.

3.4 Optimal economic dispatch for complex power system models

Determination of the optimal operating point for a complex variable power system model is in general a difficult exercise. The real power generation schedule and the bus voltage magnitude profile must both be adjusted to give the minimum cost of generation. In doing so, none of the constraints 3 to 6 in section 1.3 may be violated. Recent research (14,15,16,50) on this problem has produced computer programs capable of finding an optimal point automatically. Lacking such a program, the author resorted to manual searches for optimal operating points, making use of CMLFP.

The search is begun by setting in a near-optimal solution, either estimated in advance or obtained from a previous

result in the literature. In some cases the voltage profile available in research papers was very near optimal, and thus could be used directly. For the two British systems, a careful adjustment of the generator bus voltages and transformer taps was required for constraint satisfaction. Then the search is carried on by swinging generation in pairs to obtain pairwise optima.

One of the busses is designated the slack bus. Raising the power output at the other bus will cause a nearly equal drop in generation at the slack bus. The locus of the system operating point projected on the generation space of these two busses is a gently curving line. Three or four decreasing 'swings' are usually sufficient to locate the minimum-cost point on the locus. A second pair, one or both members of which may have been swung before, is selected and the process repeated. The overall cycle is iterative, readjusting all possible pairs until no change in cost can be observed. A check on the optimality of the voltage profile follows the optimization of the generation pattern.

The procedure, although primitive and somewhat tedious, is easy to operate manually. Its last cycle forms a test for optimality. Up until then the operator can omit pairs he considers to have little effect on the minimization. An additional help is received when some of the generation busses have active real power constraints, since these also

can be omitted until the final check for optimality.

Table 3-2 summarizes the optimal solutions for five of the test systems. The listed cost of generation does not include the no-load cost. In the case of the American test systems, the quadratic cost curves given in Appendix 4 were used. Following Table 3-2 are included listings of optimal solutions for a single load level on each of the four smallest test systems.

3.5 Applying the network flow schedule to the complex model

Before the network flow economic schedules can be checked for accuracy against the complex variable model optimal solutions, they must be applied to the complex model itself. This is because in the network flow model the transmission losses are not included in the real power balance and the generation just equals the total load. When the network flow schedule is applied to the complex model, there will be a deficiency in generated power equal to the system losses. Using CMLFP, that amount of generation will automatically be picked up by the slack bus, resulting in off-optimal operation. This behaviour results from the method of solution of the complex model, and is not what would happen in practice on the actual power system. CMLFP must therefore be so handled as to reflect the performance of the system on being instructed by a network flow schedule;

not a difficult requirement.

For the British test systems MB23 and BCA56, generators operating 'in range' were brought up to meet the losses, and if this were not sufficient, the Merit Order 'B' Table was then followed.

In contrast, operational practice in the American test systems was simulated by increasing the level of generation on each unit in proportion to its size. The only generators not assisting in the maintenance of the real power balance were those already generating at capacity. Even those operating at their lower limits were required to pick up generation according to the size rule. Expressing this operational event algebraically, first let G_r be the set of regulating generators. Then the fraction of the losses supplied by generator m is

$$\frac{g_m^{max}}{\sum_{m \epsilon G_r} g_m^{max}}$$

In using CMLFP to simulate this action, there is a very small difference in the losses due to the slightly modified generation schedule. The newly-found losses are then re-apportioned by the same rule. Two or three applications of the procedure are sufficient to reach the desired simulated condition.

Table 3-2 Optimal economic schedules for complex power system models; a summary

Test system	Demand ÷ Σg^{max} %	Total net demand MW	Real power generated MW	Trans-mission losses MW	Cost of generatn $/h	Losses ÷demand %
United States						
GHKS5	67	266.0	271.9	5.9	962.49	2.2
	47	186.4	189.0	2.6	625.72	1.4
SQ19	68	310.7	316.5	5.8	451.39	1.9
	47	217.5	220.9	3.4	304.24	1.6
KS28	81	1730.0	1816.5	86.5	3579.37	5.0
	53	1138.5	1180.0	41.5	1700.11	3.6
Britain					£/h	
MB23	90	2643.0	2689.5	46.5	3795.71	1.8
	79	2300.0	2333.1	33.1	3158.89	1.4
BCA56	93	2118.0	---------not checked----------			
	80	1830.0	1858.9	28.9	4105.88	1.6
GR142	87	3279.0	----------no complex model---------			
SG149	87	16940.0	----------no complex model---------			

GHKS5 GLIMN, HABERMANN, KIRCHMAYER, STAGG 5 BUS SYSTEM
DOPAZO ET AL OPTIMAL SOLUTION FROM THEIR PAPER

BUS	PL	QL	PG	QG	V	THETA
1	86.00	20.00	170.40	9.23	1.050	4.734
2	80.00	10.00	46.31	31.42	1.040	0.000
3	0.00	0.00	55.20	0.15	1.050	8.255
4	70.00	3.00	0.00	13.90	1.016	-3.766
5	30.00	12.00	0.00	11.50	1.015	-3.917

LN	SB	EB	PS	PR	QS	QR
1	1	2	84.4	82.4	-10.8	-17.5
2	2	5	28.2	27.6	2.2	0.3
3	2	4	20.5	20.1	1.6	0.3
4	4	5	2.4	2.4	0.2	0.2
5	3	4	55.2	52.3	0.2	-11.0

GHKS5 GLIMN, HABERMANN, KIRCHMAYER, STAGG 5 BUS SYSTEM
NETWORK FLOW SOLN WITH GENS INCR PROP TO GMAX

BUS	PL	QL	PG	QG	V	THETA
1	86.00	20.00	173.00	8.71	1.050	4.883
2	80.00	10.00	47.39	31.56	1.040	0.000
3	0.00	0.00	51.40	0.36	1.050	7.029
4	70.00	3.00	0.00	13.34	1.016	-4.140
5	30.00	12.00	0.00	11.67	1.015	-4.177

LN	SB	EB	PS	PR	QS	QR
1	1	2	87.0	84.9	-11.3	-18.5
2	2	5	29.9	29.3	1.9	-0.3
3	2	4	22.4	21.9	1.2	-0.4
4	4	5	0.7	0.7	0.7	0.7
5	3	4	51.4	48.9	0.4	-9.3

Figure 3-1 GHKS5, 266 MW, optimal solution and check solution for 5 spb, 5 spg network flow test

GHKS 5

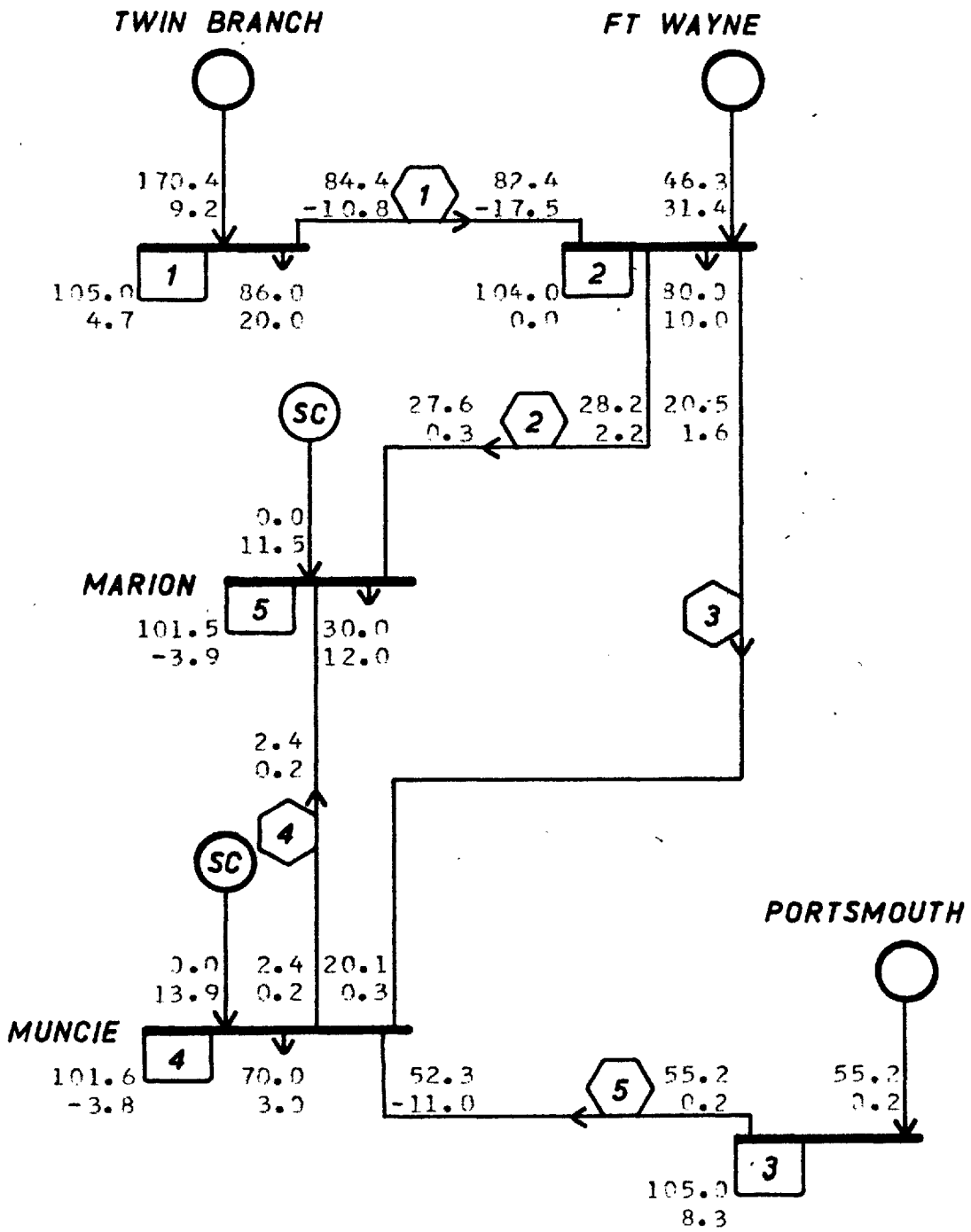


Figure 3-2 GHKS5, 266 MW, optimal solution

SQ19 . OPTIMAL SOLUTION TO THE COMPLEX MODEL
NEXT.

BR

BUS	PL	QL	PG	QG	V	THETA
1	12.70	9.20	0.00	0.00	0.991	1.327
2	42.50	18.50	0.00	0.00	0.990	1.518
3	0.00	0.00	0.00	0.00	0.997	0.477
4	26.20	11.40	0.00	0.00	0.990	-0.174
5	53.40	22.60	0.00	0.00	0.987	-0.493
6	0.00	0.00	0.00	0.00	0.994	-0.275
7	9.80	5.30	0.00	0.00	0.992	-0.772
8	15.70	10.10	0.00	0.00	0.987	-1.234
9	14.60	6.60	0.00	-0.00	0.985	-1.804
10	36.60	16.30	0.00	0.00	0.992	0.041
11	38.70	12.20	0.00	0.00	0.993	-0.113
12	60.50	22.20	0.00	0.00	0.995	-0.142
13	0.00	1.50	0.00	0.00	0.994	-0.268
14	0.00	10.60	0.00	0.00	0.995	0.581
15	0.00	1.80	0.00	0.00	0.988	-1.836
16	0.00	0.00	27.60	10.85	1.000	-0.086
17	0.00	0.00	25.70	0.11	1.000	0.591
18	0.00	0.00	222.00	2.99	1.000	10.862
19	0.00	0.00	41.24	71.69	1.000	0.000

NEXT.

PF

LN	SB	EB	PS	PR	QS	QR
1	2	1	6.6	6.6	-3.2	-3.2
2	2	3	27.0	26.8	-21.6	-22.2
3	18	2	77.6	76.1	2.2	-10.3
4	18	1	72.4	71.0	2.0	-9.9
5	18	10	72.1	70.5	1.4	-12.1
6	1	10	64.9	64.6	-18.0	-19.5
7	10	4	65.0	64.9	7.5	7.3
8	4	5	38.7	38.6	6.4	6.1
9	6	5	14.8	14.8	12.9	12.7
10	16	6	14.9	14.8	12.9	12.8
11	6	13	-0.0	-0.0	1.1	1.1
12	10	11	33.6	33.5	-37.4	-37.5
13	12	11	5.2	5.2	38.4	38.3
14	19	12	65.9	65.7	49.5	49.1
15	19	16	3.1	3.1	-1.1	-1.1
16	19	3	-27.8	-27.9	23.9	23.6
17	16	7	21.2	21.1	6.5	6.2
18	7	8	11.3	11.2	2.5	2.4
19	8	9	14.6	14.6	-0.1	-0.2
20	9	15	0.0	0.0	-5.4	-5.4
21	17	8	19.2	19.1	4.4	3.7
22	17	16	5.3	5.3	-1.0	-1.1
23	3	14	-1.1	-1.1	2.9	2.9
24	17	14	1.1	1.1	3.4	3.4

NEXT.

Figure 3-3 SQ19, 310.7 MW, optimal solution

KS28 KIRCHMAYER AND STAGG AEP 28 BUS SYSTEM

KS28 OPTIMAL SOLUTION D1 LOADS, NOMINAL VOLTAGE PROFILE

BUS	PL	QL	PG	QG	V	THETA
1	86.00	30.00	92.60	33.56	1.012	-23.267
2	51.00	25.00	150.00	14.20	1.002	4.836
3	0.00	0.00	92.00	16.44	1.001	3.120
4	81.00	38.80	177.42	40.55	1.040	0.000
5	5.10	3.10	294.00	-28.57	1.050	1.276
6	117.00	28.00	116.00	41.03	1.036	-7.620
7	84.00	10.30	47.00	29.19	1.028	-12.889
8	83.80	-6.00	82.00	21.56	1.040	-16.977
9	47.30	31.80	0.00	10.99	0.996	-23.418
10	17.90	15.20	0.00	14.34	0.990	-24.428
11	28.00	8.00	18.00	19.04	1.000	-8.785
12	0.00	0.00	111.00	-20.45	1.000	5.294
13	15.00	8.00	0.00	10.98	1.004	-13.799
14	25.10	15.00	0.00	15.67	1.008	-8.292
15	28.20	10.00	0.00	8.15	1.012	-11.635
16	53.40	20.00	0.00	0.00	1.016	-7.968
17	29.10	16.00	0.00	-1.76	1.030	-7.492
18	55.00	24.90	0.00	0.00	1.022	-16.602
19	32.00	15.00	0.00	20.62	1.000	-19.158
20	23.00	13.00	0.00	0.00	0.995	-6.077
21	17.00	8.00	0.00	0.00	0.992	-10.898
22	38.20	25.00	0.00	11.92	0.990	-23.592
23	16.90	2.90	0.00	0.00	0.990	-2.598
24	30.00	5.00	0.00	0.00	0.983	-1.184
25	127.00	86.00	0.00	52.02	0.987	-1.291
26	0.00	0.00	0.00	0.00	1.004	4.328
27	0.00	0.00	0.00	0.00	1.000	3.321
28	47.50	24.60	0.00	0.00	0.993	2.244

Figure 3-4 KS28, 1138.5 MW, optimal solution, bus quantit

KS28 KIRCHMAYER AND STAGG AEP 28 BUS SYSTEM
BRANCH POWER FLOWS

LN	SB	EB	PS	PR	QS	QR
1	1	9	6.6	6.5	13.8	13.6
2	9	10	6.7	6.7	0.3	0.2
3	9	22	1.3	1.3	1.3	1.3
4	22	10	11.3	11.2	-3.1	-3.2
5	19	9	49.8	48.8	-10.1	-13.7
6	13	19	84.3	81.8	-18.7	-26.3
7	14	13	91.6	89.0	-18.8	-27.4
8	4	14	87.7	84.9	5.7	-6.8
9	5	4	14.1	14.0	1.8	1.4
10	5	16	128.8	122.9	-0.4	-20.4
11	15	22	50.9	48.2	-2.5	-12.7
12	16	15	80.6	79.1	-16.3	-21.3
13	17	16	11.1	11.1	10.1	9.9
14	17	6	1.0	1.0	-16.6	-16.7
15	17	7	98.3	95.0	-27.9	-37.0
16	7	18	58.0	57.0	-9.2	-12.9
17	18	8	2.0	1.8	-30.0	-30.6
18	4	25	22.7	22.1	25.1	23.4
19	2	25	86.8	84.7	-2.3	-11.3
20	3	25	56.6	55.1	-7.6	-11.7
21	26	27	12.2	12.1	-3.4	-3.6
22	21	13	10.5	10.3	-6.2	-6.8
23	11	21	27.8	27.5	-2.3	-3.3
24	20	11	38.5	37.8	-16.9	-18.7
25	20	14	32.2	31.7	-23.1	-24.7
26	23	20	55.9	54.6	-23.7	-27.2
27	12	20	40.7	39.1	-3.1	-11.0
28	12	23	39.0	37.7	-3.9	-9.1
29	12	24	31.3	30.4	-1.1	-4.5
30	25	23	35.3	35.0	-17.0	-17.8
31	24	25	0.4	0.4	-5.1	-5.1
32	5	17	146.0	139.6	-13.2	-34.6
33	28	3	-47.5	-47.5	-24.6	-25.5
34	3	27	-12.1	-12.1	3.7	3.6
35	2	26	12.2	12.2	-3.3	-3.4

Figure 3-5 KS28, 1138.5 MW, optimal solution, branch power flows

MB23 OPTIMAL SOLUTION TO THE COMPLEX MODEL
NEXT.
ALL TRANSFORMER TAPPING RATIOS AT 1.03
NEXT.
BR

BUS	PL	QL	PG	QG	V	THETA
1	64.00	21.00	45.00	96.82	1.050	-1.056
2	101.00	34.00	105.45	103.82	1.050	0.000
3	0.00	0.00	0.00	0.00	1.041	-0.521
4	47.00	15.00	0.00	0.00	1.013	-0.299
5	51.00	17.00	-0.00	0.00	1.017	1.314
6	41.00	13.00	0.00	0.00	1.010	1.090
7	48.00	16.00	0.00	0.00	0.999	0.625
8	1.00	0.00	0.00	0.00	1.018	4.263
9	150.00	50.00	0.00	0.00	0.999	2.139
10	177.00	59.00	0.00	0.00	0.993	-1.014
11	130.00	43.00	210.00	120.33	1.050	0.473
12	6.00	2.00	0.00	0.00	1.026	8.002
13	-4.00	-1.00	0.00	0.00	1.040	10.676
14	480.00	160.00	498.00	235.34	1.050	9.412
15	201.00	66.00	0.00	0.00	1.000	10.286
16	132.00	43.00	0.00	0.00	1.012	18.430
17	344.00	114.00	0.00	0.00	1.000	16.558
18	104.00	34.00	0.00	0.00	1.001	15.757
19	376.00	124.00	0.00	-0.00	0.996	15.814
20	-100.00	-33.00	803.00	222.98	1.050	24.504
21	375.00	123.00	0.00	0.00	1.021	20.276
22	-210.00	-70.00	0.00	0.00	1.020	19.470
23	129.00	43.00	1028.00	219.94	1.050	23.222

NEXT.

LN	SB	EB	PS	PR	QS	QR
1	1	3	-9.8	-9.9	23.1	22.8
2	1	4	0.9	0.2	54.2	52.3
3	2	5	-10.6	-11.2	41.3	39.8
4	8	5	63.4	62.2	-21.4	-24.6
5	2	7	4.9	4.3	31.4	29.9
6	3	6	-9.9	-10.3	25.2	24.2
7	4	9	-46.8	-47.7	38.8	36.3
8	9	7	44.1	43.7	-15.0	-16.2
9	8	6	52.4	51.3	-11.2	-13.9
10	11	10	80.0	77.6	79.9	73.6
11	8	10	65.9	63.7	-3.8	-9.6
12	9	10	36.4	35.7	-8.2	-10.2
13	13	14	63.7	63.5	-36.3	-38.1
14	14	12	81.5	81.1	64.0	60.6
15	15	12	120.6	119.8	-93.0	-100.3
16	18	15	325.0	321.6	-16.7	-47.6
17	23	13	333.8	325.8	14.7	-57.5
18	16	17	426.8	424.9	107.5	92.3
19	17	18	80.9	80.7	-13.2	-14.3
20	19	18	3.5	3.4	-44.5	-44.8
21	20	19	385.6	379.5	126.0	62.8
22	22	18	347.5	344.8	73.6	49.9
23	20	21	517.4	512.7	150.0	108.4
24	21	22	137.7	137.5	-5.2	-7.1
25	23	16	565.2	558.7	195.2	142.0
26	12	8	65.9	65.9	-19.8	-24.6
27	13	8	116.8	116.8	-2.1	-15.2
28	12	9	129.0	129.0	2.0	-11.2
29	13	9	149.3	149.3	21.3	-1.1
30	1	2	-10.2	-10.2	0.1	-0.1

NEXT.

Figure 3-6 MB23, 2643 MW, optimal solution

3.6 Computational results

The experiments are summarized in Tables 3-4 and 3-5. Their purpose was to check the speed and accuracy of the method, as well as to investigate the speed-accuracy trade-off inherent in the network flow formulation.

An important feature of the network flow model is the facility with which the user may exchange speed for accuracy by increasing the number of arcs modelling a piece of plant. The optimal trade-off between these two parameters will depend on the power system, and upon the requirements of the study. Tables 3-4 and 3-5 are analyzed with this question in mind.

Seeking additional confirmation of the method, two demand levels were computed for all but the two largest systems. Problems of collecting and handling data, and the difficulty of finding optimal complex variable solutions prevented checks on the network flow schedules found for GR142 and SG149. Facsimile reproductions of the results of certain key and representative trials on the four smallest test systems are included in this section. Each test comprises a network flow model schedule and its corresponding complex variable model check solution.

3.6.1 British test systems

Modern British turbogenerators are modelled with a single cost segment. A few of the older 'range' type

sets remain in service however, and these are modelled with a two-segment cost curve. Thus, with the generator models fixed, it only remains to study the effect on the economic schedule of adding segments to the branch model.

Looking down the cost difference columns in Table 3-4, it can be seen that the savings in going from 1 to 2 or more segments are probably unaccountable in practice, as they are less than 0.04%.

The accuracy of the network flow method is also placed in evidence by Table 3-4. Note that the largest scheduling error is 0.065%, again a very small figure. CEGB practice of scheduling by merit order without regard to transmission costs is confirmed in these results.

Solution time is a function of model size (number of arcs) and load level. For a given system, there is a strong motive to keep the model size to a minimum at one segment per branch. Higher load levels require longer to solve since more arcs are initially out-of-kilter and require flow increases. The algorithm computing time appears to vary as the square, or fractionally higher exponent, of the power system size, measured in arcs.

The action of the network flow method in accounting for transmission costs is highlighted in the results listed for MB23. Incremental costs of generation are nearly equal at busses 2 and 11, and so there exists 'marginal' generation there. The situation is summarized in the following

table, showing how the two network flow solutions bracket the optimal one. The latter is at an intermediate point on the generator cost curves, while the former two solutions rest on breakpoints in the overall cost function at busses 2 and 11. λ_i is the incremental cost of generation at bus i.

Table 3-3 MB23 marginal busses

	Optimal solution		Network flow check solutions			
			Out-of-merit schedule, 1spb		Merit-order schedule, 2spb	
Bus	g_i	λ_i	g_i	λ_i	g_i	λ_i
2	105	2.20	137	2.20	85	2.20
11	210	2.17	179	2.14	231	2.19

In the single segment per branch model, the cost of transporting power is too high at the levels present in the problem, forcing out-of-merit running and slight cost errors. The two segment per branch model corrects this condition.

Table 3-4 Summary of computational results for the British test systems

Dem-	Network flow						Complex model check			
and	Model		Computing time				Tran	Genern	Cost	Cost
			Algorthm		Total		loss	cost	diff	diff
MW	spb	arcs	min	sec	min	sec	MW	£/h	£/h	%
MB23										
2643	5	348	0	11.88	0	32.27	47.0	3796.66	0.95	0.025
2643	2	168	0	3.08	0	22.25	47.0	3796.66	0.95	0.025
2643	1	108	0	1.58	0	20.42	47.0	3798.17	2.46	0.065
2300	5	348	0	8.70	0	30.32	33.0	3158.89	0.00	0.00
2300	2	168	0	2.60	0	21.73	33.0	3158.89	0.00	0.00
2300	1	108	0	1.33	0	20.17	33.0	3158.89	0.00	0.00
BCA56										
2118	5	942	2	4.68	2	45.35	-----not checked---------			
2118	2	438	0	31.77	1	8.77	-----not checked---------			
2118	1	270	0	12.70	0	48.53	-----not checked---------			
1830	5	942	1	15.92	1	56.57	28.9	4105.88	0.00	0.00
1830	2	438	0	21.45	0	58.40	28.9	4105.88	0.00	0.00
1830	1	270	0	8.60	0	44.40	28.9	4106.51	0.63	0.015
GR142										
3279	2	854	1	29.65	time trap		-----no complex models---			
3279	1	504	0	39.10	1	39.67				
SG149										
16940	2	1014	3	0.40	4	41.53				
16940	1	644	1	19.45	2	57.23				

LINE DATA AND POWER FLOWS

25 LINES

LIN NO.	STT BUS NO.	END BUS NO.	DES VOL KV	LIN LEN MI	INCRML COST L/MWHR	THML LIMT MVA	POWR FLOW MW
1	1	3	132	15	0.0385	100	0
2	1	4	132	18	0.0460	100	-23
3	2	5	132	24	0.0599	100	0
4	8	5	132	19	0.0480	100	51
5	2	7	132	42	0.1040	100	0
6	3	6	132	40	0.0974	100	0
7	4	9	132	18	0.0451	100	-70
8	9	7	132	13	0.0338	100	48
9	8	6	132	27	0.0662	100	41
10	11	10	132	29	0.0709	200	42
11	8	10	132	36	0.0887	100	68
12	9	10	132	36	0.0887	100	67
13	13	14	275	53	0.0389	620	0
14	14	12	275	53	0.0389	620	18
15	15	12	275	47	0.0341	620	174
16	18	15	275	44	0.0324	620	375
17	23	13	275	108	0.0785	620	306
18	16	17	275	12	0.0090	620	461
19	17	18	275	24	0.0173	620	117
20	19	18	275	18	0.0131	620	0
21	20	19	275	56	0.0408	620	376
22	22	18	275	29	0.0213	620	362
23	20	21	275	25	0.0181	620	527
24	21	22	275	16	0.0116	620	152
25	23	16	275	25	0.0182	620	593

TRANSFORMER AND SERIES REACTOR DATA AND POWER FLOWS

4 TRANSFORMERS 1 SERIES REACTORS

XFR NO.	STT BUS NO.	END BUS NO.	HV KV	LV KV	INCRML COST L/MWHR	THML RATG MVA	POWR FLOW MW
26	12	8	275	132	0.0070	155	6
27	13	8	275	132	0.0070	155	155
28	12	9	275	132	0.0067	180	180
29	13	9	275	132	0.0070	155	155
30	1	2	132	132	0.0031	90	4

DEMAND DATA AND CHECK

23 DEMANDS

BUS NO.	DEMD POWR MW
1	64
2	101
3	0
4	47
5	51
6	41
7	48
8	1
9	150
10	177
11	130
12	6
13	-4
14	480
15	201
16	132
17	344
18	104
19	376
20	-100
21	375
22	-210
23	129

2643 MW TOTAL NET DEMAND

Figure 3-7 MB23, 2643 MW, 1 spb network flow solution, branch power flows and demand data

GENERATOR DATA AND POWER OUTPUT

24 GENERATORS

BUS NO.	S T A	S E T	GEN PWR MW	MIN GEN MW	ECR MW	MAX GEN MW	INCR COST L/MH	2ND COST L/MH
1	B	1	15	15		61	3.22	
1	B	2	15	15		61	3.22	
1	B	3	15	15		61	3.22	
2	B	4	15	15		61	3.22	
2	B	5	52	30		61	2.20	
2	B	6	30	30		61	2.20	
11		1	43	43		57	2.16	
11		2	43	43		58	2.19	
11		3	43	43		58	2.17	
11		4	43	43		58	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	B	5	112	22		112	1.71	
20	C	10	334	135		334	1.42	
20	C	12	357	143		357	1.21	
23	B	6	112	22		112	1.67	
23	B	7	112	22		112	1.71	
23	B	8	112	22		112	1.67	
23	C	9	334	135		334	1.35	
23	C	11	358	143		358	1.15	

2643 MW TOTAL GENERATED POWER

MERIT-ORDER TABLE

24 GENERATORS

BUS NO.	S T A	S E T	GEN PWR MW	INCR COST L/MH	OPIN COST L/MH	M I N	E C R	M A X
14		1	83	0.85	0.85			U
14		2	83	0.85	0.85			U
14		3	83	0.85	0.85			U
14		4	83	0.85	0.85			U
14		5	83	0.85	0.85			U
14		6	83	0.85	0.85			U
23	C	11	358	1.15	1.15			U
20	C	12	357	1.21	1.21			U
23	C	9	334	1.35	1.35			U
20	C	10	334	1.42	1.42			U
23	B	6	112	1.67	1.67			U
23	B	8	112	1.67	1.67			U
23	B	7	112	1.71	1.71			U
20	B	5	112	1.71	1.71			U
11		4	43	2.14	2.14	L		
11		1	43	2.16	2.16	L		
11		3	43	2.17	2.17	L		
11		2	43	2.19	2.19	L		
2	B	5	52	2.20	2.20		R	
2	B	6	30	2.20	2.20	L		
1	B	1	15	3.22	3.22	L		
1	B	2	15	3.22	3.22	L		
1	B	3	15	3.22	3.22	L		
2	B	4	15	3.22	3.22	L		

Figure 3-8 MB23, 2643 MW, 1 spb, network flow solution, economic schedule

MB23 NETWORK FLOW SCHEDULE ON THE COMPLEX MODEL
NEXT.
ONE SEGMENT PER BRANCH NF MODEL
NEXT.
BR

BUS	PL	QL	PG	QG	V	THETA
1	64.00	21.00	45.00	93.72	1.050	-1.924
2	101.00	34.00	136.98	94.81	1.050	0.000
3	0.00	0.00	0.00	0.00	1.041	-1.473
4	47.00	15.00	0.00	0.00	1.013	-1.418
5	51.00	17.00	-0.00	0.00	1.017	0.558
6	41.00	13.00	0.00	0.00	1.010	-0.117
7	48.00	16.00	0.00	0.00	0.999	-0.400
8	1.00	0.00	0.00	0.00	1.017	2.884
9	150.00	50.00	0.00	0.00	0.999	0.762
10	177.00	59.00	0.00	0.00	0.992	-3.824
11	130.00	43.00	179.00	133.35	1.050	-3.413
12	6.00	2.00	0.00	0.00	1.025	6.626
13	-4.00	-1.00	0.00	0.00	1.040	9.300
14	480.00	160.00	498.00	235.98	1.050	8.036
15	201.00	66.00	0.00	0.00	1.000	8.911
16	132.00	43.00	0.00	0.00	1.012	17.055
17	344.00	114.00	0.00	0.00	1.000	15.184
18	104.00	34.00	0.00	0.00	1.001	14.382
19	376.00	124.00	0.00	-0.00	0.996	14.439
20	-100.00	-33.00	803.00	223.08	1.050	23.129
21	375.00	123.00	0.00	0.00	1.021	18.901
22	-210.00	-70.00	0.00	0.00	1.020	18.095
23	129.00	43.00	1028.00	220.15	1.050	21.847

NEXT.

LN	SB	EB	PS	PR	QS	QR
1	1	3	-7.1	-7.2	22.0	21.7
2	1	4	6.6	6.0	52.0	50.2
3	2	5	2.5	2.0	35.9	34.8
4	8	5	49.7	49.0	-17.6	-19.6
5	2	7	15.0	14.5	27.6	26.2
6	3	6	-7.2	-7.6	24.1	23.2
7	4	9	-41.0	-41.8	36.7	34.7
8	9	7	33.8	33.5	-11.7	-12.4
9	8	6	49.5	48.6	-10.5	-13.0
10	11	10	49.0	46.9	92.9	87.5
11	8	10	82.5	79.0	-7.7	-16.9
12	9	10	52.6	51.1	-12.7	-16.8
13	13	14	63.7	63.5	-36.6	-38.4
14	14	12	81.5	81.1	64.4	60.9
15	15	12	120.6	119.8	-92.9	-100.1
16	18	15	325.0	321.6	-16.6	-47.4
17	23	13	333.8	325.8	14.9	-57.4
18	16	17	426.8	424.9	107.5	92.4
19	17	18	80.9	80.7	-13.1	-14.2
20	19	18	3.5	3.4	-44.5	-44.7
21	20	19	385.6	379.5	126.0	62.9
22	22	18	347.5	344.8	73.6	50.0
23	20	21	517.4	512.7	150.1	108.5
24	21	22	137.7	137.5	-5.1	-7.1
25	23	16	565.2	558.8	195.3	142.1
26	12	8	65.9	65.9	-19.5	-24.3
27	13	8	116.8	116.8	-1.8	-15.0
28	12	9	129.0	129.0	2.3	-11.0
29	13	9	149.3	149.3	21.5	-0.9
30	1	2	-18.5	-18.5	0.3	-0.3

NEXT.

Figure 3-9 MB23, 2643 MW, 1 spb, check solution

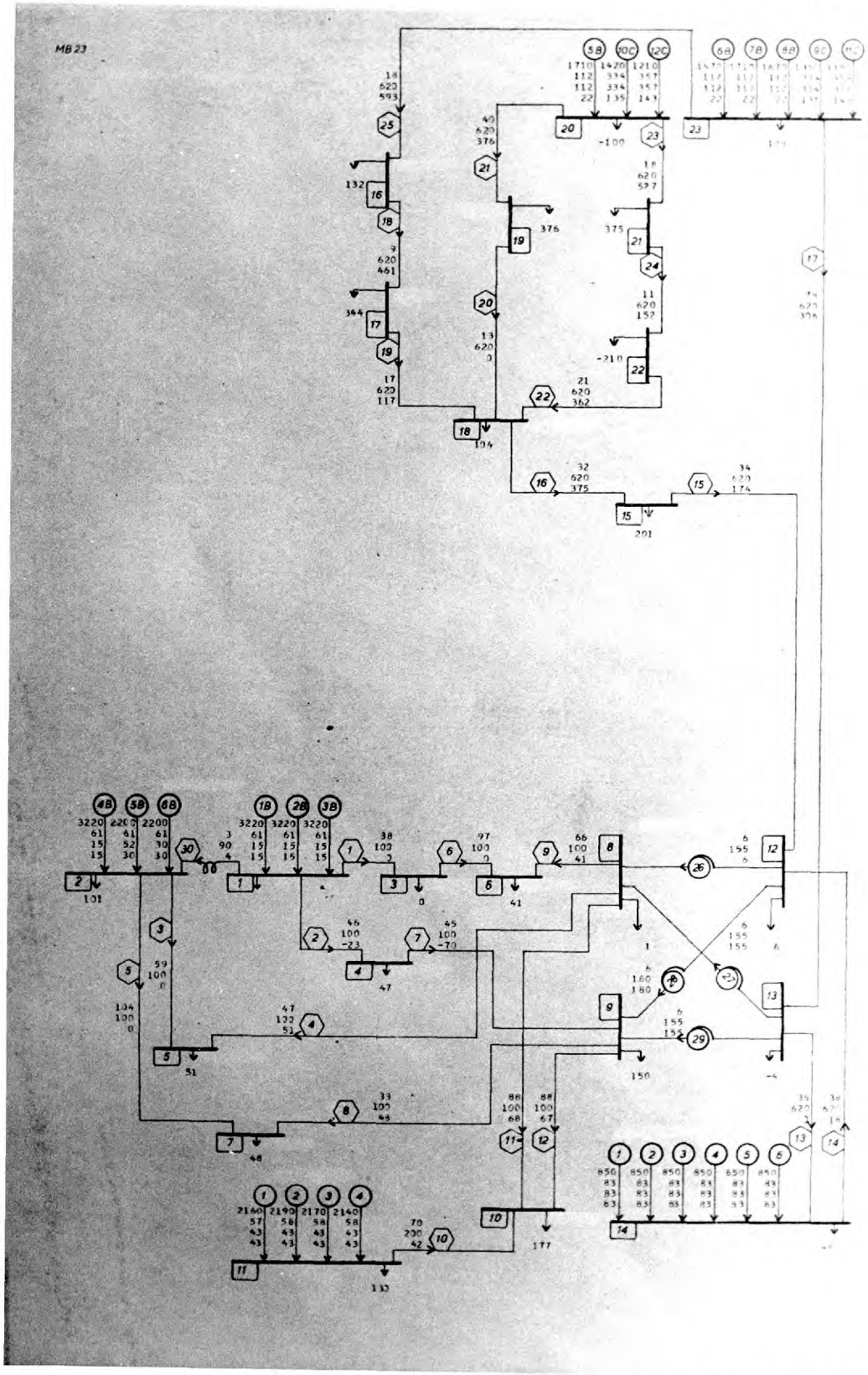


Figure 3-10 MB23, 2643 MW, 1 spb network flow solution

LINE DATA AND POWER FLOWS
25 LINES

LIN NO.	STT BUS NO.	END BUS NO.	DES VOL KV	LIN LEN MI	INCRML COST L/MWHR	THML LIMT MVA	POWR FLOW MW
1	1	3	132	15	0.0192	100	0
2	1	4	132	18	0.0230	100	-38
3	2	5	132	24	0.0299	100	-5
4	8	5	132	19	0.0720	100	56
5	2	7	132	42	0.0520	100	-2
6	3	6	132	40	0.0487	100	0
7	4	9	132	18	0.0676	100	-85
8	9	7	132	13	0.0169	100	50
9	8	6	132	27	0.0331	100	41
10	11	10	132	29	0.0354	200	64
11	8	10	132	36	0.1330	100	63
12	9	10	132	36	0.0443	100	50
13	13	14	275	53	0.0195	620	4
14	14	12	275	53	0.0195	620	22
15	15	12	275	47	0.0170	620	170
16	18	15	275	44	0.0485	620	371
17	23	13	275	108	0.0393	620	310
18	16	17	275	12	0.0135	620	457
19	17	18	275	24	0.0087	620	113
20	19	18	275	18	0.0066	620	0
21	20	19	275	56	0.0612	620	376
22	22	18	275	29	0.0320	620	362
23	20	21	275	25	0.0271	620	527
24	21	22	275	16	0.0058	620	152
25	23	16	275	25	0.0273	620	589

TRANSFORMER AND SERIES REACTOR DATA AND POWER FLOWS
4 TRANSFORMERS 1 SERIES REACTORS

XFR NO.	STT BUS NO.	END BUS NO.	HV KV	LV KV	INCRML COST L/MWHR	THML RATG MVA	POWR FLOW MW
26	12	8	275	132	0.0035	155	6
27	13	8	275	132	0.0104	155	155
28	12	9	275	132	0.0101	180	180
29	13	9	275	132	0.0104	155	155
30	1	2	132	132	0.0015	90	19

DEMAND DATA AND CHECK
23 DEMANDS

BUS NO.	DEMD POWR MW
1	64
2	101
3	0
4	47
5	51
6	41
7	48
8	1
9	150
10	177
11	130
12	6
13	-4
14	480
15	201
16	132
17	344
18	104
19	376
20	-100
21	375
22	-210
23	129

2643 MW

TOTAL NET DEMAND

Figure 3-11 MB23, 2643 MW, 2 spb network flow solution, branch power flows and demand data

GENERATOR DATA AND POWER OUTPUT
24 GENERATORS

BUS NO.	STA	SEG	GEN PWR MW	MIN GEN MW	ECR MW	MAX GEN MW	INCR COST L/MH	2ND COST L/MH
1	B	1	15	15		61	3.22	
1	B	2	15	15		61	3.22	
1	B	3	15	15		61	3.22	
2	B	4	15	15		61	3.22	
2	B	5	30	30		61	2.20	
2	B	6	30	30		61	2.20	
11		1	50	43		57	2.16	
11		2	43	43		58	2.19	
11		3	43	43		58	2.17	
11		4	58	43		58	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	B	5	112	22		112	1.71	
20	C	10	334	135		334	1.42	
20	C	12	357	143		357	1.21	
23	B	6	112	22		112	1.67	
23	B	7	112	22		112	1.71	
23	B	8	112	22		112	1.67	
23	C	9	334	135		334	1.35	
23	C	11	358	143		358	1.15	

2643 MW TOTAL GENERATED POWER

MERIT-ORDER TABLE
24 GENERATORS

BUS NO.	STA	SEG	GEN PWR MW	INCR COST L/MH	OPIN COST L/MH	MIN	ECR	MAX
14		1	83	0.85	0.85			U
14		2	83	0.85	0.85			U
14		3	83	0.85	0.85			U
14		4	83	0.85	0.85			U
14		5	83	0.85	0.85			U
14		6	83	0.85	0.85			U
23	C	11	358	1.15	1.15			U
20	C	12	357	1.21	1.21			U
23	C	9	334	1.35	1.35			U
20	C	10	334	1.42	1.42			U
23	B	6	112	1.67	1.67			U
23	B	8	112	1.67	1.67			U
23	B	7	112	1.71	1.71			U
20	B	5	112	1.71	1.71			U
11		4	58	2.14	2.14			U
11		1	50	2.16	2.16		R	
11		3	43	2.17	2.17	L		
11		2	43	2.19	2.19	L		
2	B	5	30	2.20	2.20	L		
2	B	6	30	2.20	2.20	L		
1	B	1	15	3.22	3.22	L		
1	B	2	15	3.22	3.22	L		
1	B	3	15	3.22	3.22	L		
2	B	4	15	3.22	3.22	L		

Figure 3-12 MB23, 2643 MW, 2 spb network flow solution, economic schedule

MB23 NETWORK FLOW SCHEDULE ON THE COMPLEX MODEL
NEXT.
ALSO A MERIT ORDER SCHEDULE
NEXT.

BR

BUS	PL	QL	PG	QG	V	THETA
1	64.00	21.00	45.00	99.20	1.050	-0.492
2	101.00	34.00	85.03	110.32	1.050	0.000
3	0.00	0.00	0.00	0.00	1.041	0.100
4	47.00	15.00	0.00	0.00	1.013	0.432
5	51.00	17.00	-0.00	0.00	1.017	1.810
6	41.00	13.00	0.00	0.00	1.010	1.880
7	48.00	16.00	0.00	0.00	0.999	1.297
8	1.00	0.00	0.00	0.00	1.018	5.169
9	150.00	50.00	0.00	0.00	0.999	3.042
10	177.00	59.00	0.00	0.00	0.993	0.838
11	130.00	43.00	231.00	112.64	1.050	3.041
12	6.00	2.00	0.00	0.00	1.026	8.906
13	-4.00	-1.00	0.00	0.00	1.040	11.580
14	480.00	160.00	498.00	235.37	1.050	10.316
15	201.00	66.00	0.00	0.00	1.000	11.190
16	132.00	43.00	0.00	0.00	1.012	19.334
17	344.00	114.00	0.00	0.00	1.000	17.462
18	104.00	34.00	0.00	0.00	1.001	16.661
19	376.00	124.00	0.00	-0.00	0.996	16.718
20	-100.00	-33.00	803.00	222.98	1.050	25.407
21	375.00	123.00	0.00	0.00	1.021	21.180
22	-210.00	-70.00	0.00	0.00	1.020	20.374
23	129.00	43.00	1028.00	219.95	1.050	24.126

NEXT.

LN	SB	EB	PS	PR	QS	QR
1	1	3	-11.5	-11.7	23.9	23.6
2	1	4	-2.8	-3.5	55.8	53.8
3	2	5	-19.0	-19.8	45.2	43.1
4	8	5	72.4	70.8	-23.8	-28.0
5	2	7	-1.7	-2.3	34.1	32.4
6	3	6	-11.7	-12.1	26.0	24.9
7	4	9	-50.5	-51.6	40.3	37.4
8	9	7	50.9	50.4	-17.1	-18.7
9	8	6	54.3	53.1	-11.6	-14.6
10	11	10	101.0	98.1	72.2	64.6
11	8	10	55.0	53.5	-0.8	-4.9
12	9	10	25.8	25.4	-4.9	-5.9
13	13	14	63.7	63.5	-36.3	-38.1
14	14	12	81.5	81.1	64.1	60.6
15	15	12	120.6	119.8	-93.0	-100.2
16	18	15	325.0	321.6	-16.7	-47.6
17	23	13	333.8	325.8	14.7	-57.5
18	16	17	426.7	424.9	107.5	92.3
19	17	18	80.9	80.7	-13.2	-14.3
20	19	18	3.5	3.4	-44.5	-44.8
21	20	19	385.6	379.5	126.0	62.8
22	22	18	347.5	344.8	73.6	49.9
23	20	21	517.4	512.7	150.0	108.4
24	21	22	137.7	137.5	-5.2	-7.1
25	23	16	565.3	558.8	195.2	142.0
26	12	8	65.8	65.8	-19.8	-24.5
27	13	8	116.8	116.8	-2.0	-15.2
28	12	9	129.0	129.0	2.0	-11.2
29	13	9	149.3	149.3	21.3	-1.1
30	1	2	-4.7	-4.7	0.0	-0.0

NEXT.

Figure 3-13 MB23, 2643 MW, 2 spb check solution

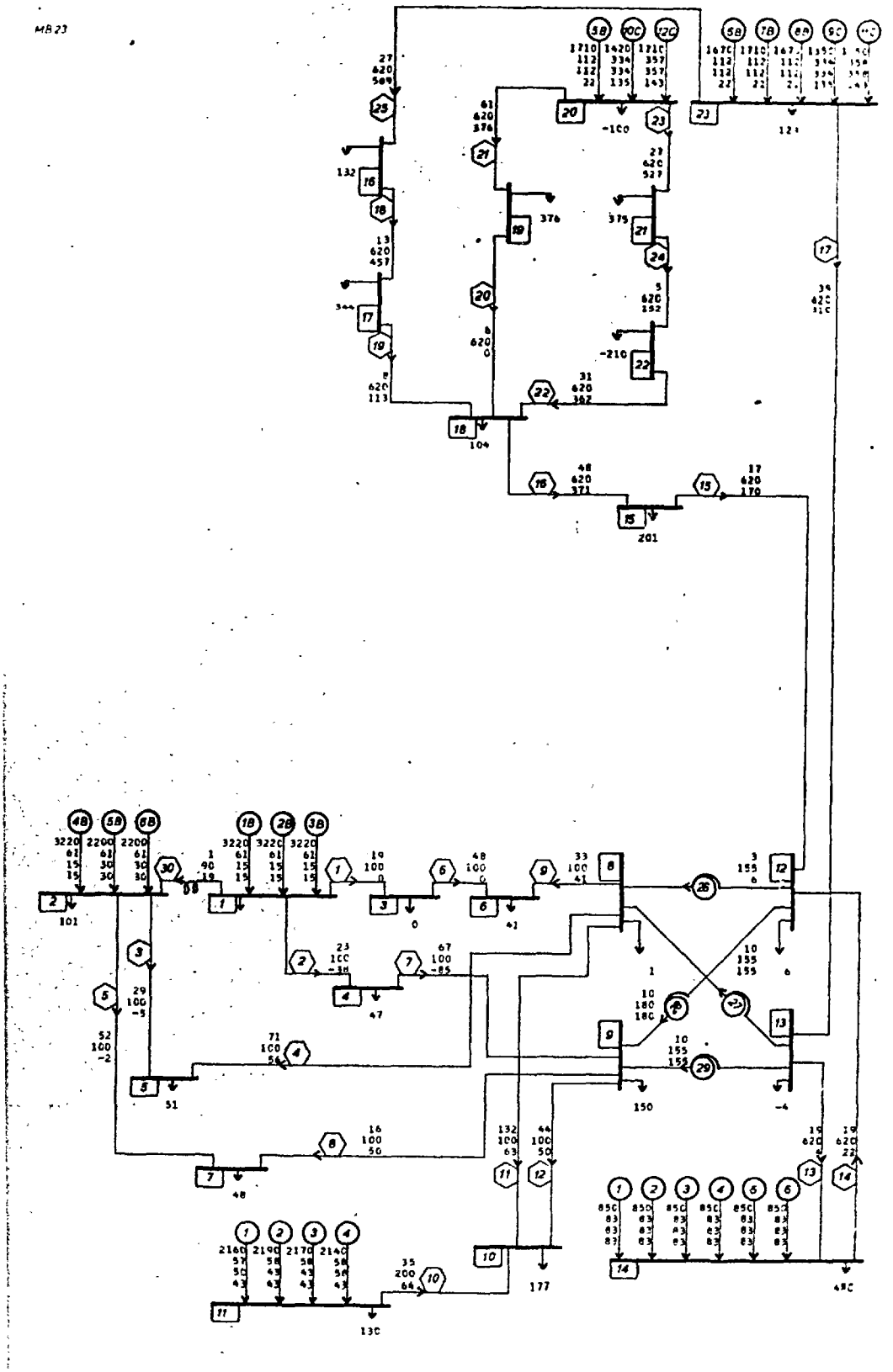


Figure 3-14 MB23, 2643 MW, 2 spb network flow solution

3.6.2 American test systems

The familiar quadratic cost of generation curves were used in the tests on the American systems because data for them was included in the original works. More basic data that includes the effect of the valve throttling losses to form the characteristically looped actual cost curve was not available. Ringlee and Williams (46) have shown that if it is, fuel cost savings of 0.1 to 0.2 per cent may be achieved using the 'stepped incremental' method instead of a quadratic approximation. Since, as far as the generation costing is concerned, the stepped incremental and the network flow piecewise linear models are equivalent, in practice the latter will produce schedules superior to those of the quadratic model.

Recall from Chapter 2 that the number of generator model segments is fixed by the number of valve loading points in the turbine itself. A unit will have from three to eight segments in its cost curve, (25) and a typical one might have five.(15) Each station in the American test systems is assumed to contain at least two turboalternator units. It is likely then, that ten or more segments are required to model adequately the generation at each bus. However, for purposes of comparison with the network flow results, the quadratic generator cost functions were retained with the complex variable models.

It can be seen in Table 3-5 that the schedules for

SQ19 are unusually accurate. This can be explained by noting that transmission losses are only 1.9% of the total load, and that the a_m coefficients in the generator cost curves are, on average, much lower than those of the other two systems. SQ19 is therefore not a very useful system for experimentation.

Initially, it may be disconcerting to note from Table 3-5 that the cost differences do not decrease monatonically with increasing segments per branch or per generator. The reason for this is that as the number of segments is changed, there is nothing to prevent a fortuitous arrangement of breakpoints yielding an unexpectedly accurate schedule. Selecting the best number of segments per branch is made somewhat uncertain because of this. Furthermore, the generator segmentation error tends in these experiments to obscure the branch segmentation error. In systems where transmission losses are important we may estimate that two or three segments per branch will permit the network flow algorithm to establish accurate economic load dispatches.

If KS28 with 10 to 20 segments per generator is chosen as the most representative test system and setting aside the experiment with one segment per branch, the maximum total scheduling error is 0.22%. Recalling that some fraction of this is due to avoidable generator segmentation error that would not be present in practice, an indication of the accuracy of the network flow method is obtained.

As with the British test systems, the solution time depends on the model size, and a strong motive exists to keep the number of arcs in the model to a minimum.

Let us check the operation of the network flow method by examining Figure 3-25, a semi-geographic diagram of KS28. Consider the generators at Sporn and Logan, for example. Their incremental generation costs are quite different: 2.10 and 2.59 $/h. Logan generation is at a minimum because it is cheaper to supply the loads there from Sporn over lines 32 and 15 at a delivered cost of 2.10+0.15+0.12 = 2.37$/h.

Figure 3-26 pictures the solution space of GHKS5 for the 266 MW load level. Transmission losses and generation cost are plotted in the feasible region of operation of the system. Note that for minimum losses, the Fort Wayne generator must be at full power output, and that this point is quite remote from the optimal one.

Table 3-5 Summary of computational results for the American test systems

Dem-	Network flow					Complex model check			
and	Model			Comput.time		Tran	Genern	Cost	Cost
				Alg.	Tot.	loss	cost	diff	diff
MW	spb	spg	arcs	sec	sec	MW	$/h	$/h	%
GHKS5									
266	5	5	70	0.32	7.12	5.9	962.66	0.17	0.018
266	2	2	31	0.08	6.53	5.5	963.81	1.32	0.14
266	1	5	30	0.13	6.62	6.3	964.56	2.07	0.22
266	2	5	40	0.15	6.72	7.0	964.75	2.26	0.23
186	5	5	70	0.22	7.05	2.7	625.80	0.08	0.013
186	2	2	31	0.07	6.62	4.1	633.16	7.44	1.19
SQ19									
311	5	5	271	3.48	15.32	5.4	451.62	0.23	0.051
311	2	2	115	1.00	11.73	5.8	451.52	0.13	0.029
311	10	10	531	9.82	23.25	5.9	451.39	0.00	0.00
217	5	5	271	1.97	13.78	3.1	304.46	0.22	0.072
217	2	2	115	0.70	11.52	3.3	304.36	0.12	0.039
KS28									
1730	5	5	417	23.98	41.88	87.9	3580.58	1.21	0.033
1730	5	10	457	31.35	49.38	87.3	3579.65	0.28	0.008
1730	2	10	247	10.90	27.52	88.9	3580.45	1.08	0.030
1139	5	5	417	11.68	29.65	52.0	1718.72	18.61	1.10
1139	5	10	457	13.77	31.80	40.9	1703.44	3.33	0.20
1139	2	10	247	4.95	21.53	41.0	1703.61	3.50	0.21
1139	2	20	327	8.97	26.17	41.3	1703.81	3.70	0.22
1139	1	10	177	clock off		40.6	1705.00	4.89	0.29

LINE DATA AND POWER FLOWS

5 LINES

LIN NO.	STT BUS NO.	END BUS NO.	DES VOL KV	LIN LEN MI	INCRML COST L/MWHR	THML LIMT MVA	POWR FLOW MW	START BUS NAME	END BUS NAME
1	1	2	132	75	0.2364	300	84	TWIN BRANC	FT WAYNE
2	2	5	132	47	0.2070	100	30	FT WAYNE	MARION
3	2	4	132	52	0.0966	100	20	FT WAYNE	MUNCIE
4	4	5	132	35	0.0259	100	0	MUNCIE	MARION
5	3	4	132	190	0.4740	140	50	PORTSMOUTH	MUNCIE

GENERATOR DATA AND POWER OUTPUT

3 GENERATORS

BUS NO.	S T A	S E T	GEN PWR MW	MIN GEN MW	MAX GEN MW	INCR COST L/MH	2ND COST L/MH	GENERATOR BUS NAME
1		1	170	50	200	2.45	5.00	TWIN BRANC
2		1	46	10	100	3.89	5.00	FT WAYNE
3		1	50	10	100	3.51	5.00	PORTSMOUTH

266 MW TOTAL GENERATED POWER

DEMAND DATA AND CHECK

4 DEMANDS

BUS NO.	DEMD POWR MW	DEMAND BUS NAME
1	86	TWIN BRANC
2	80	FT WAYNE
4	70	MUNCIE
5	30	MARION

266 MW TOTAL NET DEMAND

Figure 3-15 GHKS5, 266 MW, 5 spb 5spg network flow solution

GHKS 5

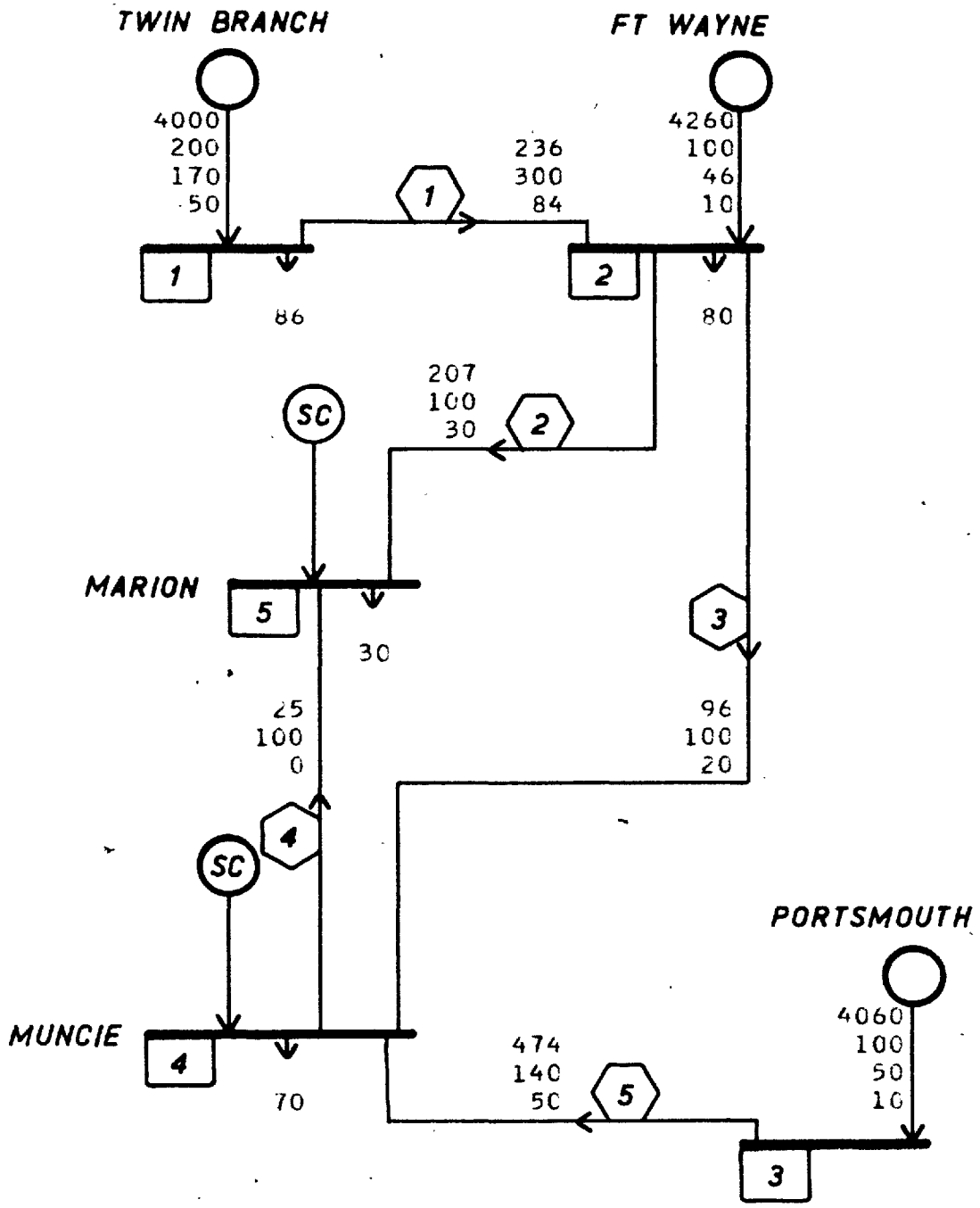


Figure 3-16 GHKS5, 266 MW, 5 spb 5 spg network flow solution

GHKS 5

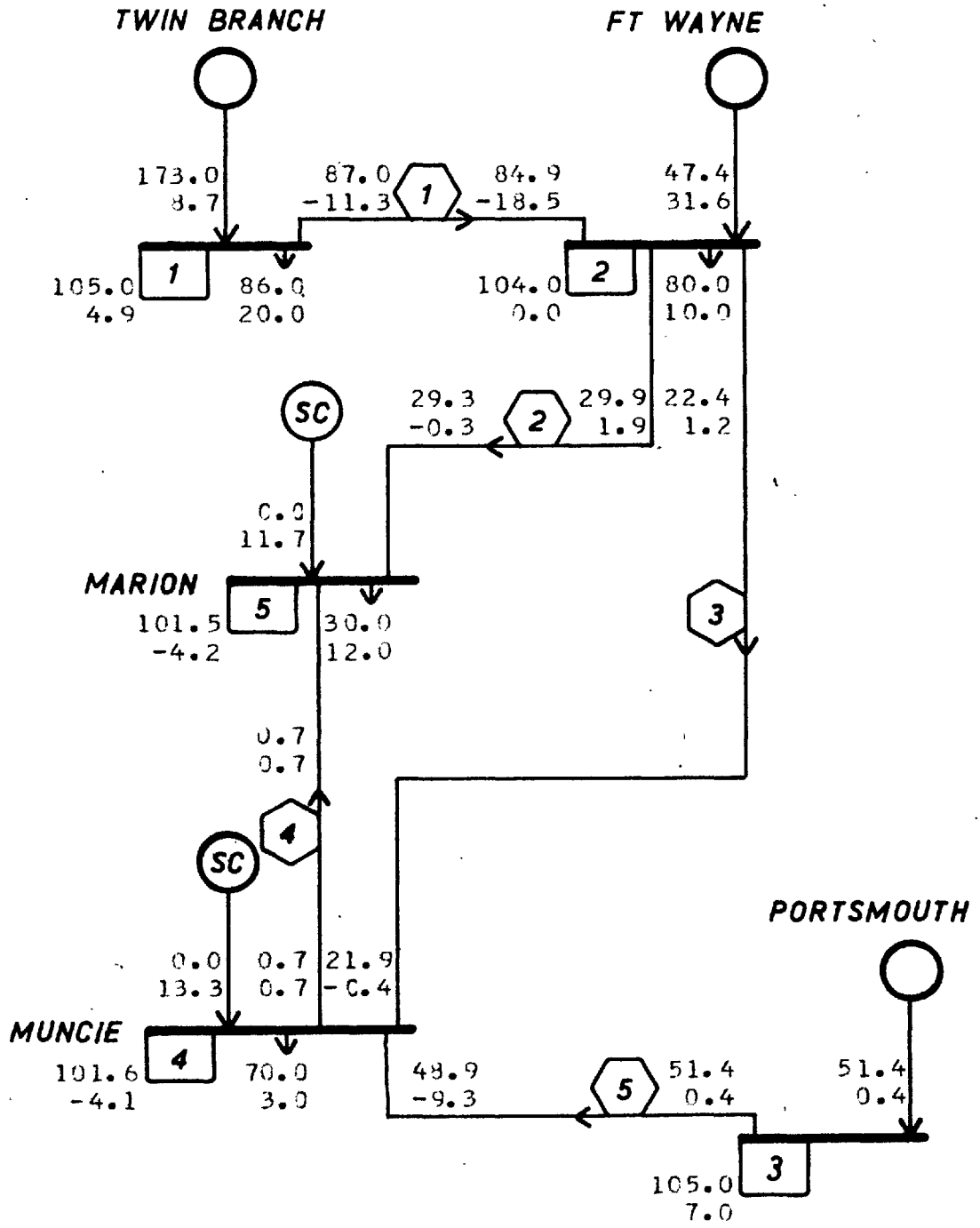


Figure 3-17 GHKS5, 266 MW, 5 spb 5 spg check solution

LINE DATA AND POWER FLOWS

24 LINES

LIN NO.	STT BUS NO.	END BUS NO.	DES VOL KV	LIN LEN MI	INCRML COST L/MWHR	THML LIMT MVA	POWR FLOW MW
1	2	1	-0	42	0.0125	250	0
2	2	3	-0	52	0.0150	250	17
3	18	2	-0	208	0.0662	275	60
4	18	1	-0	227	0.0748	275	93
5	18	10	-0	259	0.0322	275	55
6	1	10	-0	32	0.0130	200	80
7	10	4	-0	6	0.0056	100	58
8	4	5	-0	14	0.0032	300	32
9	6	5	-0	36	0.0110	250	21
10	16	6	-0	32	0.0095	250	21
11	6	13	-0	30	0.0067	250	0
12	10	11	-0	5	0.0027	100	40
13	12	11	-0	4	0.0008	100	-1
14	19	12	-0	6	0.0056	100	59
15	19	16	-0	42	0.0127	250	0
16	19	3	-0	23	0.0068	250	-17
17	16	7	-0	62	0.0185	250	14
18	7	8	-0	76	0.0224	250	4
19	8	9	-0	66	0.0088	200	14
20	9	15	-0	50	0.0067	200	0
21	17	8	-0	171	0.0228	200	26
22	17	16	-0	214	0.0285	200	0
23	3	14	-0	84	0.0247	250	0
24	17	14	-0	142	0.0383	250	0

DEMAND DATA AND CHECK

10 DEMANDS

BUS NO.	DEMD PWR MW	DEMAND BUS NAME
1	13	
2	43	
4	26	
5	53	
7	10	
8	16	
9	14	
10	37	
11	39	
12	60	

311 MW TOTAL NET DEMAND

GENERATOR DATA AND POWER OUTPUT

4 GENERATORS

BUS NO.	S T A	S E T	GEN PWR MW	MIN GEN MW	MAX GEN MW	INCR COST L/MH	2ND COST L/MH
16		1	35	5	55	1.52	1.67
17		1	26	10	45	1.48	2.83
18		1	208	70	300	1.26	0.62
19		1	42	15	60	1.20	5.00

311 MW TOTAL GENERATED POWER

Figure 3-18 SQ19, 311 MW, 5 spb 5 spg network flow solution

SQ19 NETWORK FLOW SCHEDULE ON THE COMPLEX MODEL
NEXT.

BR

BUS	PL	QL	PG	QG	V	THETA
1	12.70	9.20	0.00	0.00	0.991	1.209
2	42.50	18.50	0.00	0.00	0.990	1.388
3	0.00	0.00	0.00	0.00	0.997	0.448
4	26.20	11.40	0.00	0.00	0.990	-0.204
5	53.40	22.60	0.00	0.00	0.987	-0.496
6	0.00	0.00	0.00	0.00	0.994	-0.206
7	9.80	5.30	0.00	0.00	0.992	-0.646
8	15.70	10.10	0.00	0.00	0.987	-1.112
9	14.60	6.60	0.00	-0.00	0.985	-1.682
10	36.60	16.30	0.00	0.00	0.992	-0.002
11	38.70	12.20	0.00	0.00	0.993	-0.142
12	60.50	22.20	0.00	0.00	0.995	-0.159
13	0.00	1.50	0.00	0.00	0.994	-0.199
14	0.00	10.60	0.00	0.00	0.995	0.604
15	0.00	1.80	0.00	0.00	0.988	-1.714
16	0.00	0.00	35.55	8.24	1.000	0.044
17	0.00	0.00	26.45	-0.15	1.000	0.703
18	0.00	0.00	211.45	2.47	1.000	10.292
19	0.00	0.00	42.60	71.28	1.000	0.000

NEXT.

PF

LN	SB	EB	PS	PR	QS	QR
1	2	1	6.2	6.1	-3.1	-3.1
2	2	3	24.0	23.8	-20.7	-21.2
3	18	2	73.9	72.6	2.1	-9.3
4	18	1	68.9	67.7	1.9	-9.0
5	18	10	68.6	67.2	1.2	-11.0
6	1	10	61.2	60.9	-17.0	-18.3
7	10	4	62.0	61.9	8.4	8.2
8	4	5	35.7	35.6	7.3	7.1
9	6	5	17.8	17.7	11.9	11.8
10	16	6	17.9	17.8	12.0	11.8
11	6	13	-0.0	-0.0	1.1	1.1
12	10	11	29.5	29.4	-36.1	-36.2
13	12	11	9.3	9.3	37.0	37.0
14	19	12	70.0	69.8	48.2	47.8
15	19	16	-1.6	-1.6	0.6	0.6
16	19	3	-25.8	-25.9	23.1	22.9
17	16	7	21.3	21.2	6.5	6.2
18	7	8	11.4	11.3	2.5	2.4
19	8	9	14.6	14.6	-0.1	-0.2
20	9	15	0.0	0.0	-5.4	-5.4
21	17	8	19.1	19.0	4.4	3.8
22	17	16	5.2	5.2	-1.0	-1.0
23	3	14	-2.1	-2.1	3.2	3.2
24	17	14	2.1	2.1	3.1	3.1

NEXT.

Figure 3-19 SQ19, 310.7 MW, 5 spb 5 spg check solution

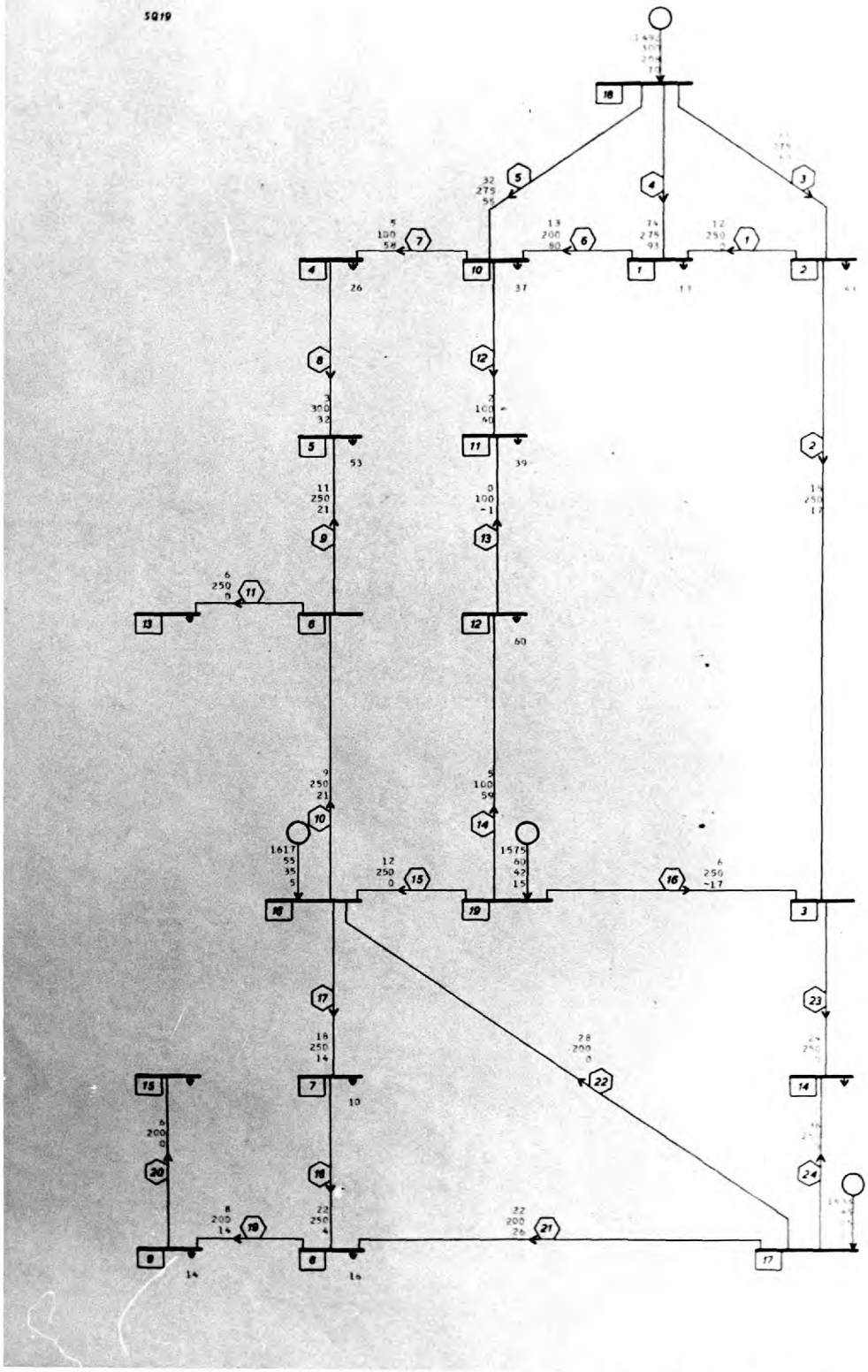


Figure 3-20 SQ19, 311 MW, 5 spb 5 spg network flow solution

LINE DATA AND POWER FLOWS
32 LINES

LIN NO.	STT BUS NO.	END BUS NO.	DES VOL KV	LIN LEN MI	INCRML COST L/MWHR	THML LIMT MVA	POWR FLOW MW	START BUS NAME	END BUS NAME
1	1	9	132	70	0.0289	270	-1	TWIN BRANC	FT WAYNE
2	9	10	132	70	0.0253	90	0	FT WAYNE	MARION
3	9	22	132	80	0.0355	90	0	FT WAYNE	MUNCIE
4	22	10	132	35	0.0086	90	18	MUNCIE	MARION
5	19	9	132	56	0.0793	180	48	LIMA	FT WAYNE
6	13	19	132	54	0.1133	180	80	FOSTORIA	LIMA
7	14	13	132	45	0.1038	180	86	HOWARD	FOSTORIA
8	4	14	132	80	0.1322	180	98	PHILO	HOWARD
9	5	4	132	75	0.0358	180	9	SPORN	PHILO
10	5	16	132	60	0.1718	180	138	SPORN	SOUTH POIN
11	15	22	132	175	0.1983	180	56	PORTSMOUTH	MUNCIE
12	16	15	132	35	0.0736	180	84	SOUTH POIN	PORTSMOUTH
13	17	16	132	52	0.0226	180	0	TURNER	SOUTH POIN
14	17	6	132	20	0.0117	270	1	TURNER	CABIN CREK
15	17	7	132	42	0.1227	180	98	TURNER	LOGAN
16	7	18	132	45	0.0623	180	59	LOGAN	SWITCHBACK
17	18	8	132	33	0.0122	180	4	SWITCHBACK	GLEN LYN
18	4	25	132	75	0.0579	270	0	PHILO	CANTON
19	2	25	132	50	0.0840	180	99	TIDD HV	CANTON
20	3	25	132	60	0.0302	180	36	WINDSOR HV	CANTON
21	26	27	66	15	0.0163	50	0	TIDD LV	WINDSOR LV
22	21	13	132	90	0.0415	90	9	SANDUSKY	FOSTORIA
23	11	21	132	30	0.0419	90	26	LORAIN	SANDUSKY
24	20	11	132	42	0.0245	180	36	ASHLAND	LORAIN
25	20	14	132	25	0.0151	90	13	ASHLAND	HOWARD
26	23	20	132	22	0.0521	90	27	MASSILLON	ASHLAND
27	12	20	132	110	0.1473	90	45	BURGER	ASHLAND
28	12	23	132	79	0.0776	90	36	BURGER	MASSILLON
29	12	24	132	82	0.0872	90	30	BURGER	ALLIANCE
30	25	23	132	12	0.0094	90	8	CANTON	MASSILLON
31	24	25	132	19	0.0098	90	0	ALLIANCE	CANTON
32	5	17	132	50	0.1520	180	128	SPORN	TURNER

TRANSFORMER AND SERIES REACTOR
DATA AND POWER FLOWS
3 TRANSFORMERS 0 SERIES REACTORS

XFR NO.	STT BUS NO.	END BUS NO.	HV KV	LV KV	INCRML COST L/MWHR	THML RATG MVA	POWR FLOW MW	START BUS NAME	END BUS NAME
33	28	3	132	0	0.0113	90	-48	WINDSOR HV	WINDSOR CP
34	3	27	0	66	0.0022	45	0	WINDSOR CP	WINDSOR LV
35	2	26	132	66	0.0022	60	0	TIDD HV	TIDD LV

Figure 3-21 KS28, 1139 MW, 5 spb 10 spg network flow solution, branch power flows

DEMAND DATA AND CHECK
24 DEMANDS

BUS NO.	DEMD POWR MW	DEMAND BUS NAME
1	86	TWIN BRANC
2	51	TIDD HV
4	81	PHILO
5	5	SPORN
6	117	CABIN CREK
7	84	LOGAN
8	84	GLEN LYN
9	47	FT WAYNE
10	18	MARION
11	28	LORAIN
13	15	FOSTORIA
14	25	HOWARD
15	28	PORTSMOUTH
16	54	SOUTH POIN
17	29	TURNER
18	55	SWITCHBACK
19	32	LIMA
20	23	ASHLAND
21	17	SANDUSKY
22	38	MUNCIE
23	17	MASSILLON
24	30	ALLIANCE
25	127	CANTON
28	48	WINDSOR HV

1139 MW TOTAL NET DEMAND

GENERATOR DATA AND POWER OUTPUT
10 GENERATORS

BUS NO.	S T A	S T E	GEN PWR MW	MIN GEN MW	MAX GEN MW	INCR COST L/MH	2ND COST L/MH	GENERATOR BUS NAME
1		1	85	50	400	1.28	8.20	TWIN BRANC
2		1	150	60	240	0.80	4.40	TIDD
3		1	84	40	160	1.81	1.90	WINDSOR
4		1	170	100	455	0.66	4.29	PHILO
5		1	280	70	310	0.89	2.22	SPORN
6		1	116	116	220	0.30	12.00	CABIN CREK
7		1	45	45	85	0.63	20.80	LOGAN
8		1	80	80	136	0.57	12.70	GLEN LYN
11		1	18	18	18	0.70	15.00	LORAIN
12		1	111	111	111	2.00	2.00	BURGER

1139 MW TOTAL GENERATED POWER

Figure 3-22 KS28, 1139 MW, 5 spb 10 spg network flow solution

KS28 KIRCHMAYER AND STAGG AEP 28 BUS SYSTEM
KS28 D1 LOADS NETWORK FLOW SCHEDULE ON THE COMPLEX MODEL

BUS	PL	QL	PG	QG	V	THETA
1	86.00	30.00	93.10	33.42	1.012	-23.176
2	51.00	25.00	154.80	13.23	1.002	5.006
3	0.00	0.00	87.20	17.99	1.001	2.934
4	81.00	38.80	179.28	39.91	1.040	0.000
5	5.10	3.10	286.20	-27.29	1.050	1.062
6	117.00	28.00	120.40	39.64	1.036	-7.502
7	84.00	10.30	46.70	29.14	1.028	-12.844
8	83.80	-6.00	82.70	21.31	1.040	-16.859
9	47.30	31.80	0.00	10.98	0.996	-23.358
10	17.90	15.20	0.00	14.34	0.990	-24.391
11	28.00	8.00	18.00	19.03	1.000	-8.769
12	0.00	0.00	111.00	-20.46	1.000	5.303
13	15.00	8.00	0.00	10.94	1.004	-13.771
14	25.10	15.00	0.00	15.63	1.008	-8.276
15	28.20	10.00	0.00	7.90	1.012	-11.689
16	53.40	20.00	0.00	0.00	1.016	-8.040
17	29.10	16.00	0.00	-2.41	1.030	-7.472
18	55.00	24.90	0.00	0.00	1.022	-16.510
19	32.00	15.00	0.00	20.58	1.000	-19.116
20	23.00	13.00	0.00	0.00	0.995	-6.064
21	17.00	8.00	0.00	0.00	0.992	-10.879
22	38.20	25.00	0.00	11.85	0.990	-23.566
23	16.90	2.90	0.00	0.00	0.990	-2.590
24	30.00	5.00	0.00	0.00	0.983	-1.178
25	127.00	86.00	0.00	51.76	0.987	-1.285
26	0.00	0.00	0.00	0.00	1.005	4.394
27	0.00	0.00	0.00	0.00	1.000	3.176
28	47.50	24.60	0.00	0.00	0.993	2.058

Figure 3-23 KS28, 1138.5 MW, 5 spb 10 spg check solution

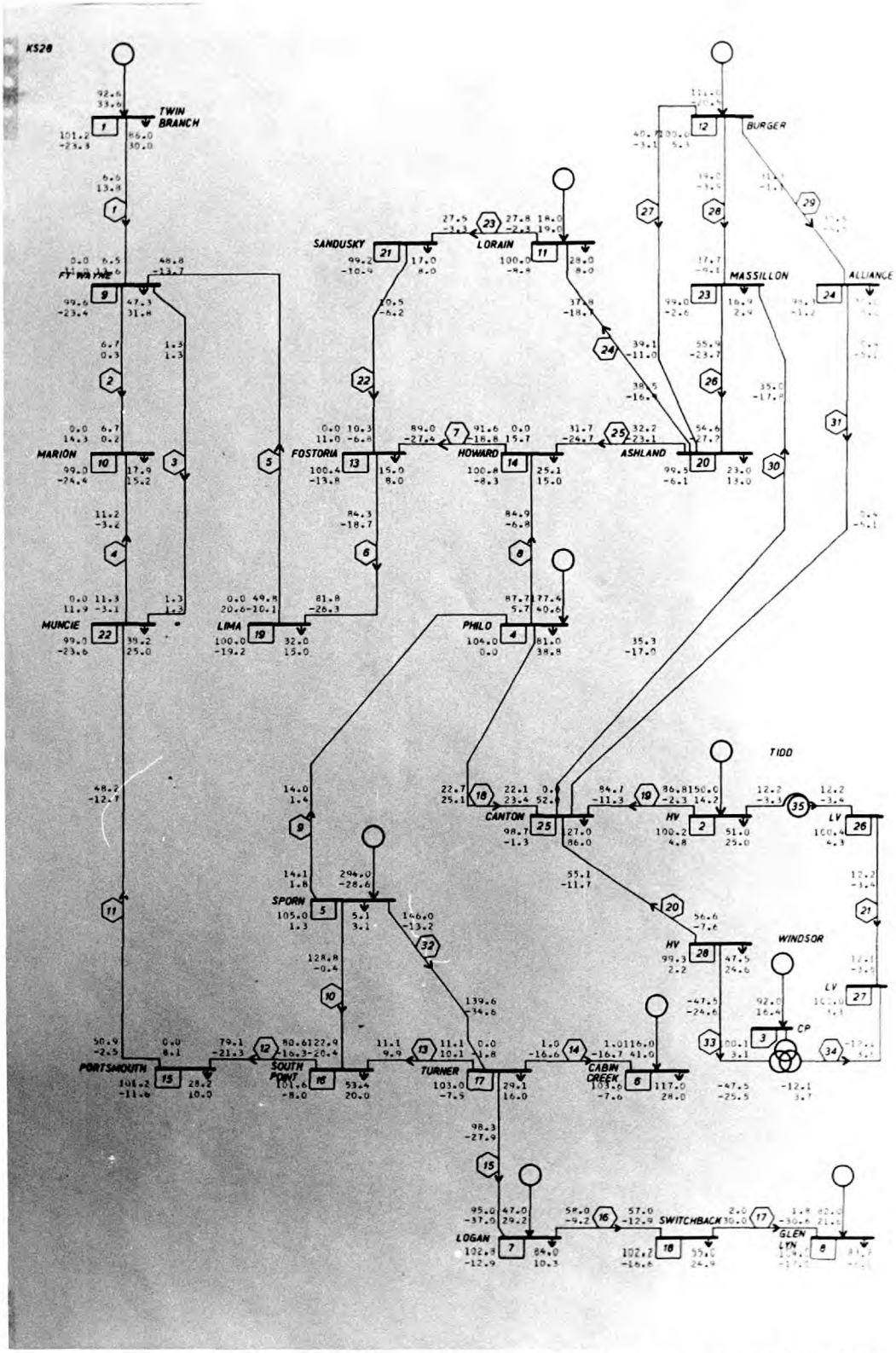


Figure 3-24 KS28, 1138.5 MW, 5 spb 10 spg check solution

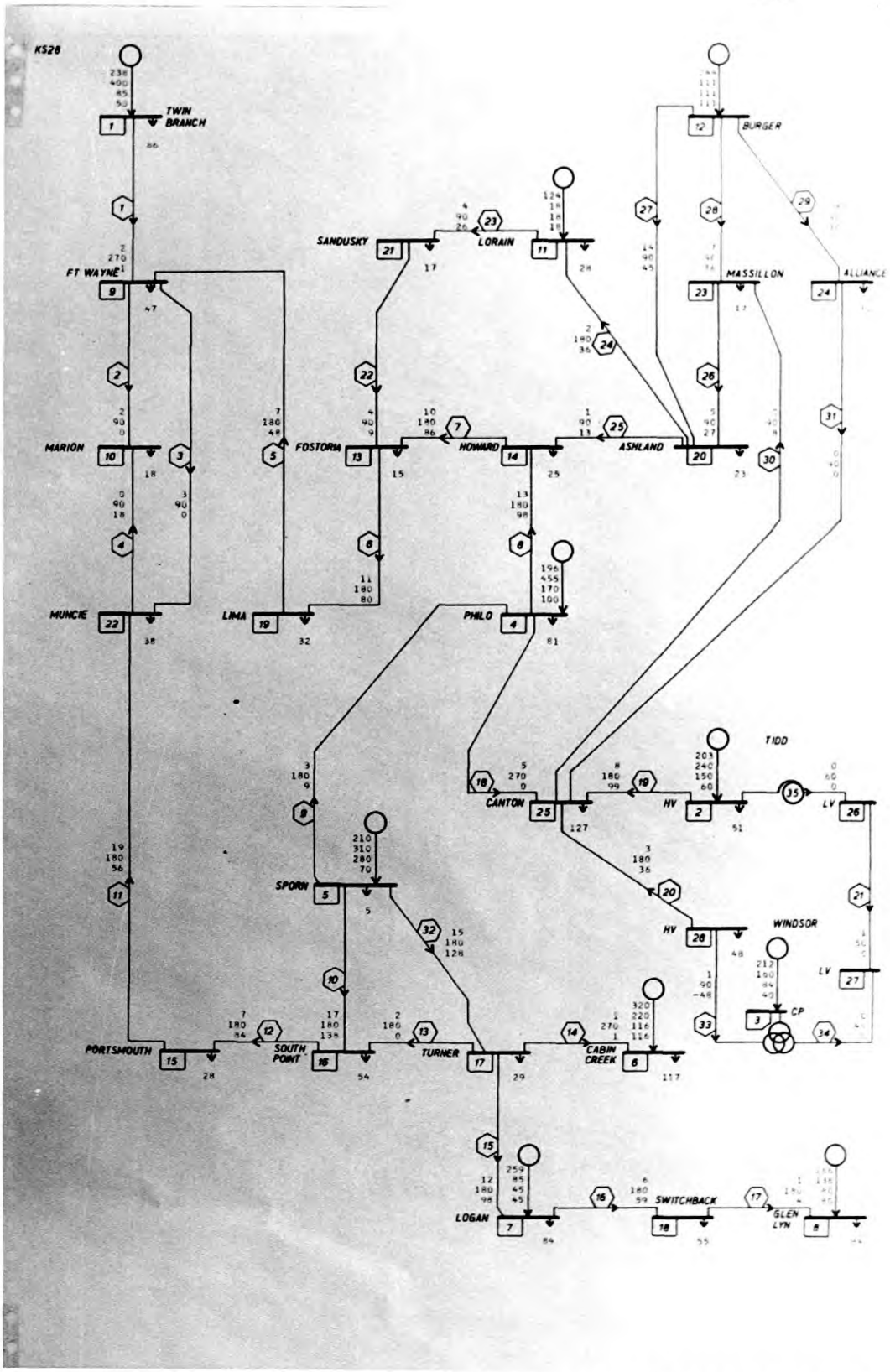


Figure 3-25 KS28, 1139 MW, 5 spb 10 spg network flow solution

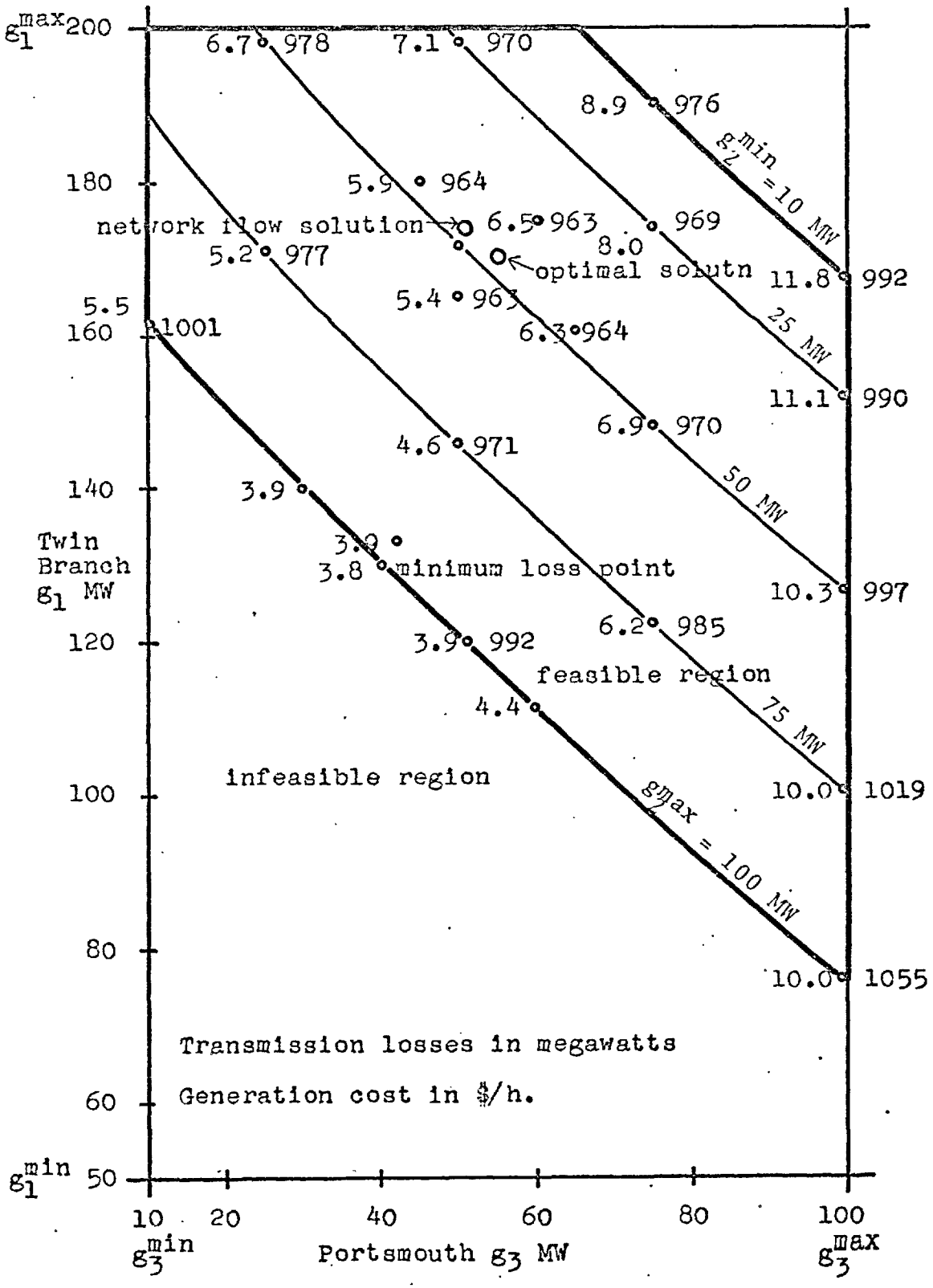


Figure 3-26 GHKS5 solution space

Chapter 4 FURTHER INVESTIGATIONS

4.1 Cost of load prediction errors

In the previous chapter, the magnitude of the cost errors associated with the network flow method was found and discussed. It is of some interest to identify the source and estimate the size of scheduling errors due to other causes, for comparison with those inherent in the network flow approach. This concern is taken up in the next three sub-sections.

The error in prediction of the system load is a function of: prediction span, time of day, power system studied, noise in the demand measuring system, and the prediction method. It can lie between 0.7% (38) and 4.0% (19) r.m.s. depending upon these conditions. For the purposes of illustration, a 3.0% load prediction error was selected for the experiment.

CMLFP was used on the 1138.5 MW demand level of KS28 to calculate the precise cost of a 3.0% error in load prediction. From the optimal solution at that load level, the generators were allowed to pick up the 3.0% load difference in proportion to their size. They were then generating 1215.5 MW to supply the augmented load of 1172.6 MW at a cost of 1787.85 $/h. An optimal solution was found for the 3.0% higher load level; being the schedule the generators would have been instructed to follow in the absence of a load prediction error. In this case they

generated 1216.12 MW at a cost of 1785.14 $/h.

The difference between the two cases is the cost of the load prediction error, and is 2.71 $/h, or 0.15%. Thus our example has demonstrated that this error is comparable in magnitude to that intrinsic to the network flow method.

4.2 Optimizing the voltage profile

System voltage is controlled by changing the excitation on the alternators and synchronous capacitors, and by adjusting transformer taps. The primary goal of this activity is to satisfy the bus voltage magnitude constraints, both for the intact network and for one that has suffered the loss of a key item of plant. Nevertheless, at medium load levels there may be sufficient operational latitude to permit a second objective to be addressed; that of optimizing the voltage profile to minimize transmission losses.

The network flow formulation of the thesis does not treat this problem. Therefore we should estimate the cost error that is incurred in failing to make optimizing adjustments to the network voltage profile.

The medium-level demand of 1138.5 MW on test system KS28 was chosen as a typical situation permitting voltage profile improvement. With the optimal solution as a base, CMLFP was used to make random displacements of between

one and four per cent to the generator bus voltages. The resulting cost difference was 1.29 $/h, or 0.08% of the optimal cost.

In practice then, it appears that the cost savings realizable through optimization of the voltage profile are small and similar in size to the scheduling errors.

4.3 Cost of generator data inaccuracies

What cost errors will be experienced for a given error in one of the data items associated with a generator? Such an error can arise through measurement inaccuracy, by failure to report a change, or in data transmission. Test system MB23 furnishes two convenient examples.

Suppose first, that a one per cent error is present in the incremental cost for a marginal generator such as B5 in bus 2, Figure 3-8. Its true incremental cost is then 2.22 £/MWh. Swinging 22 MW of generation to B6, a cheaper set, would save 0.44 £/h.

Consider next a 1 MW error in the capacity of generator B5 at bus 20. If this unit were operated at its true limit of 113 MW, and the output of the 'in range' set B5 at bus 2 reduced by 1 MW, the exchange would result in a saving of 0.49 £/h

These cost errors are but 0.01% of the total generation cost; an order of magnitude below other scheduling-related errors.

4.4 Including estimated losses with demands

It is evident from section 4.1 that some unnecessary expense can be incurred when generation is allowed to regulate unscheduled, even for load differences as small as three per cent. In Table 3-2 it may be observed that system transmission losses are a similar percentage of the total load. Recall that the network flow power balances do not include the transmission losses, and that in the check solutions the generators pick up the losses. As a consequence, an error much like the load forecasting error is present.

An improvement on this situation is apparent: imbed the losses in the predicted demands to minimize the schedule discrepancy. Even though the losses will vary with total load level, demand diversity, and generation pattern; a first-order estimate of their magnitude will be available and adequate. The total loss can then be apportioned to the various demand points. Then when the network flow schedule is applied to the system, much less unscheduled generator regulation will take place.

KS28 was employed in checking the proposed improvement. Adding 41 MW losses to the 1139 MW demand pattern gave demands totalling 1180 MW as input to NFESP. The system model had 5 segments per branch and 10 segments per generator. The resulting schedule was checked on CMLFP using the original 1138.5 MW demand data. An off-optimal

cost error of 2.35 $/h was observed, a 0.06% improvement over the original 0.20% cost error.

4.5 Comparison of branch power flows

The branch power flows produced by the network flow scheduling program are not accurate. While they satisfy a power balance equation at each bus, they do not obey Ohm's Law and hence voltage equilibrium equations. The object of the network flow formulation was to produce accurate generation schedules, and some lack of realism in the branch power flows is a consequence of the model we have chosen. Thus, while the power flow inaccuracies are not of primary concern, it is of interest to compare the flows produced by the network flow approximation to those of higher models.

As the number of segments per branch is increased, the realism of the branch power flows improves, approaching that of the linear model, (Number 2 in section 1.2). Naturally, it is prohibitive from a computational standpoint to follow this course in a practical application.

Table 4-1 compares the branch power flows resulting from network flow models of GHKS5 with those of a complex variable load flow. There are some minor a priori discrepancies superimposed on the comparison, due to transmission losses and slightly differing generation schedules.

Table 4-1 GHKS5, 266 MW, branch power flows

Branch number	Branch power flows			
	Network flow model			Complex variable model
	1spb	2spb	5spb	
1	70	84	84	84.4
2	36	36	30	28.2
3	0	0	20	20.5
4	-6	-6	0	2.4
5	64	64	50	55.2

4.6 Generator model additions

In the older type of CEGB 'range' station, several boilers feed steam to the turbogenerators through a common header. If the total boiler capacity happens to be less than the generation capacity, not all the generators can run at full power output. This operational situation can easily be modelled by the insertion of an arc between the source node and the generator arcs. The arc upper bound corresponds to the boiler limit.

The same type of model addition can account for the effect of station limits. For example, a station may have a contract to sell low pressure steam and this will place a lower limit on the total generation there. Alternatively, a temporary restriction such as the failure of an auxilliary or insufficient cooling water may limit the total power output to a value below the capabilities of the spinning

plant.

Some system security requirements can be met by scheduling generation so as to maintain a reserve of spinning capacity. The schedule could be modified by reducing generator limits on the more expensive sets, correctly distributed geographically, until the total spinning spare required is available. As an alternative to this informal approach, a network flow formulation incorporating arcs that model the flow of spare capacity in megawatts should be possible.

4.7 Transmission security

The transmission system must be secured against the loss of any line or any likely combination of lines. An off-line, multiple load flow computer study can identify lines that, when lost, will result in overloaded lines in the remaining network. As indicated in section 2.1, the power flow limits on the sensitive lines can then be set to lower, secure values. When the network flow economic scheduling program is re-applied, the new limits will force a modification of the economic schedule. This new schedule will be more costly, but will have a geographical distribution such that the network is secure against line outages.

An alternative method of enhancing transmission security is that of group limits. From operational experience,

or following off-line security checks, a group of generators and lines can be identified as insecure under some conditions. 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 will not be permitted to exceed a fixed value, I_g for import, E_g for export.

Expressing the rule algebraically

$$\sum_{i \varepsilon D_g} d_i - I_g \leq \sum_{m \varepsilon G_g} g_m \leq \sum_{i \varepsilon D_g} d_i + E_g$$

leads us directly to its 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 modelled by a single arc as was done in the preceeding section.

Chapter 5 CONCLUSIONS

This thesis has proposed a network flow approach to modelling and solving the economic load dispatching problem. In the present chapter we examine the results of the foregoing ones, and draw some conclusions on the applicability of the method to power system operations and planning.

The first question to be answered is:"How accurate is the method, and is it sufficiently accurate to be of practical use?" The question is relevant because we began by seeking a rather coarse-structured formulation, in the interest of solution speed.

From the computational results of Chapter 3 on the British test systems, it may be concluded that a 1 segment per branch model will produce accurate generation schedules. Indeed, with this degree of approximation, no cost error greater than 0.065% was recorded. Cost errors engendered by the load prediction task are of this magnitude or greater. For example, in the experiment of section 4.1, a 3% load prediction error caused a 0.15% error in cost. Also, reference 7 reports that a 1% prediction error gave a 0.08% cost error, and a 5% prediction error brought on a 0.4% cost error. Thus from these cases we may affirm that the accuracy of the network flow method is appropriate to the accuracy of the problem demand data.

The results for the American test systems were partially obscured by the presence of two components of error

that in practice can be avoided. After estimating the magnitude of these errors, a better indication of the accuracy of the network flow formulation will be obtained.

From a theoretical point of view it was interesting to follow in Table 3-5 the marked improvement in schedule accuracy with increasing the number of arcs in the generator models. That is, the piecewise linear approximation to the quadratic cost curves became finer and more exact in its effect. However, as was noted in the same sub-section, it is the linearly segmented function that closely represents reality. The quadratic is merely a less accurate mathematical model of the looped generator input-output cost relation. Ringlee and Williams (46) observed that the piecewise linear model is 0.1 to 0.2% more accurate than the quadratic one. Thus a significant portion of the cost error listed in Table 3-5 should not be attributed to the network flow method, but to any approach using the classical quadratic cost curves.

A second unnecessary component of error is that described in section 4.4. If the model is adjusted so that the demands include an estimate of the transmission losses, cost error reductions similar in magnitude to load prediction cost errors can be realized. The experiment of section 4.4 demonstrated a 0.06% reduction, about half the expected improvement. It is likely that the generator segmentation error of the preceeding paragraph rendered partially

unobservable the full cost reduction.

Based on the American test system results, it was decided in sub-section 3.6.2 that two or three segments per branch were adequate to accurately account for the cost of power transport. With the generator segmentation set at practical values, a worst-case error of 0.23% was experienced, for a two segment per branch network model. The two avoidable error effects just described each have a magnitude of about 0.1% Thus it seems reasonable to expect a final scheduling cost error of not greater that 0.1% for the network flow method. As with the British systems, this error magnitude compares favourably with the best load prediction cost errors. Hence we should be permitted to conclude that the network flow approach can accurately schedule power systems of the American type, where transmission losses usually have an influence on the economic dispatch.

The problem of calculating economic generation schedules occurs in two areas of application: off-line predictive studies, and on-line control action. The same network flow formulation and solution method can be used in both situations, but the emphasis may be seen to differ somewhat between the two. The second question requiring an answer is:"Are the network flow model solution times low enough to permit its application to problems of practical size?"

Off-line planning studies may require economic schedules of generation to be calculated as an integral part of each study. Computing time is not of central concern, as large utilities can usually justify problem solving times of many minutes on large machines.

The CEGB operates a national power system supplying England and Wales. Although of limited geographical extent, in terms of installed capacity and number of items of plant it is one of the world's largest. There are approximately 1000 generating units in the system. However, for any given study, the number requiring scheduling will be reduced by the following considerations. Firstly, only those sets ordered up to meet the demand are included. Secondly, nuclear plant and other large units supplying the base load can be netted out of the study since they have no economic competitors. If the load level is high, more sets can be withdrawn from the computation in this way. If the load level is low, fewer sets will be synchronized to the system in the first place. Lastly, units running at a fixed output level may be omitted; for example, pumped storage generators.

After this has been done, the remaining sets are subject to economic scheduling and possible participation as suppliers of marginal power transfers. Test system SG149 contained 177 scheduled generators connected to the supergrid transmission network. The two segment per branch model allowed an accurate generation schedule to be computed

in 3 minutes. Including lower voltage networks and some additional generators, the estimated size of model required for a national economic schedule would be three times that of SG149. With model solution time varying as the square or fractionally higher exponent of the model size, it is expected that a national economic schedule can be produced in less than one hour.

Test system GR142 serves to illustrate the second area of applicability of the economic scheduling program, that of on-line instruction of generator output power levels. In this case solution time is critical, as the control computer has other tasks to perform besides the loading one, and can devote only a fraction of its time to that job.

A Grid Control Area is the largest section of the national system scheduled as a whole, and on a continuous basis. The economic scheduling program calculated an accurate dispatch for a typical GCA (test system GR142) in 1 minute and 30 seconds. In the control experiment of reference 7, a 10 minute interval between loading instructions was found to be best. The solution time to produce a network flow economic schedule is but a minor portion of the inter-instruction period. Hence the network flow method shows promise of successful application to on-line calculation of economic load dispatches.

There is no essential difference between the British and American systems at the level of the network flow model.

Generator segments in the American systems can be regarded as separate units ready to supply power to the system when it is incrementally advantageous for them to do so. In 1967 the American Electric Power (AEP) System contained 176 busses, 268 lines, and 36 steam units (15), making it the largest investor-owned electric power company in the USA. The turboalternators averaged 5 segments each, giving a total of 180 generating increments. These problem size parameters are similar to those of SG149, and therefore solution times of a few minutes would suffice for calculating dispatches. In summary then, the speed-accuracy characteristics of the network flow method permit it to schedule successfully large American electric power utilities.

Two opportunities exist for improving the solution times. As was indicated in sub-section 3.3.1, a more carefully coded solution algorithm will bring reductions in solution time of at least 50%. Second, it was observed that starting from a previous solution, instead of from the so-called 'flat start' of zero flow, could effect similar savings.

We have submitted the network flow method as a new computational tool for the solution of the economic scheduling problem. A comprehensive set of test results, their analysis, and conclusions have been put forward in support of this proposition. The thesis is completed by giving three advantages of the network flow approach to economic

scheduling.

1. It is a formal, mathematical programming model whose technique of solution ensures the finding of a global optimal solution if one exists. If no feasible solution can be found, the algorithm will indicate this fact and the partial solution will point up the reason for the infeasibility.

2. The network flow model has a simple form. It is therefore readily understood in its detail, and its implementation as a computer program is not difficult. The model can be extended, as described in Chapter 4.

3. Economic dispatches can be computed for large power systems in machine times of practical duration.

REFERENCES

1. Anstine, L.T., Henderson, J.H., Kramer, F.A., Light, F.H., Pall, G.A., Reed, F.M., Stewart, H.G., and Wetter, C.H.:'Application of digital computer techniques to the economic scheduling of marginal generating units for the Pennsylvania-New Jersey-Maryland interconnection', IEEE Transactions on Power Apparatus and Systems, 1964, vol 83, pp 316-320

2. Baldwin, C.J., Dale, K.M., and Dittrich, R.F.:'A study of the economic shutdown of generating units in daily dispatch', Transactions of the American Institute of Electrical Engineers Part III Power Apparatus and Systems, 1959, vol 78, pp 1272-1284

3. Benthall, T.P.:'Automatic load scheduling in a multi-area power system', Proceedings of the Institution of Electrical Engineers, 1968, vol 115, pp 592-596

4. Bernas, S., and Szumilin, W.:'Le critère de détermination des points économiques de la mise en service et hors service des ensembles dans les centrales thermiques', World Power Conference, Lausanne, 1964, Section IIB-1, paper 40.

5. Bernholtz, B., and Graham, L.J.:'Hydrothermal economic scheduling Part III Scheduling the thermal subsystem using constrained steepest descent', Transactions of the American Institute of Electrical Engineers Part III Power Apparatus and Systems, 1962,

vol 80, pp 1096-1105

6. Biggerstaff, B.E., and Jackson, T.M.:'The Markov process as a means of determining generating-unit state probabilities for use in spinning reserve applications', IEEE Transactions on Power Apparatus and Systems, 1969, vol 88, pp 423-430

7. Brewer, C., Charles, G.F., Parish, C.C.M., Prewett, J.N., Salthouse, D.C., and Terry, B.J.:'Performance of a predictive automatic load dispatching system', Proceedings of the Institution of Electrical Engineers, 1968, vol 115, pp 1577-1586

8. Calvert, J.F., and Sze, T.W.:'A new approach to loss minimization in electric power systems', Transactions of the American Institute of Electrical Engineers Part III Power Apparatus and Systems, vol 76, pp 1439-1446

9. Carpentier, J.:'Contribution à l'étude du dispatching économique', Bulletin Sociéte francaise des Électriciens, 8e série tome III, pp431-447

10. Castelli, F., Leardini, T., and Buzzi, E.:'Conception des centrales a vapeur, en vue de leur participation au réglage sans augmentation excessive des consommations spécifiques', World Power Conference, Lausanne, 1964, Section IIB-1, paper 9

11. Cuenod, M., Durling, A., and Pun, L.:'Optimum fitting method and its application to dynamic economic dis-

patching of power systems', International Federation of Automatic Control Congress, London, 1966, Session 21 Power Systems (Thermal), Paper 21.D

12. Decker, G.L., and Brooks, A.D.:'Valve point loading of turbines', Transactions of the American Institute of Electrical Engineers Part III Power Apparatus and Systems, 1958, vol 77, pp 481-486

13. Di Perna, A., and Ferrara, E.:'Optimum scheduling of a hydroelectric system by linear programming', Power System Computation Conference Proceedings, Stockholm, 1966, Part 4 Report 6.14

14. Dommel, H.W., and Tinney, W.F.:'Optimal power flow solution', IEEE Transactions on Power Apparatus and Systems, 1968, vol PAS-87, pp 1866-1876

15. Dopazo, J.F., Klitin, O.A., Stagg, G.W., and Watson, M.:'An optimization technique for real and reactive power allocation', Proceedings of the IEEE, 1967, vol 55, pp 1877-1885

16. El-Abiad, A.H., and Jaimes, F.J.:'A method for optimum scheduling of power and voltage magnitude', IEEE Transactions on Power Apparatus and Systems, 1969, vol 88, pp 413-422

17. Fanshel, S., and Lynes, E.S.:'Economic power generation using linear programming', IEEE Transactions on Power Apparatus and Systems, 1964, vol 83, pp 347-356

18. Farmer, E.D., James, K.W., Moran, F., and Pettit, P.: 'The development of automatic digital control of a power system from the laboratory to a field installation', Proceedings of the Institution of Electrical Engineers, 1969, vol 115, pp 436-444

19. Farmer, E.D., and Potton, M.J.:'Development of online load-prediction techniques with results from trials in the South West Region', Proceedings of the Institution of Electrical Engineers, 1968, vol 115, pp 1549-1558

20. Friedlander, G.D.:'Giant generators for growing power demand', IEEE Spectrum, 1965, vol 2, pp 70-86

21. Friedlander, G.D.:'Computer-controlled power systems Part II-Area controls and load dispatch', IEEE Spectrum, 1965, vol 2, pp 72-92

22. Garver, L.L.:'Power generation scheduling by integer programming-development of theory', Transactions of the American Institute of Electrical Engineers Part III Power Apparatus and Systems, 1963, vol 81, pp 730-735

23. Glimn, A.F., Habermann, R. Jr., Kirchmayer, L.K., and Stagg, G.W.:'Loss formulas made easy', Transactions of the American Institute of Electrical Engineers Part III, 1953, vol 72, pp 730-737

24. Glimn, A.F., Kirchmayer, L.K., Habermann R.Jr., and Thomas, R.W.:'Automatic digital computer applied to generation scheduling', Transactions of the American

Institute of Electrical Engineers Part III, 1954, vol 73, pp 1267-1275

25. Happ, H.H., Ille, W.B., and Reisinger, R.H.:'Economic system operation considering valve throttling losses I-Method of computing valve-loop heat rates on multi-valve turbines', Transactions of the American Institute of Electrical Engineers Part III Power Apparatus and Systems, 1963, vol 81, pp 609-615

26. Hara, K., Kimura, M., and Honda, N.:'A method for planning economic unit commitment and maintenance of thermal power systems', IEEE Transactions on Power Apparatus and Systems, 1966, vol PAS-85, pp 427-436

27. Hayward, A.P., Kerr, R.H., Taylor, C.E., and Kirchmayer, L.K.:'Minimization of fuel costs by the technique of linear programming', Transactions of the American Institute of Electrical Engineers Part III Power Apparatus and Systems, 1958, vol 76, pp 1288-1295

28. Hewson, J., and Oughton, A.G.:'Economic load scheduling by digital computer', Central Electricity Generating Board Computing Department, 1963, Report CD/DEV/2

29. Hill, E.F., and Stevenson, W.D.Jr.:'An improved method of determining incremental loss factor from power system admittances and voltages', IEEE Transactions on Power Apparatus and Systems, 1968, vol PAS-87, pp 1419-1425

30. Hovanessian, S.A., and Stout, T.M.:'Optimum fuel allocation in power plants', IEEE Transactions on Power Apparatus and Systems, 1963, vol 82, pp 329-335

31. Kerr, R.H., Scheidt, J.L., Fontana, A.J., and Wiley, J.K.:'Unit commitment', IEEE Transactions on Power Apparatus and Systems, 1966, vol PAS-85, pp 417-421

32. Kirchmayer, L.K.:'Economic operation of power systems' (Wiley, 1958)

33. Kirchmayer, L.K.:'Economic control of interconnected systems'(Wiley, 1959)

34. Kirchmayer, L.K., and Stagg, G.W.:'Analysis of total and incremental losses in transmission systems', Transactions of the American Institute of Electrical Engineers Part II, 1951, vol 70, pp 1197-1204

35. Knight, U.G.:'Mathematical programming methods in the planning, design, and operation of electricity supply networks',(PhD thesis, University of London, 1967)

36. Long, R.W., and Barrios, J.R.:'A parametric equations approach to the economic dispatch problem', IEEE Transactions on Power Apparatus and Systems, 1967, vol PAS-86, pp 141-149

37. Lowery, P.G.:'Generating unit commitment by dynamic programming', IEEE Transactions on Power Apparatus and Systems, 1966, vol PAS-85, pp 422-426

38. Matthewman, P.D., and Nicholson, H.:'Techniques for load prediction in the electricity supply industry',

Proceedings of the Institution of Electrical Engineers, 1968, vol 115, pp 1451-1457

39. Miller, A.R., Koen, H.R.Jr., and Deliyannides, J.S.: 'The use of power transfer equations to derive economic coordination relationships expressed as functions of voltage phase angles', Transactions of the American Institute of Electrical Engineers Part III Power Apparatus and Systems, 1959, vol 78, pp 747-754

40. Mitter, S.K.:'Function space methods in optimal control with applications to power systems', (PhD thesis, University of London, July, 1965)

41. Muckstadt, J.A., and Wilson, R.C.:'An application of mixed-integer programming duality to scheduling thermal generating systems', IEEE Transactions on Power Apparatus and Systems, 1968, vol PAS-87, pp 1968-1978

42. Nagel, T.G., and Vassell, G.S.:'Basic principles of planning VAR control on the American Electric Power System', IEEE Transactions on Power Apparatus and Systems, 1968, vol PAS-87, pp 488-495

43. Navarro, S.O., and Baker, A.D.:'The use of cubic cost curves in the economic loading of electric power systems', IEEE Transactions on Power Apparatus and Systems, 1965, vol PAS-84, pp 439-445

44. Peschon, J., Piercy, D.S., Tinney, W.F., Tveit, O.J., and Cuenod, M.:'Optimum control of reactive power flow',

IEEE Transactions on Power Apparatus and Systems, 1968, vol PAS-87, pp 40-48

45. Pettit, P.:'Digital computer programs for an experimental automatic load-dispatching system', Proceedings of the Institution of Electrical Engineers, 1968, vol 115, pp 597-605

46. Ringlee, R.J., and Williams, D.D.:'Economic system operation considering valve throttling losses II-Distribution of system loads by the method of dynamic programming', Transactions of the American Institute of Electrical Engineers Part III Power Apparatus and Systems, 1963, vol 81, pp 615-622

47. Riordon, J.S.:'Adaptive control of discrete state Markov processes',(PhD thesis, University of London, February, 1967)

48. Sasson, A.M.:'Nonlinear programming solutions for load-flow, minimum-loss, and economic dispatching problems', IEEE Transactions on Power Apparatus and Systems, 1969, vol PAS-88, pp 399-409

49. Scano, E.:'Gestion économique du démarrage des groupes thermique', Power Systems Computation Conference Proceedings, Stockholm, 1966, Part 4 Report 6.8

50. Shen, C.M., and Laughton, M.A.:'Determination of optimum power-system operating conditions under constraints', Proceedings of the Institution of Electrical Engineers,

1969, vol 116, pp 225-239

51. Smith, H.M., and Tong, Shih-Yung:'Minimizing power transmission losses by reactive-volt-ampere control', IEEE Transactions on Power Apparatus and Systems, 1963, vol 82, pp 542-544

52. Sokkappa, B.G.:'Optimum scheduling of hydrothermal sytems-a generalized approach', IEEE Transactions on Power Apparatus and Systems, 1963, vol 82, pp 97-103

53. Squires, R.B.:'Economic dispatch of generation directly from power system voltages and admittances', Transactions of the American Institute of Electrical Engineers Part III Power Apparatus and Systems, 1961, vol 79 pp 1235-1245

54. Stevenson, W.D.Jr.:'Elements of power system analysis', (McGraw-Hill, 1962)

55. Sze, T.W., Garnett, J.R., and Calvert, J.F.:'Some applications of a new approach to loss minimization in electrical utility systems', Transactions of the American Institute of Electrical Engineers Part III Power Apparatus and Systems, 1958, vol 77, pp 1577-1585

56. Tudor, J.R., and Lewis, W.A.:'Transmission losses and economy loading by the use of admittance constants', IEEE Transactions on Power Apparatus and Systems, 1963, vol 82, pp 676-683

57. Vassell, G.S., and Tibberts, N.:'An approach to the

analysis of generating-capacity reserve requirements', IEEE Transactions on Power Apparatus and Systems, 1965, vol PAS-84, pp 64-79

58. Wells, D.W.:'Method for economic secure loading of a power system', Proceedings of the Institution of Electrical Engineers, 1968, vol 115, pp 1190-1194

Program title: Multiple Arc Out-of-Kilter Network Flow Routine

Program name: MULTAR

Author: D.G. Stone

Source language: FORTRAN IV. See references 1 and 2.

Machine: IBM 7094, model II.
Core memory size is 32768_{10} locations.
Word length is 36 bits.

Monitor: IBSYS Operating System, Version 13.

Date written: 1 March 1968.

Source program length:

	text lines	FORTRAN statements
Subroutine MULTAR	50	31
Subroutine FINDST	55	43
Subroutine CPIPI	13	9
Subroutine LABLEM	81	55
Subroutine FLOCHM	36	19
Subroutine NODPRI	34	22
Subroutine NODARC	32	18
totals	301	197

Media: Cards punched in the standard 47 character code of Appendix D, reference 1.

Required routines: None

Location: Centre for Computing and Automation
Electrical Engineering Building
Imperial College
Exhibition Road
LONDON S.W. 7
U.K.

ABSTRACT

MULTAR is a set of subroutines, written in standard FORTRAN language, for solving minimal-cost network flow problems. It is an implementation of D.R. Fulkerson's 'Out-of-Kilter Method'; which is a flexible algorithm, applicable to a variety of problems. Notational agreement is maintained between the mathematical description and the computer program. Instructions for using MULTAR are presented with the aid of a complete test example. This document also serves as a programmer's reference manual, since it includes flowcharts, a variables list, details of program construction, as well as the source program listings comprising 197 FORTRAN statements. Although it was written for the IBM7094, the program is independent of system features and machine configuration.

TABLE OF CONTENTS

1 Purpose

2 Method

3 Description

- 3.1 Network
- 3.2 Path
- 3.3 Network flow
- 3.4 The Out-of-Kilter Method
  - 3.4.1 Structure of the method
  - 3.4.2 Arc states
  - 3.4.3 Path search
  - 3.4.4 Flow change
  - 3.4.5 Node price increase

4 Usage

- 4.1 Program NFTEST details
- 4.2 Further notes on MULTAR usage
- 4.3 Source program listing NFTEST
- 4.4 Run deck setup
- 4.5 Input data preparation
- 4.6 Messages
- 4.7 Output from program NFTEST

5 Example

- 5.1 Diagram, network flow example
- 5.2 Input data deck listing
- 5.3 Line printer output

6 Restrictions

7 Accuracy

8 Data checking

9 Storage

- 9.1 Core memory map

10 Execution time
11 Machine dependencies
12 Program modification
13 Checkout status
14 Program details
14.1 Subroutine MULTAR
14.2 Subroutine FINDST
14.3 Subroutine CPIPI
14.4 Subroutine LABLEM
14.5 Subroutine FLOCHM
14.6 Subroutine NODPRI
14.7 Subroutine NODARC
14.8 Flowcharts
14.8.1 Subroutine MULTAR
14.8.2 Subroutine FINDST
14.8.3 Subroutine LABLEM
14.8.4 Subroutine FLOCHM
14.8.5 Subroutine NODPRI
14.8.6 Subroutine NODARC
14.9 Variables list
14.10 Source program listing
15 References
16 Similar programs

1 Purpose

To find an optimal flow pattern in a network with upper and lower bounds on the flow in each arc.

2 Method

D.R. Fulkerson's 'Out-of-Kilter Method', exactly as presented in reference 3; except for a minor change which allows multiple parallel arcs to connect pairs of nodes.

3 Description

Graph-theoretic and network flow ideas, upon which the discussion of the algorithm and its implementation depend, are reviewed briefly in the first three subsections. The next subsection, 3.4, describes the Out-of-Kilter Method in detail. Both the terminology and notation of Fulkerson's paper are generally adhered to. Subsection 14.9 lists the program variables in alphabetical order, giving each a short description, and a notational equivalent where relevant.

3.1 Network

A network, or graph, 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 $i,j \in N$. Each arc k 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. Diagrammatically, arc k is represented as a line joining point i to point j with direction indicated by an arrowhead.

Originally the nodes of the network may have been identified with unique names or numbers. Instead, for efficient computation let them be renumbered by the first n positive integers, n being the number of nodes in the network. Then, to scan all network

nodes, index i will take the values 1,2,...,n.

In the same way, number the arcs with the first a positive integers, since there are a arcs in the network. Arc index k is independent of the node numbering and k = 1,2,...,a scans all arcs. With this indexing scheme multiple parallel arcs connecting the same two nodes i and j can be individually referenced by the arc index k. They will not have unique representations as node-pairs. The formulation of reference 3 uses node-pairs to describe arcs; not convenient for parallel-arc problems. This minor change of separate arc indexing does not render inapplicable the theoretical development or conclusions of reference 3.

3.2 Path

A path, from one node of a network through the network to another node, may be described as a continuous sequence of connected arcs linking the two nodes. Let k_m be the arc number of the mth arc in the path, and p be the number of arcs in the path. Then

$$(k_1, k_2, \ldots, k_m, \ldots, k_p)$$

denotes the path. In the present definition, the path may include arcs of both forward and reverse orientations. The diagram of subsection 5.1 can provide an example of a path. Originating at node 11, the path (1, 8, 13, 25, 29, 33) terminates on node 10. Paths in the current formulation may not contain internal loops or side spurs. A path in which the origin node coincides with the terminal node is called a cycle. In the example, (1, 8, 13, 25, 29, 33, 43) forms a cycle, or closed path.

3.3 Network flow

The steady-state flow of a single commodity through a

given network is the central concept of our discussion. Let x_k be the amount of flow from the start node i to the terminal node j of arc k = (i,j). Since the arc flow x_k has this definition, it may not reverse in arc k, and hence is a restricted variable taking non-negative values. If the arc flow x_k is regarded as a component of the overall network flow, the latter can be denoted by the vector

$$\underline{x} = (x_1, x_2, \ldots, x_k, \ldots, x_a) .$$

The flow entering a node must balance the flow leaving, a fact expressed algebraically by the conservation equations

$$\sum_{k \in A_i^-} x_k = \sum_{k \in A_i^+} x_k \qquad i = 1,2,\ldots,n$$

A_i^- is just the set of arcs impinging on node i, while A_i^+ is the set of arcs radiating from node i. Given a conservative starting flow, the solution algorithm automatically maintains conservation of flow throughout its operation. Therefore there is no need to state the flow balance constraints explicitly as part of the problem data. The pattern of conservative flow over a network may also be called a circulation.

An upper limit to the flow through each arc is set by its capacity, u_k. Similarly, a lower limit, l_k, is imposed. The flow in all arcs must obey both bounds:

$$l_k \leq x_k \leq u_k \qquad k = 1,2,\ldots,a$$

(Typographical note: 'l' used for the 12th letter and digit one.) A network flow which satisfies these inequalities and is conservative is called a feasible flow.

The cost of transporting a unit of commodity over arc k is called the arc cost, c_k. An arc flow of x_k units will cost

$c_k x_k$ to maintain. The total cost of the network flow is simply the sum of the individual arc flow costs

$$\sum_{k=1}^{a} c_k x_k \quad .$$

A feasible flow which minimizes this linear cost form is an optimal feasible flow, $\underline{x}^o$. The Out-of-Kilter Method will find a minimal-cost network flow solution, if it exists.

In establishing the optimal network flow, the algorithm develops and makes use of an additional set of variables called the node prices. The node price vector

$$\underline{\pi} = (\pi_1, \pi_2, \ldots, \pi_i, \ldots, \pi_n)$$

has as its components the price of each node in the network.

An important and desirable property of the formulation and the algorithm is that all quantities are integers.

3.4 The Out-of-Kilter Method

The mechanics of Fulkerson's algorithm are described here so as to form a basis for understanding the computer program. In turn, the algorithm depends on the theoretical development and proofs of reference 3.

The complete method segments naturally into several interrelated sub-tasks, each of which is programmed as a subroutine. These are: arc state determination, optimizing cost calculation, path searching by a labelling technique, flow changing, node price augmentation, and construction of network topology tables. Subroutine MULTAR embodies the logical skeleton of the overall method, controlling the execution of the sub-tasks.

3.4.1 Structure of the method

A valuable characteristic of the algorithm is that it may begin with any network flow $\underline{x}$, and any node price vector $\underline{\pi}$. For example, a convenient starting point is $\underline{x} = 0$, $\underline{\pi} = 0$.

There is a property associated with each arc of the network called its state. An arc is in either an 'in-kilter' state or an 'out-of-kilter' state. Basically, kilter means good condition, or order: here it 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 carried out, since the algorithm is constructed so that subsequent operations will not put this arc out of kilter. However, if the arc is in an out-of-kilter state, 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 the 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.

3.4.2 Arc states

Let $k = (s,t)$ be the arc currently being examined. It may be in any one of the nine states listed in the table below, describing the condition of the arc with respect to optimality and constraint satisfaction. 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 bearing subscript 1 and the second a 2-subscript. 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 α_2 or γ_1, where the flow is feasible but non-optimal, can be put in kilter through a node price increase.

The basic sub-task of Fulkerson's algorithm is the search for a path over which to change the flow in the out-of-kilter arc under inspection. Path searching is done by means of a node labelling procedure, as described in detail in the next sub-section. If found, the path extends from one node of the inspected arc, through the network to the other node of the arc, thus forming a cycle over which a flow change is to take place.

Suppose first, that the scanned arc, k = (s,t) is in a 1-subscript state, thus requiring an increase in flow. The origin for labelling is t, and the terminal for labelling is s. That is, the path search begins at the terminal node of arc k, and finishes when the start node of arc k is encountered. After a successful path search, the flow is increased in the resulting cycle by an amount ε. Subsection 3.4.4 describes this operation in detail: the flow in arc k is increased by ε, and the flow in the path from t to s is also increased by ε .

If arc k is in state α_1, the scanned arc flow change difference $\varepsilon_k = l_k - x_k$. In the case that the path allows this flow increase, the resulting flow will be at the scanned arc lower bound l_k. With arc k in states β_1 or γ_1 an increase that may bring the flow to its upper bound is required, so that $\varepsilon_k = u_k - x_k$.

Now consider the case of arc k = (s,t) in a 2-subscript state. The origin for labelling is s, and the terminal for labelling is t; meaning that our path search commences at the start node of the scanned arc k, and ends when its terminal node is met. If a path is established, it will thus have an orientation reversed from that of arc k. Decreasing the flow in the cycle

just found by ε is equivalent to reducing the flow in arc k by ε, and increasing the flow in the path from s to t by ε. In this way, by reversing the direction of path search, the same labelling and flow changing procedure applies to the paths associated with scanned arcs requiring an increase, or a decrease in flow.

If arc k is in state γ_2, $\varepsilon_k = x_k - u_k$, attempting to reduce the scanned arc flow to its upper limit. For states α_2 and β_2, $\varepsilon_k = x_k - l_k$; a decrease which, if it takes place, will set the arc flow on its lower bound.

Every arc of the network must be in one of the nine mutually exclusive states listed in the following table. $\bar{c}_k$ is calculated from the arc cost and the prices of the nodes of the given arc, according to:

$$\bar{c}_k = c_k + \pi_i - \pi_j \qquad k = (i,j);\ k = 1,2,\ldots,a$$

It is referred to here as an optimizing cost.

kilter	flow	arc state	optimizing cost	constraints	labelling: origin	labelling: terminal	ε_k
in kilter	correct	α	$\bar{c}_k > 0$	$x_k = l_k$			
		β	$\bar{c}_k = 0$	$l_k \leq x_k \leq u_k$			
		γ	$\bar{c}_k < 0$	$x_k = u_k$			
out of kilter	too low	α_1	$\bar{c}_k > 0$	$x_k < l_k$			$l_k - x_k$
		β_1	$\bar{c}_k = 0$	$x_k < l_k$	t	s	$u_k - x_k$
		γ_1	$\bar{c}_k < 0$	$x_k < u_k$			$u_k - x_k$
	too high	α_2	$\bar{c}_k > 0$	$x_k > l_k$			$x_k - l_k$
		β_2	$\bar{c}_k = 0$	$x_k > u_k$	s	t	$x_k - l_k$
		γ_2	$\bar{c}_k < 0$	$x_k > u_k$			$x_k - u_k$

3.4.3 Path search

The operations of flow changing and node price increasing both depend on the node labels deposited during the path search. A label is simply an integer associated with a node to convey some information. The Out-of-Kilter Method sets and uses two labels per labelled node, i; a path label (signed), $\pm k_i$, and a flow change label, ε_i; written together as $(\pm k_i, \varepsilon_i)$.

Having had the origin for labelling defined, the labelling procedure begins by giving it the label (M, ∞), where M is any positive integer. This inconsequential change from the $(0, \infty)$ label of reference 3 is dictated by programming convenience, since zero is used in the program to denote an unlabelled node, and we wish to record the fact that the origin has been given a label. Before discussing the overall progress of the labelling procedure through the network, attention is focussed on the detailed rules for labelling a previously unlabelled node, j, from a labelled node, i, over a given arc k.

One of two possibilities holds: k is either a forward arc emanating from node i and terminating on node j, or it is a reverse arc impinging on node i from node j. Because of their opposite orientations, the available flow modifications are different in each instance, and therefore the labelling rules will differ.

Examine first the case of a forward arc k = (i,j). The terminal node j will be given the label $(+k_j, \varepsilon_j)$ only if the flow in arc k can be increased in an optimal way. Specifically, arc k must be in either condition (1.a) or condition (1.b) defined below.

The path label, $+k_j$, notes that node j was reached over the forward arc k. The labelling scheme of reference 3 records the preceeding node rather than the preceeding arc. The authors of reference 4 suggest the method used here, on page 156 of that work.

A simpler computer program results, the theory of the algorithm is not affected, and multiple parallel arcs can be handled.

ε_j, the flow change label, is a record of the maximum permissible flow increase along the path up to node j. The rule for computing ε_j is included in the two cases:

(1.a) $\bar{c}_k > 0$, $x_k < l_k$, state α_1, $\varepsilon_j = \min(\varepsilon_i, l_k - x_k)$

(1.b) $\bar{c}_k \leq 0$, $x_k < u_k$, states β, β_1, γ_1, $\varepsilon_j = \min(\varepsilon_i, u_k - x_k)$

Next, consider the second possibility: k = (j,i) is an arc reversed to the direction of the path. In attempting to increase the flow over this path, we will require a flow decrease in the reverse arcs of the path. Thus the procedure will label the start node j only if the arc flow x_k can be decreased; again optimally, as specified in the two conditions (2.a), and (2.b) to follow. The node label is $(-k_j, \varepsilon_j)$. Giving the path label a minus sign is an economical way of recording the orientation of arc k with respect to the path. ε_j has the same meaning as previously, but is determined by the rule added to the cases:

(2.a) $\bar{c}_k \geq 0$, $x_k > l_k$, states β, β_2, α_2, $\varepsilon_j = \min(\varepsilon_i, x_k - l_k)$

(2.b) $\bar{c}_k < 0$, $x_k > u_k$, state γ_2, $\varepsilon_j = \min(\varepsilon_i, x_k - u_k)$

With the origin for labelling bearing a label, continued application of the labelling rules is equivalent to a methodical search for an optimal path over which to augment the flow in the scanned arc k = (s,t). If the problem is being worked by hand, the network is usually given as a diagram, and the labelling process carried forward by eye. In contrast, a computer version will hold the network topology in tabular form and a systematic procedure must be devised and followed to ensure that the labelling method

neglects no arcs during its progress. As the algorithm is independent of this machine-oriented topic, its discussion is deferred until subsection 14.4.

Each node receiving a label is checked to see if it is the terminal for labelling, and if so the labelling procedure halts. This is called breakthrough, and means that a flow-augmenting path has been found. The alternative, nonbreakthrough, occurs when at some intermediate stage in the labelling no additional labels can be assigned; and therefore the terminal for labelling cannot be reached.

3.4.4 Flow change

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

$$\varepsilon = \min(\varepsilon_T, \varepsilon_k)$$

ε_T is the flow change label on the terminal for labelling, and so 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 the flow in the path from origin to terminal for labelling by ε. The path label, $\pm k$, enables us to trace the path 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 we will find another label leading further, and so on until the origin for labelling is reached. 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. These orientations are with respect to the original path direction so for example, a forward arc will be pointing at us as we retrace it.

Last, change the flow in the scanned arc itself by ε. If arc k is in a 1-subscript state it requires a flow increase; conversely a 2-subscript state calls for a flow decrease in arc k.

3.4.5 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 L be the set of nodes labelled during the path search, and $\bar{L}$ the set of unlabelled nodes. Next, identify two sets of arcs A_1, and A_2, according to the following definitions:

$$A_1 = \{k = (i,j) \mid i \in L,\ j \in \bar{L},\ \bar{c}_k > 0,\ x_k \leq u_k\}$$

$$A_2 = \{k = (j,i) \mid i \in L,\ j \in \bar{L},\ \bar{c}_k < 0,\ x_k \geq l_k\}$$

The arcs of set A_1 radiate from L, while the arcs of set A_2 are directed into L.

Increase the prices of the unlabelled nodes $\bar{L}$ by an amount δ equal to the smallest absolute value of optimizing cost $\bar{c}_k$, of arcs defined above. More precisely, define:

$$\delta_1 = \min_{k \in A_1}(\bar{c}_k)\ ,\ A_1 \text{ non-empty} \qquad \delta_2 = \min_{k \in A_2}(-\bar{c}_k),\ A_2 \text{ non-empty}$$

$$= \infty\ ,\ A_1 \text{ empty} \qquad = \infty \quad A_2 \text{ empty}$$

and,

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

If sets A_1 and A_2 are both empty an infinite node price increase, δ, is required. Fulkerson has shown this to imply that no feasible flow exists for the given network. The node price vector cannot be adjusted, and the second terminating condition has arisen.

4 Usage

The MULTAR implementation's modular form makes it flexible in use. It is designed to be called from a program written by the user for a specific application. Such a program will have its own input data format, as well as custom-tabulation of the output data and results. Therefore, except for the termination messages, MULTAR is free of input/output statements. Communication is via the COMMON block BKKIL; both within MULTAR itself, and with the user's calling program.

NFTEST is a short, general-purpose calling program written to demonstrate the use of MULTAR and permit the running of an example. Subsection 4.3 is a FORTRAN listing of NFTEST, while the network flow example may be found in section 5.

4.1 Program NFTEST details

Program variables are the same as those of the MULTAR subroutines with the addition of:

II	I	node index
KK	I	arc index
PRICES	I	number of non-zero node prices read in
PRTITL	AV	alphabetic array holding the problem title

Pre-solution initialization required by MULTAR is best described by referring to program NFTEST. As with subroutine arguments, certain of the variables in COMMON block BKKIL must be given values at the outset. These are: L,U,C,X,S,T,PI,ARCS, and

NODES. Zero is a convenient initial value for the flow vector X and the node price vector PI. Statement number 8, a call to subroutine NODARC, causes the node-to-signed arc table to be built up in arrays KS and NODLOC. Finally, the optimizing cost array CBAR is initialized by calling subroutine CPIPI.

Following the initialization, a single call to subroutine MULTAR completes the solution without user intervention. The flow vector X will contain this new solution, and array PI the corresponding node prices. Whether the solution is feasible or not, the output section tabulates the problem data and results, first arc by arc, then node by node. If no feasible solution can be found, examination of arrays X and PI should help the user to determine the cause of the infeasibility.

4.2 Further notes on MULTAR usage

The inherent feature of the Out-of-Kilter Method that it may start with any flow vector and any node price vector makes it quite flexible in use. For example, if the user wishes to examine the effect of changes in arc costs or flow bounds on the solution, he may issue repeated calls to subroutine MULTAR. The previous solution is used as the starting point for the next one; incidentally achieving significant economies in computing time. Between solutions there is no need to call subroutines NODARC and CPIPI. The user may wish to monitor the performance of the algorithm through the variables TBT, TNBT, LDFN, and FECIRC. FECIRC is an integer equivalent of the termination messages, and may be employed to control the user's calling program.

The example, section 5, has several optimal solutions. MULTAR has no facility for finding more than one optimal solution. The solution it does find may depend on the starting values.

Since the network flow algorithm operates with integer quantities only, variables in the original problem with significant

fractional parts must be scaled up until the least significant digit in the fractional part occupies the units position in the network flow integer. If the user's machine word length is short, the largest available FORTRAN integer will be correspondingly small. Then he must be careful not to scale the arc bounds, costs and flows higher than necessary, so as to avoid overflow in the node prices.

A variety of transportation and network flow problems can be expressed in a form suitable for solution by the Out-of-Kilter Method. The following supplementary points may assist in such an undertaking:

1 An arc lower bound can be set to zero. L(K) = 0.

2 To fix the flow in an arc, its upper bound can be set equal to its lower bound. U(K) = L(K).

3 Both bounds may be set to zero, stopping flow in that arc. L(K) = U(K) = 0 .

4 An unbounded arc can be simulated by giving a large value to its upper bound. U(K) = 99999

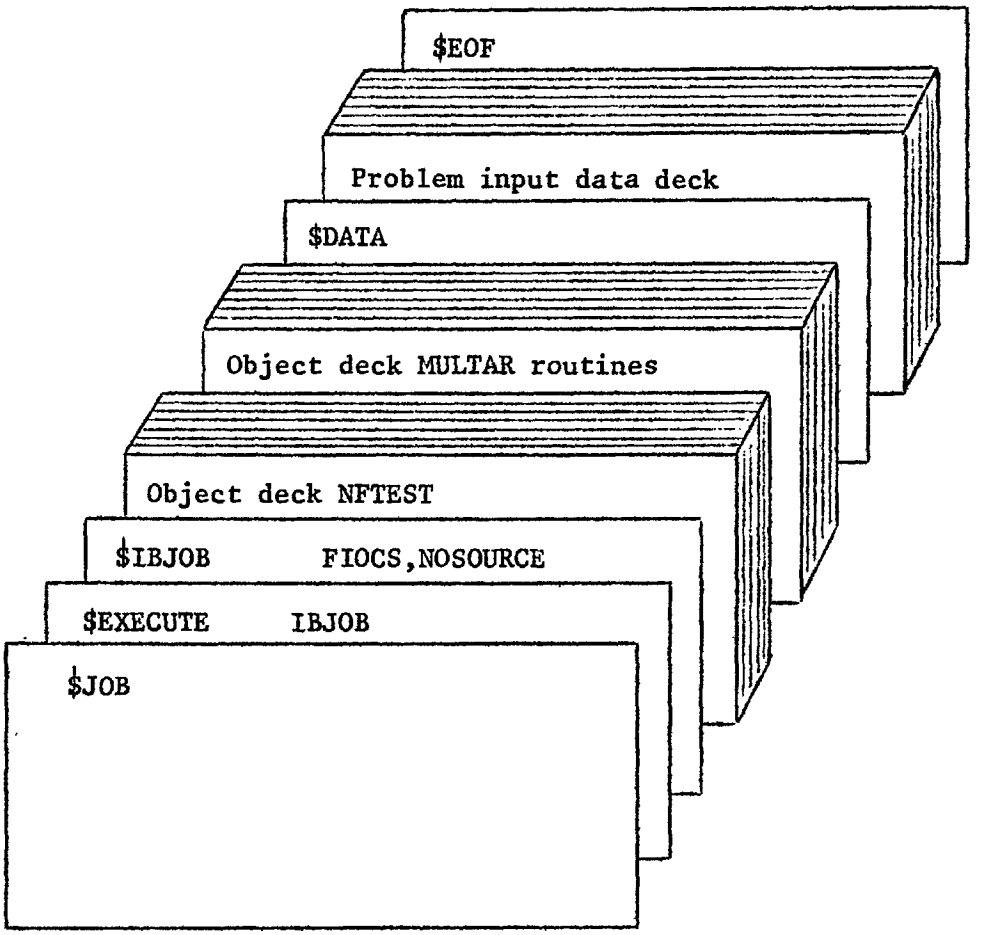
5 Arc costs may be negative, zero, or positive.

4.3 Source Program listing

```
C       NFTEST
C       PROGRAM TO TEST NETWORK FLOW ROUTINE MULTAR
C       MAXIMUM PROBLEM SIZE FOR PRESENT DIMENSIONS
C            ARCS  1200      NODES  400
        INTEGER ARCS,S,T,U,C,X,PI,PRICES
C       ALPHABETIC PRTITL
        DIMENSION PRTITL(10)
        COMMON /BKKIL/ L(1200),U(1200),C(1200),X(1200),
       1CBAR(1200),S(1200),T(1200),PI(400),KLABEL(400),
       2ELABEL(400),KS(2400),NODLOC(401),DIF,FECIRC,ARCS,NODES,
       3STAIND,STADES,K,NEXT,OFL,TFL,INFINY,TBT,TNBT,LDFN,STAGES
C            READ AND WRITE THE PROBLEM TITLE
        READ(5,901) PRTITL
  901   FORMAT(10A6)
        WRITE(6,906) PRTITL
  906   FORMAT(1H1,10A6)
C            INPUT THE NETWORK SIZE
        READ(5,905) ARCS,NODES
  905   FORMAT(2I5)
C            INPUT THE NETWORK DATA, ARC BY ARC
        READ(5,900)(K,S(K),T(K),L(K),U(K),C(K),X(K),KK=1,ARCS)
  900   FORMAT(7I5)
C            ZERO THE NODE PRICE ARRAY
        DO 800 I=1,NODES
  800   PI(I)=0
C            INPUT THE NON-ZERO NODE PRICES
        READ(5,905) PRICES
        IF(PRICES.EQ.0) GO TO 8
        READ(5,905)(I,PI(I),II=1,PRICES)
C            INITIALIZATION
    8   CALL NODARC
        CALL CPIPI
C            NETWORK FLOW ROUTINE ATTEMPTS TO FIND
C            AN OPTIMAL, FEASIBLE SOLUTION TO GIVEN PROBLEM
        CALL MULTAR
C            OUTPUT THE PROBLEM DATA AND SOLUTION
        WRITE(6,902) ARCS,NODES
  902   FORMAT( 8H ARCS = I4,3X,8HNODES = I4/)
        WRITE(6,903)
  903   FORMAT(2X,3HARC,4X,1HS,4X,1HT,4X,1HL,4X,1HU,4X,1HC,
       14X,1HX)
        WRITE(6,900) (K,S(K),T(K),L(K),U(K),C(K),X(K),K=1,ARCS)
        WRITE(6,904)
  904   FORMAT(/5H NODE,3X,2HPI)
        WRITE(6,905) (I,PI(I),I=1,NODES)
        STOP
        END
```

4.4 Run deck setup

Having obtained binary object program card decks from a compilation run, the user sets up his normal run deck as in the diagram. Since the run requirements are standard, no extra IBSYS monitor features are used and the run deck contains a minimal number of control cards.



4.5 Input data preparation

In a typical applications program, the assembly of the network flow model data from more basic problem data can be a relatively complicated operation. However in the case of NFTEST, our short calling program, the network flow data is read in directly from cards.
Section 5.2 lists the input data deck for the example.

The first card is a title for the problem to be run, and consists of any characters punched in columns 1 to 60. All subsequent data is punched into integer fields 5 columns wide. The second card defines the network size by giving, first the number of arcs, and then the number of nodes. On each of the following cards is punched the data for one arc, thus specifying the network topology, bounds, arc costs, and starting flow.

For a network of a arcs and n nodes, the arcs must be numbered with the first a positive integers, and the nodes with the first n. If the network elements are otherwise numbered or named in the user's applications program, he must arrange for a renumbering before entry to the MULTAR routines. The consecutive numbering scheme of MULTAR is not restricted by any additional rule.

The final card holds a zero to indicate that no non-zero node prices are to be read in for this run. Otherwise, this card contains the number of node prices to be read in, and is followed by the node data cards punched according to FORMAT 905.

4.6 Messages

MULTAR does not output any messages while searching for a solution. See section 12 Program modification. However, just before returning to the user's program, MULTAR generates one of two termination messages. Refer to the listing of subroutine MULTAR. If the algorithm was successful in finding an optimal

solution, the message in FORMAT statement number 5 will be given. Conversely, if no feasible flow could be established, MULTAR prints the message of statement number 4. In both cases, statement 6 informs the user of the number of path searches undertaken by the algorithm in its hunt for an optimal solution.

4.7 Output from program NFTEST

Section 5.3 is a copy of the line printer output for the example. The tabulation of problem data and results produced by NFTEST is particularly simple, following closely the input data format.

Line 1 identifies the output by the problem title, printed exactly as read in. Lines 2 and 3 are the termination message from subroutine MULTAR, indicating success in finding an optimal solution, and informing the user that a total of 24 path searches were made in finding the solution. Line 4 contains the problem size, and line 5 is a heading for the arc data and results. These are tabulated, one arc per print line, arcs in numerical order. The new network flow solution appears under the letter X. Node prices follow the arc quantities and are copied here as an insert.

5 Example

To demonstrate the operation of MULTAR and NFTEST the example of page 123, reference 4 was selected. Subsection 5.1 is a diagram of this network. Arc bounds and costs are tabulated in subsection 5.2. Node 11 may be regarded as a source node, and node 10 as a sink. Flow conservation is maintained by the return arc, 43 = (10,11). The problem is to establish a maximal flow from the source node to the sink node, that at the same time minimizes the cost of flow from source to sink.

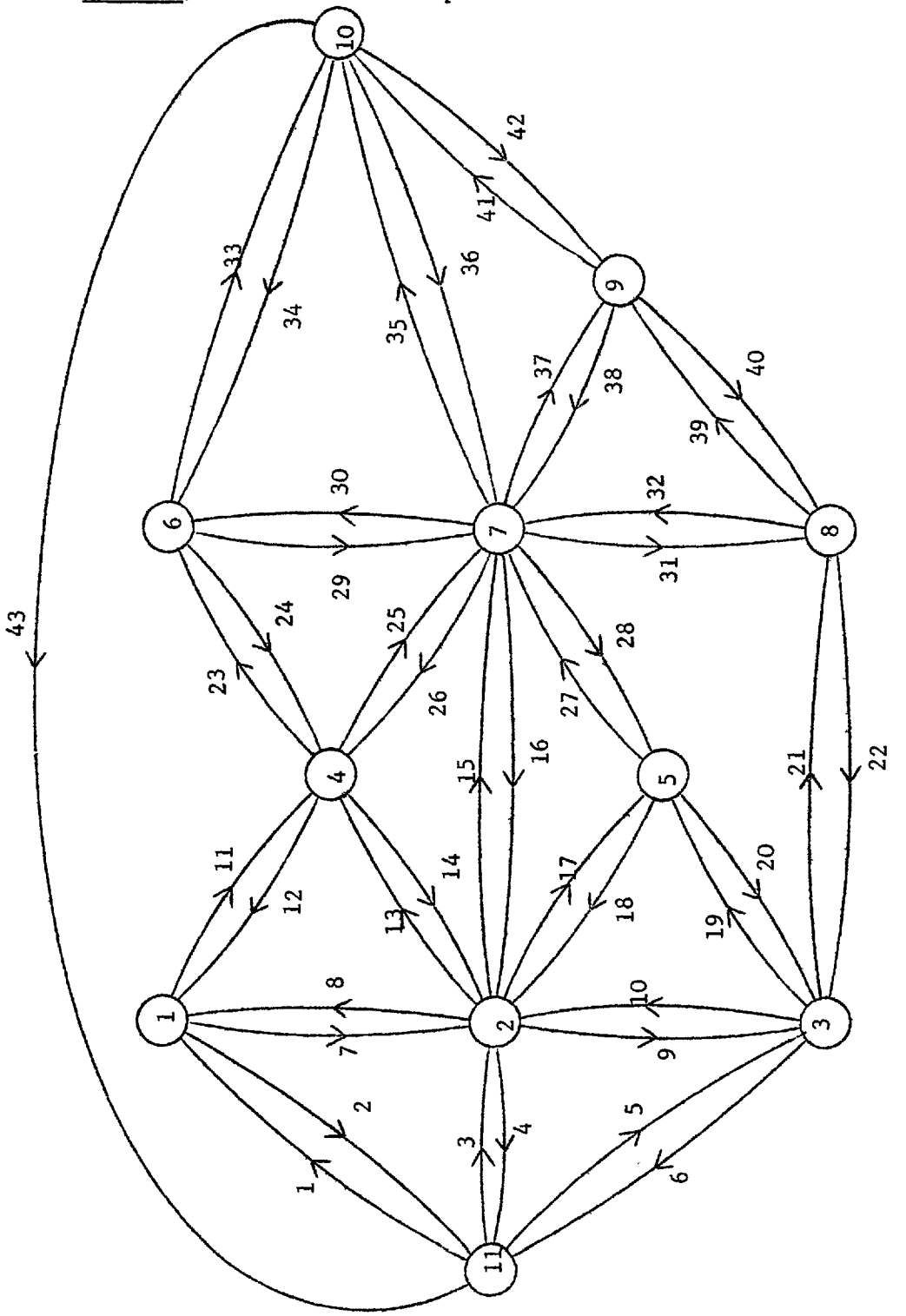
Each undirected arc in the original network has been replaced by a pair of opposed, directed arcs, to conform to the present

formulation. The return arc has been given a large negative cost. Since the algorithm is a minimizing one, it will maximize the flow in the return arc, and hence the flow through the network from source to sink. The positive costs on the network arcs will cause this flow to distribute itself so that the cost of the maximal flow is a minimum.

The maximal flow is 85 units. See subsection 5.3. There is more than one minimal-cost flow pattern, however. For example, two equal-cost paths exist from node 8 to the sink; namely 32,35 and 39,41. The solution of reference 4 has 5 flow units switched from the first path to the second.

This network flow example fails to illustrate two important features of the Out-of-Kilter Method: non-zero lower bounds, and multiple parallel arcs of the same orientation joining two nodes.

5.1 Diagram, Network flow example



5.2 Input data deck listing

```
MINIMAL COST MAXIMAL FLOW EXAMPLE. P123. FLOWS IN NETWORKS
   43   11
    1   11    1    0   50    3    0
    2    1   11    0   50    3    0
    3   11    2    0   30    6    0
    4    2   11    0   30    6    0
    5   11    3    0   15    8    0
    6    3   11    0   15    8    0
    7    1    2    0   50    2    0
    8    2    1    0   50    2    0
    9    2    3    0   15    2    0
   10    3    2    0   15    2    0
   11    1    4    0   25    2    0
   12    4    1    0   25    2    0
   13    2    4    0   45    1    0
   14    4    2    0   45    1    0
   15    2    7    0   15    8    0
   16    7    2    0   15    8    0
   17    2    5    0   10    3    0
   18    5    2    0   10    3    0
   19    3    5    0   10    1    0
   20    5    3    0   10    1    0
   21    3    8    0   20    3    0
   22    8    3    0   20    3    0
   23    4    6    0   90    9    0
   24    6    4    0   90    9    0
   25    4    7    0   10    8    0
   26    7    4    0   10    8    0
   27    5    7    0   60    5    0
   28    7    5    0   60    5    0
   29    6    7    0   10    1    0
   30    7    6    0   10    1    0
   31    7    8    0   20    2    0
   32    8    7    0   20    2    0
   33    6   10    0   10    2    0
   34   10    6    0   10    2    0
   35    7   10    0   80    4    0
   36   10    7    0   80    4    0
   37    7    9    0   10    1    0
   38    9    7    0   10    1    0
   39    8    9    0   10    3    0
   40    9    8    0   10    3    0
   41    9   10    0   10    3    0
   42   10    9    0   10    3    0
   43   10   11    0  555 -100    0
    0
```

5.3. Line printer output

MINIMAL COST MAXIMAL FLOW EXAMPLE, P123, FLOWS IN NETWORKS
ALGORITHM TERMINATES WITH FEASIBLE, OPTIMAL SOLUTION.
BREAKTHROUGHS 10 NONBREAKTHROUGHS 14
ARCS = 43 NODES = 11

ARC	S	T	L	U	C	X
1	11	1	0	50	3	50
2	1	11	0	50	3	0
3	11	2	0	30	6	20
4	2	11	0	30	6	0
5	11	3	0	15	8	15
6	3	11	0	15	8	0
7	1	2	0	50	2	25
8	2	1	0	50	2	0
9	2	3	0	15	2	15
10	3	2	0	15	2	0
11	1	4	0	25	2	25
12	4	1	0	25	2	0
13	2	4	0	45	1	5
14	4	2	0	45	1	0
15	2	7	0	15	8	15
16	7	2	0	15	8	0
17	2	5	0	10	3	10
18	5	2	0	10	3	0
19	3	5	0	10	1	10
20	5	3	0	10	1	0
21	3	8	0	20	3	20
22	8	3	0	20	3	0
23	4	6	0	90	9	20
24	6	4	0	90	9	0
25	4	7	0	10	8	10
26	7	4	0	10	8	0
27	5	7	0	60	5	20
28	7	5	0	60	5	0
29	6	7	0	10	1	10
30	7	6	0	10	1	0
31	7	8	0	20	2	0
32	8	7	0	20	2	20
33	6	10	0	10	2	10
34	10	6	0	10	2	0
35	7	10	0	80	4	75
36	10	7	0	80	4	0
37	7	9	0	10	1	0
38	9	7	0	10	1	0
39	8	9	0	10	3	0
40	9	8	0	10	3	0
41	9	10	0	10	3	0
42	10	9	0	10	3	0
43	10	11	0	555	-100	85

NODE	PI
1	4
2	6
3	90
4	7
5	91
6	16
7	96
8	94
9	97
10	100
11	0

6 Restrictions

1 Dimensioning of the FORTRAN arrays sets the maximum problem size at 1200 arcs and 400 nodes. Section 9 Storage describes adjustments to the maximum number of nodes and arcs.

2 Nodes and arcs must be numbered consecutively as specified in subsection 4.5.

3 The magnitude of the program variables is restricted only by the word length of the computer. See sections 4.2 and 11.

7 Accuracy

As all quantities are integers, and the sole arithmetic operations performed are addition and subtraction; there is no mechanism whereby numerical errors can develop. Solutions are exact to the least significant digit, the units digit.

8 Data checking

Checks on the validity of the problem data are not done by MULTAR, or by NFTEST. However, in his applications program, the user may wish to check for:

1 Unusual network topology: isolated nodes or arcs, dead-end spurs, a split network.
2 Maximum problem size exceeded.
3 Arc lower bound L(K) greater than the upper bound U(K).
4 Initial flow X non-conservative.
5 Overflow in the node prices PI.

9 Storage

Subsection 9.1 Core memory map presents the program storage requirements in diagrammatic form. It is drawn for the given maximum problem size of 1200 arcs and 400 nodes, and for

standard operation under the IBSYS monitor. Note that the named COMMON block BKKIL resides with program NFTEST.

The MULTAR subroutines are shown as occupying 1651_{10} locations. However, one of the routines, LABLEM, contains a large local array, NPSTAG, which is used as working storage. It occupies 800_{10} locations, twice the maximum number of nodes in the network. Not including array NPSTAG then, the MULTAR routines and their remaining local variables take 851_{10} words of core memory.

In COMMON block BKKIL, variables L,U,C,X,CBAR,S, and T have dimension a_{max}, the maximum number of arcs. KS is dimensioned $2a_{max}$. The node variables PI, KLABEL, and ELABEL have n_{max} locations each, while NODLOC is dimensioned $n_{max}+1$. The number of locations in BKKIL, loc, is related to the maximum problem size by:

$$loc = 9a_{max} + 4n_{max} + 16$$

The user may change the maximum problem size specification by altering array dimensions in BKKIL and array NPSTAG. Suppose for example, an estimate of the maximum number of arcs permitted in a 700-node problem is required. Assuming a minimum input/output buffer size of 512_{10} locations, the maximum number of locations available to BKKIL and NPSTAG is

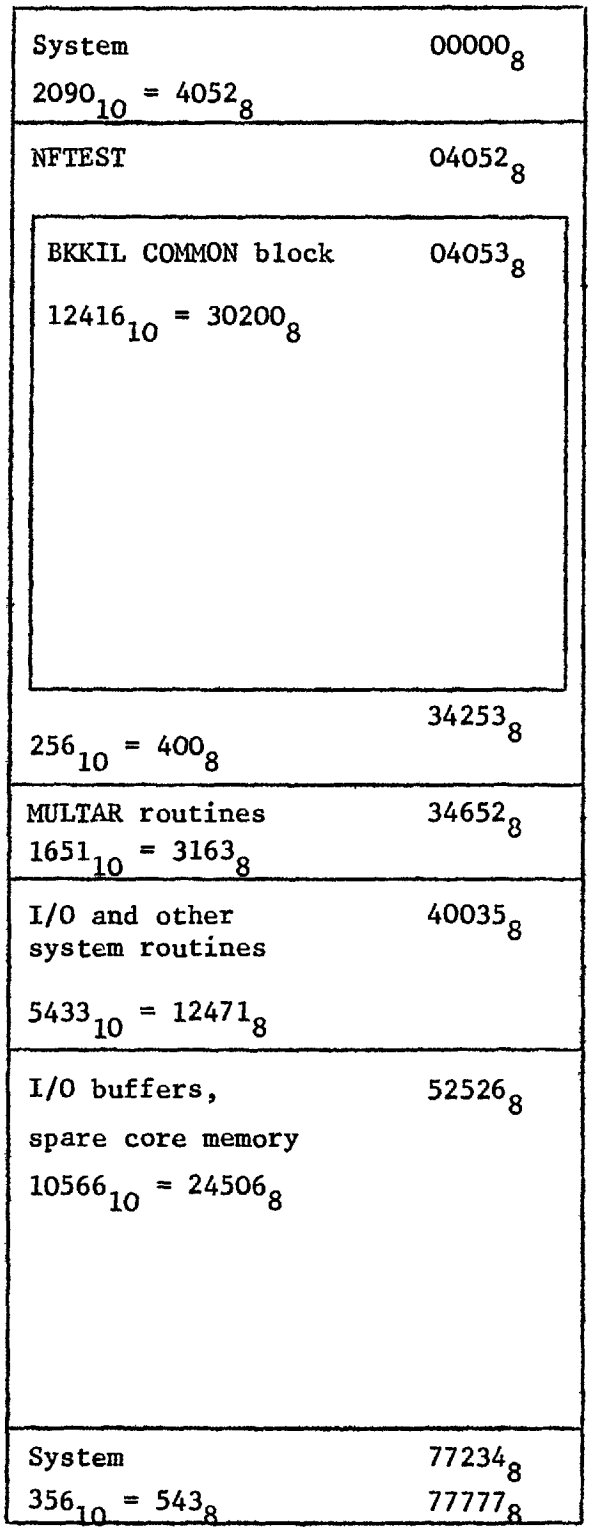
$$\begin{array}{r} 12416 \\ 800 \\ 10566 \\ \underline{-512} \\ 23270_{10} \end{array} \quad \text{locations.}$$

The choice of maximum problem size is then governed by the equation

$$23{,}254 = 9a_{max} + 6n_{max}$$

which, solved for the above requirement yields $a_{max} = 2117$ arcs.

9.1 Core memory map



10 Execution time

The computing time spent by the MULTAR subroutines in establishing an optimal flow pattern depends on the following factors:

1 Size of the network.
2 Type of optimizing problem.
3 Machine speed.
4 Nearness of the starting flow to the final flow.
5 Arc numbering, since that determines their order of examination.

However, the time taken to obtain various solutions to a given network appears not to vary in an unexpected manner.

A particular network comprising 644 arcs and 151 nodes required 1 minute and 20 seconds of machine time to solve, excluding input/output and topology table initialization via subroutine NODARC. The IBM7094 model II has a basic machine cycle time of about 1.4 microseconds, and fixed point add/subtract takes two such cycles.

11 Machine dependencies

As the word length of the IBM7094 model II is 36 bits, the largest integer that can be held in a single word is $2^{36}-1$. The variable INFINY is given this value in subroutine MULTAR, to represent the infinity of the algorithm. Implementing MULTAR on a different machine, the user should assign to INFINY a value equal to the largest FORTRAN integer representable in the precision he has specified.

In subroutine FINDST, the DATA statement and the variable STADES contain alphameric arc state descriptions of from four to six characters each. Six is the maximum number of alphameric characters per FORTRAN word for the present machine.

The FORTRAN specification statements in the preamble of each subroutine have been placed in the safest possible order:

INTEGER
DIMENSION
COMMON

to accommodate some compilers.

A difficulty will arise if a different number of storage units (machine words, bytes) are specified per FORTRAN integer variable, than are specified per real variable. Since only the variables used in any particular subroutine of the MULTAR group have an explicit INTEGER type assignment, the remainder will take the implicit type assignment REAL. Because of the different word lengths, this will throw the variables in COMMON block BKKIL out of alignment from subroutine to subroutine. The difficulty can be overcome by assigning, with the exception of STADES, implicitly REAL variables in BKKIL as type INTEGER, in all routines.

READ and WRITE statements in MULTAR and NFTEST use the FORTRAN input/output unit assignments standard under the IBSYS monitor. They are: logical unit 5 for BCD data input and 6 for BCD data output. Messages given by MULTAR occupy no more than 54 horizontal print positions. The problem title printed by NFTEST may be up to 60 characters long, however. Vertical spacing control characters are standard, conforming to references 1 and 2. Print lines written by MULTAR begin with a blank, for normal one-line spacing.

The dependence of the maximum problem size specification on machine core memory size can be inferred from the material given in section 9 Storage.

Two standard built-in (intrinsic) functions of the FORTRAN language, IABS and MINO, are referenced by the subroutines LABLEM, FLOCHM, and NODPRI.

12 Program modification

1 The maximum problem size may be reset by changing the array dimensions as indicated by section 9 Storage.

2 The messages in subroutine MULTAR may be removed. Number the RETURN statement 6.

3 The alphameric variables in subroutine FINDST are descriptive only and are not involved in program operation. Their removal would effect slight economies in storage and speed.

4 Similarly, the program operating statistics TBT, TNBT, and LDFN, and their associated statements may be withdrawn from subroutines MULTAR and LABLEM.

5 To observe the arc state transitions, insert a WRITE statement after the call to FINDST in subroutine MULTAR. Print out K and STADES.

13 Checkout status

Every detail of internal operation was observed and checked for a 4 node, 6 arc test problem. From the initial date of writing to the present time, no program errors have been found. No program failure has been experienced. Therefore there are no suspected errors remaining in the program.

However the MULTAR subroutine set cannot be considered to have full certification for the following reasons:

1 It has had extensive use on a single problem type only.

2 The sole implementation to date is for the IBM7094 model II.

3 No attempt has been made to make the program carry out all possible state transitions.

4 It has been used only by the author of the program.

14 Program details

Many comment cards are included with the source program. This section aims at relating the FORTRAN program to the description of the algorithm given in Section 3. Some program details appear in sections 4.6, 9, 11, and 12. The short calling program NFTEST is discussed separately, in subsection 4.1.

14.1 Subroutine MULTAR

The name MULTAR is used in this documentation to describe both the collection of network flow routines taken together, and the individual main subroutine forming the framework of the algorithm. FECIRC is initially set to -1 as a signal of incomplete algorithm operation, if the program has been prematurely terminated by an external interruption.

One of the results of the call to subroutine FINDST is that the variable STAIND is given a value as follows:

STAIND	arc K
0	in kilter
1	out-of-kilter, 1-subscript
2	out-of-kilter, 2-subscript

via the COMMON block. STAIND is also used in subroutine FLOCHM.

If variable NEXT is zero on return from subroutine LABLEM a nonbreakthrough has occurred. NEXT serves here as a breakthrough indicator, a positive integer marking a successful path search.

Subroutine NODPRI will set FECIRC to zero if it discovers that the problem in hand has no feasible solution, and algorithm operation will end. Otherwise, array CBAR is reevaluated by subroutine CPIPI, working from the new node prices. Control transfers to statement 1 after the calls to FLOCHM and NODPRI, since the scanned arc must be checked to find if these operations have

yet placed it in an in-kilter condition.

The three IF-statements in subroutine MULTAR are in a sense redundant, since they test conditions already known by the other routines. They arise from the modular form of program construction, and a desire to avoid the non-standard RETURN feature available on the author's machine.

When DO-loop 100 is satisfied, all arcs have been put in kilter and array X now holds the optimal solution. FECIRC takes the value 1 to signify this fact.

Subroutine MULTAR is the only routine in the program that calls on other routines. It contains calls to all of them except subroutine NODARC.

14.2 Subroutine FINDST

Besides finding the present state of K, the arc currently being inspected; this routine does related work. First the arc differences are precomputed. The scanned arc flow change difference, DIF, will take one of these values depending on which state the arc is in. Five arithmetic IF statements comprise the program logic realizing the arc state table of subsection 3.4.2. The first IF, by testing the value of CBAR, detects whether arc K is in the ALPHA-group, BETA-group, or GAMMA-group of states. Arithmetic IF statement 1 deals with the ALPHA states, 3 the GAMMA states; while the BETA states require two IF statements, namely 2 and 4.

The DATA statement holds alphameric arc state descriptors. When the state of arc K is ascertained, the appropriate descriptor is assigned to variable STADES for possible observation by the user. In addition, STAIND is given a numerical value, as set out in the previous subsection. Lastly, by assigning the origin and terminal for labelling, FINDST orients the path search according to the requirement of arc K for an increase or decrease in flow.

14.3 Subroutine CPIPI

Following a node price increase, new values of the arc optimizing costs must be computed for use in the next path search. DO-loop 100 re-computes each element in array CBAR whether it has changed value or not. Node indicies SKJ and TKJ circumvent a FORTRAN limitation on subscripts. Rather than embedding the evaluation of CBAR in subroutine NODPRI, it was written as an independent subroutine to facilitate initialization of CBAR.

14.4 Subroutine LABLEM

The search for a flow-augmenting path through the network is carried out by a node labelling procedure which progresses in stages across the network. This staging mechanism is described in reference 6, page 340. It consists of maintaining two lists of labelled nodes; the present stage nodes, from which labelling is being done, and the next stage nodes, which are nodes that have just been labelled from the present stage. The two-dimensional array NPSTAG stores the next and present stage nodes in two column vectors.

Initially there is just one node in the present stage, the origin for labelling, OFL. MP and MN are pointers to the column vectors of array NPSTAG that hold the present and next stages, respectively. The routine starts off by putting the OFL in position one of the first column vector, and reserving the second column vector for storing the next stage nodes.

NEXT is a count of the number of nodes in the next stage, that is, the number of nodes that have been given labels from the nodes in the present stage. At the beginning of each stage NEXT is set equal to zero.

IPS, the present stage index, references in turn each of the node numbers of the present stage, extracting them, with the

aid of MP, from array NPSTAG to give the node index IL a value. Next, the number of arcs connected to node IL is computed by subtracting the two successive entries in array NODLOC. Then the inner, nested DO-loop 100 can scan quickly all the arcs connected to node IL. LSA locates the signed arc in array KS, by being initialized by NODLOC(IL) and incremented by ISCAN. The arc index KL is used as a FORTRAN subscript, and so must be obtained from the element in array KS.

The IF statement determines whether arc KL radiates from or impinges on node IL, transfering 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 unlabelled, the next IF statements make an attempt to do so according to the labelling rules of subsection 3.4.3. Program control will transfer to statement 9 if JL can be labelled, and to statement 100 if it cannot. The assignment of the path and flow change labels is exactly as described in subsection 3.4.3.

After labelling JL, the count of labelled nodes in the next stage NEXT is incremented, and the node number JL is stored at location NEXT in the next stage side MN of the array NPSTAG. The test for JL equals TFL is equivalent to examining TFL for a label.

When the outer DO-loop 100 has been satisfied, all nodes in the present stage have been employed in an attempt to label new nodes for the next stage. The value of NEXT will still be zero if no new nodes could be labelled, and the labelling process terminates in the nonbreakthrough condition.

Usually however, NEXT will contain a positive count and the staging mechanism then progresses by regarding what has been called the next stage, as the present stage. This is done simply by setting node count PRES equal to NEXT, and interchanging pointers MP and MN via the dummy location MD.

14.5 Subroutine FLOCHM

The amount of flow change, EPSILN, is a positive integer computed by means of the FORTRAN function MINO which takes the value of the smallest of its arguments. As the path is retraced from the terminal for labelling, index IA will record the number of the present node.

Although the path label KLABEL (IA) identifies the next arc of the path, statement 1 is necessary for two reasons. The corresponding arc index KA cannot be a subscripted variable, or negative, as it is used later as a subscript itself. The next statement finds the orientation of the traced arc KA with respect to the path, by testing the sign of the path label. The flow in the arc is changed by EPSILN and the arc passed over by making the present node IA the node on the other end of the arc.

Statement 3 is in effect the terminator of the main path tracing loop. Exit from the loop occurs when the present node is the origin for labelling. The path flow change is complete, and it remains to change the flow in the scanned arc. The state indicator STAIND, set by subroutine FINDST, will recall for subroutine FLOCHM whether an increase or a decrease was to be made.

14.6 Subroutine NODPRI

DO-loop 100 references the arcs of the network in turn, inspecting each for membership in set A1 or set A2 by means of two logical IF statements. The program does not maintain tables corresponding to the sets of labelled and unlabelled nodes. Rather, the convention is adopted that a path label of value zero identifies an unlabelled node, while a labelled node will naturally have a non-zero path label.

DELTA, the node price increase, is computed by keeping its current minimum value in the variable DELTA, and comparing that

with a possibly lower absolute value of CBAR, using the MINO function in statements 1 and 2. When DO 100 is satisfied, DELTA will contain the required node price increase.

At the beginning of the subroutine, DELTA is given the value INFINY. If it still has this value on exit from DO 100, sets A1 and A2 must both be empty, and the no-feasible-circulation terminating condition has been detected. FECIRC is set equal to zero, notifying subroutine MULTAR of this event.

In the usual case where DELTA is less than INFINY and an increase is possible in the prices of unlabelled nodes, these are identified as before, by examining the path labels. DO-loop 200 scans all nodes, but only the unlabelled ones undergo a price increase.

14.7 Subroutine NODARC

Machine implementation of the network topology is by tables stored in FORTRAN arrays in the COMMON block. The network connections are completely and conveniently specified by tabulating the start node S(K) and the terminal node T(K) of each arc K. The two arrays S and T form the arc-to-node table, permitting high-speed access to the nodes of a specific arc. All the subroutines except MULTAR use the arc-to-node table.

However, the labelling procedure requires a second topology-referencing operation: scan all arcs connected to a given node. A machine search of the entire arc-to-node table would be possible, but slow. Since the program execution time depends chiefly on the speed of the labelling process; an efficient scheme for locating arcs must be devised. In carrying out the labelling process manually from a diagram, the arcs connected to a given node are immediately to hand. This idea is used as the basis for a second description of the network topology, the node-to-signed-arc table.

In this table, all arcs connected to a particular node are collected together in successive entries.

The signed arc array, KS, and the node location array, NODLOC, hold the node-to-signed-arc table. Subroutine NODARC relieves the user of the burden of assembling this table. A means for referencing the set of arcs, tabulated in array KS, that are connected to a given node IM, is provided by the node location array NODLOC. Beginning with the set of arcs connected to node 1, DO-loop 200 builds up array KS by scanning all the arcs of the network. Arcs radiating from node 1 are signed positive, arcs impinging negative, and arcs not connected to node 1 are ignored. LSA is a count of the total number of arcs added to the table in array KS. When DO 200 is satisfied, all the arcs connected to node IM have been tabulated, and a search for arcs connected to the next node commences. On completion of DO 100 the table is complete, except for a final entry in array NODLOC, which defines the number of arcs connected to the highest numbered node.

Subroutine NODARC is part of the MULTAR set of routines, but is not called by any of them. See subsection 4.1.

Here is a partial listing of the example network's topology tables:

Arc-to-node table

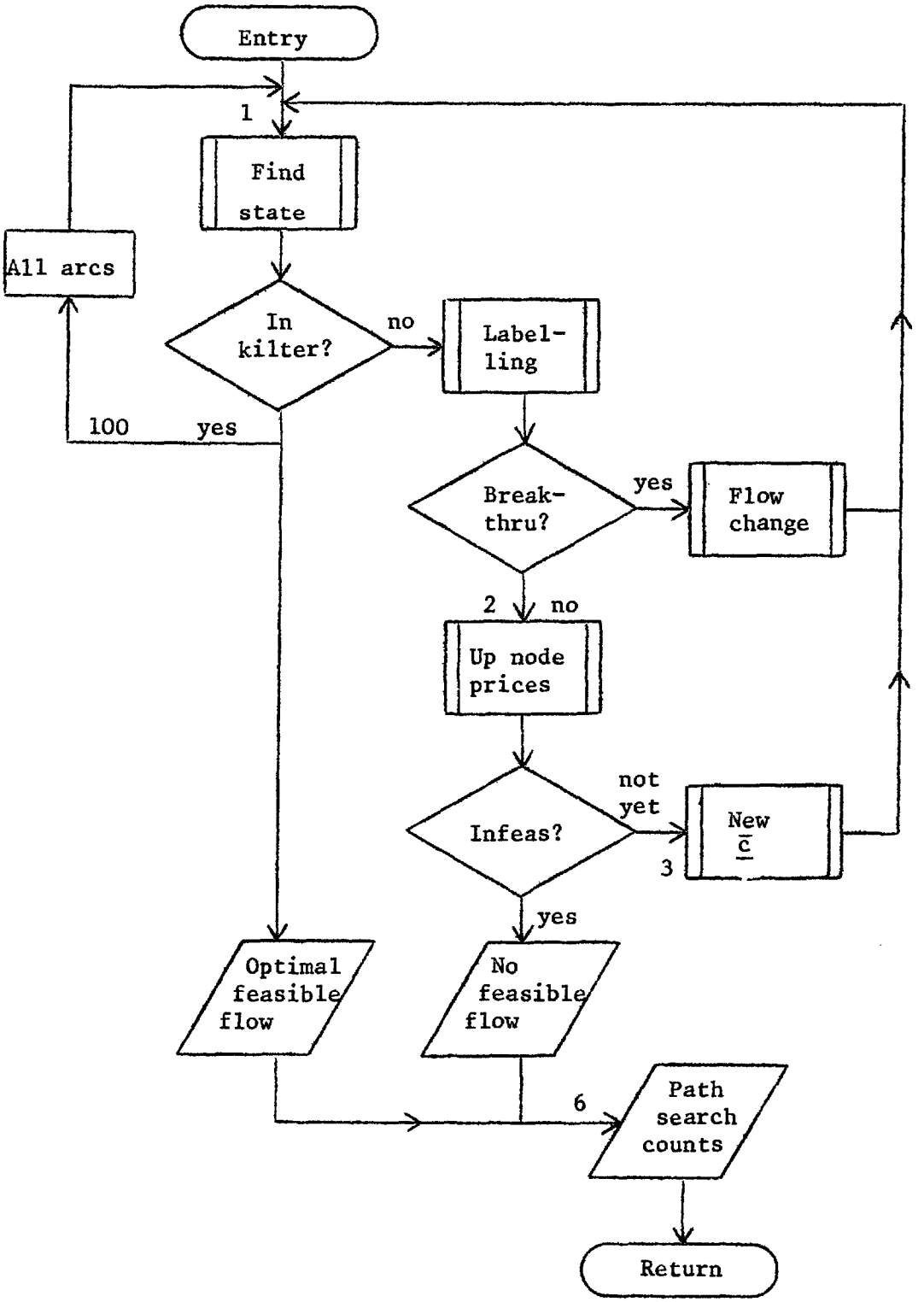
arc	start node	terminal node
K	S(K)	T(K)
1	11	1
2	1	11
3	11	2
4	2	11
5	11	3
6	3	11
7	1	2
8	2	1
9	2	3
.	.	.
.	.	.
.	.	.
41	9	10
42	10	9
43	10	11

Node-to-signed arc table

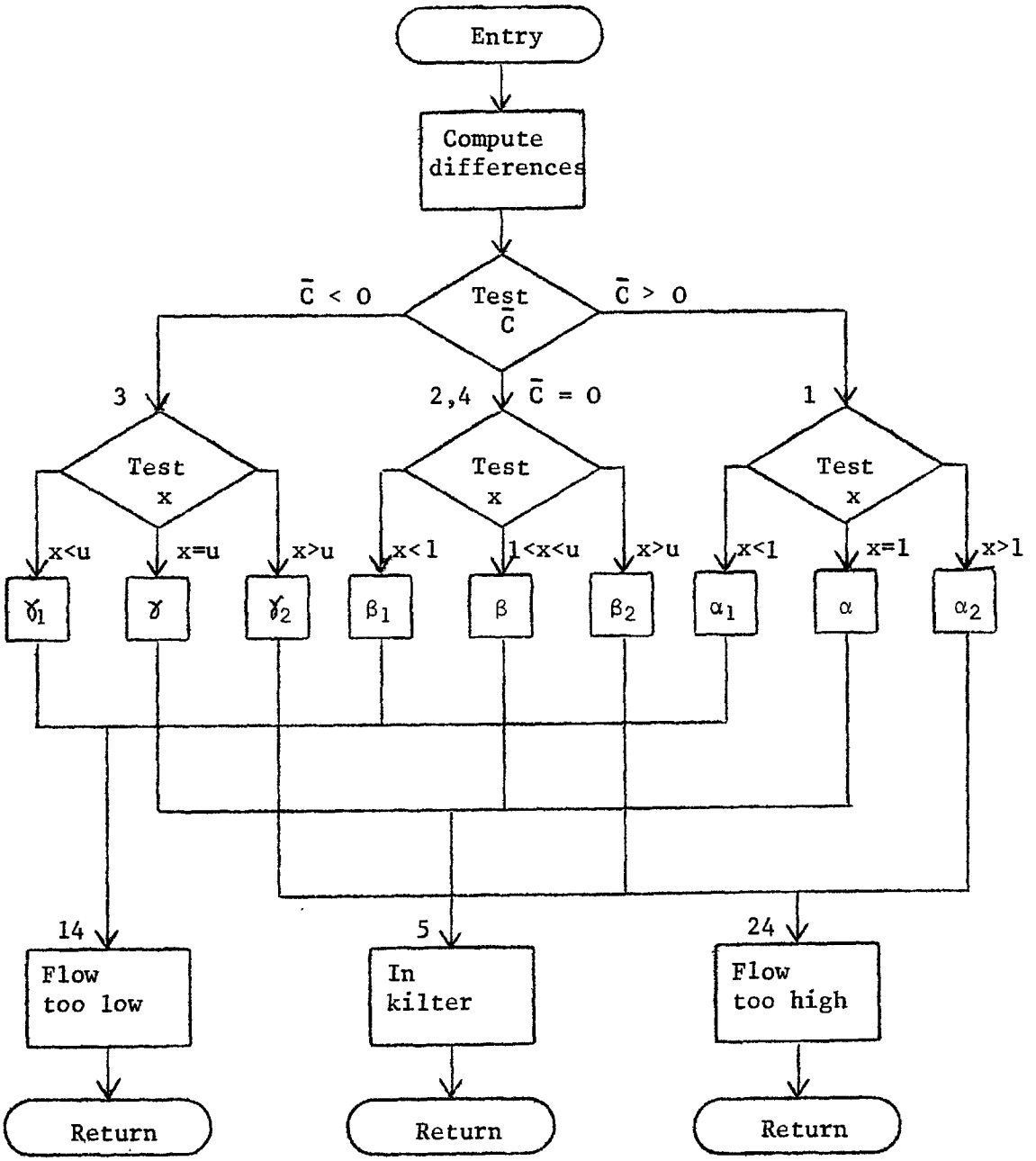
node	node location array
I	NODLOC(I)
1	1
2	7
3	.
4	.
5	.
6	
7	
8	.
9	.
10	.
11	80
12	87

node	location of signed arc	signed arc array
I	LSA	KS(LSA)
1	1	-1
	2	2
	3	7
	4	-8
	5	11
	6	-12
2	7	-3
	8	4
	9	-7
	.	.
	.	.
	.	.
	79	43
11	80	1
	81	-2
	82	3
	83	-4
	84	5
	85	-6
	86	-43

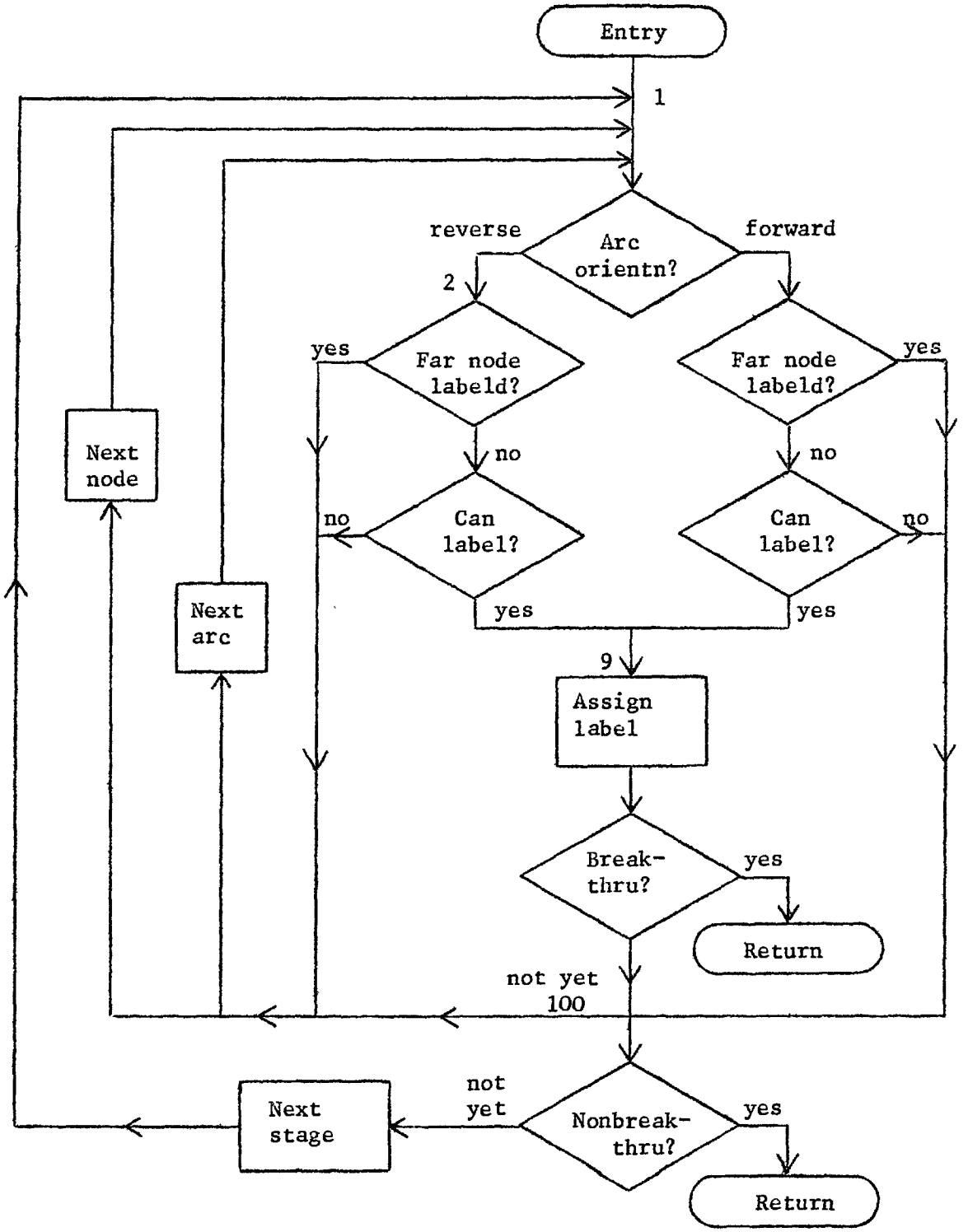
14.8.1 Flowchart Subroutine MULTAR



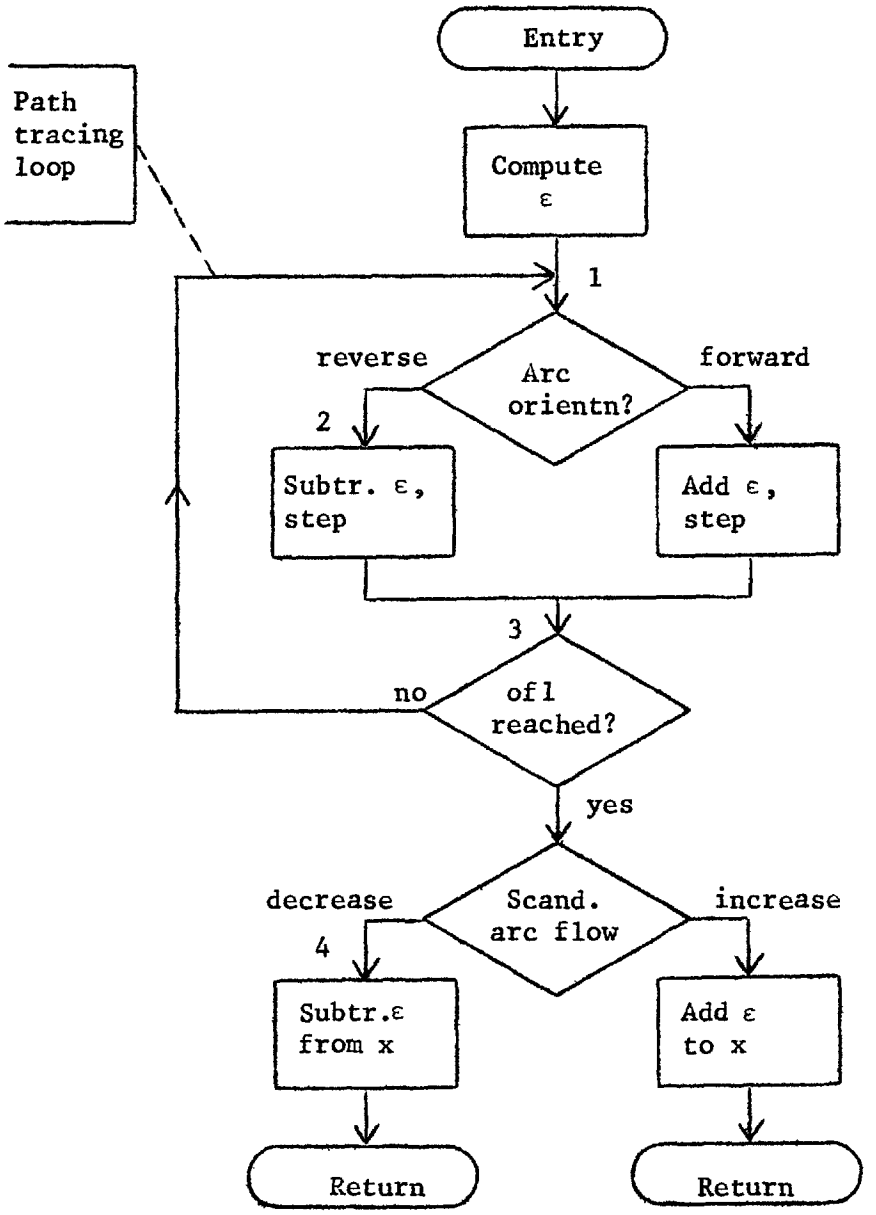
14.8.2 Flowchart Subroutine FINDST



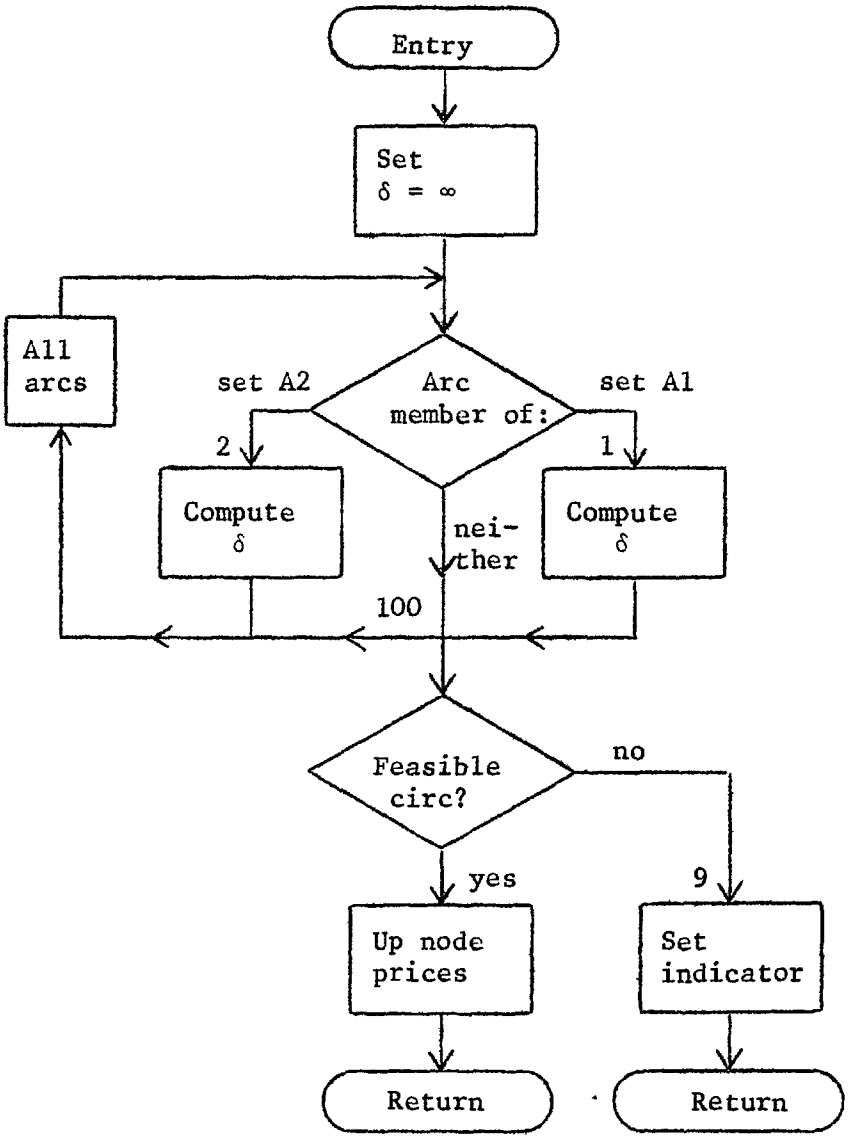
14.8.3 Flowchart Subroutine LABLEM



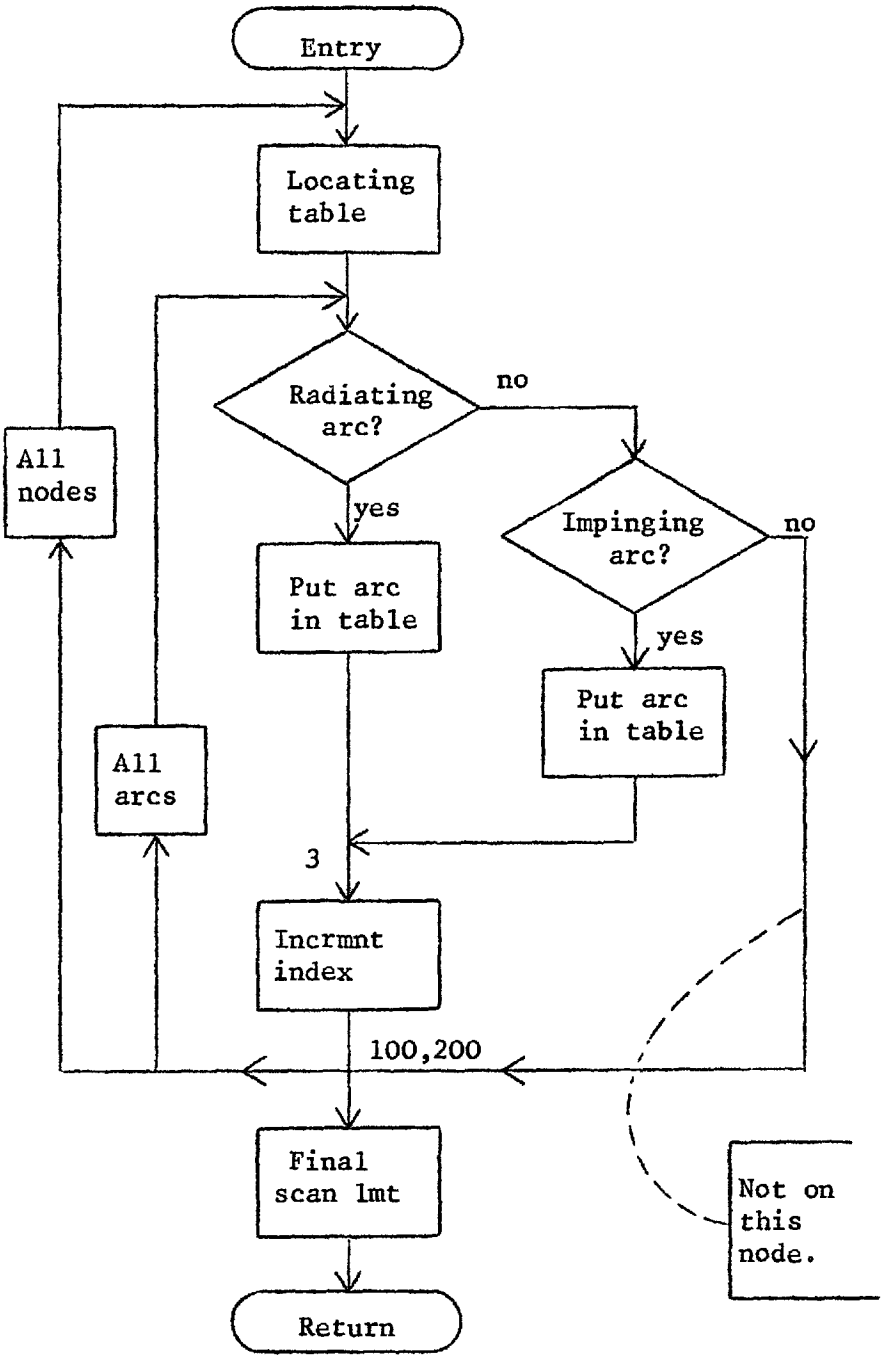
14.8.4 Flowchart Subroutine FLOCHM



14.8.5 Flowchart Subroutine NODPRI



14.8.6 Flowchart Subroutine NODARC



14.9

VARIABLES LIST

*

MULTAR MULTIPLE ARC OUT-OF-KILTER NETWORK FLOW ROUTINE

*

NAME OF FORTRAN VARIABLE	DESCRIPTION I INTEGER• A ALPHAMERIC• V VECTOR ARRAY M MATRIX ARRAY			
*				
ALF	A	ARC STATE ALPHA	α	
ALF1	A	ARC STATE ALPHA1		α_1
ALF2	A	ARC STATE ALPHA2	α_2	
ARCS	I	NUMBER OF ARCS IN THE NETWORK		a
BET	A	ARC STATE BETA	β	
BET1	A	ARC STATE BETA1		β_1
BET2	A	ARC STATE BETA2	β_2	
C	IV	ARC COSTS	$-$	c_k
CBAR	IV	OPTIMIZING COSTS	c_k	
DELTA	I	NODE PRICE CHANGE		δ
DIF	I	SCANNED ARC FLOW CHANGE DIFFERENCE	ϵ_k	
DIFL	I	ARC FLOW DIFFERENCE FOR LABELLING		
ELABEL	IV	EPSILON LABELS (FLOW CHANGE LABELS)	ϵ_i	
EPSILN	I	FLOW CHANGE		
FECIRC	I	FEASIBLE CIRCULATION INDICATOR		
GAM	A	ARC STATE GAMMA	γ	
GAM1	A	ARC STATE GAMMA1		γ_1
GAM2	A	ARC STATE GAMMA2	γ_2	
IA	I	NODE INDEX IN SUBR• FLOCHM		i
IL	I	NODE INDEX IN SUBR• LABELM		i
IM	I	NODE INDEX IN SUBR• NODARC		i
IN	I	NODE INDEX IN SUBR• NODPRI		i
INFINY	I	INFINITY• LARGEST INTEGER IN MACHINE	∞	
ISCAN	I	SCAN INDEX		
IPS	I	PRESENT STAGE NODE SCAN INDEX		
JL	I	FAR NODE INDEX	j	
K	I	ARC INDEX IN SUBR• MULTAR• AND COMMON		k
KA	I	ARC INDEX IN SUBR• FLOCHM		k
KJ	I	ARC INDEX IN SUBR• CPIPI		k
KL	I	UNSIGNED ARC INDEX		k
KLABEL	IV	PATH LABELS	$\pm k_i$	
KM	I	ARC INDEX IN SUBR• NODARC		k
KN	I	ARC INDEX IN SUBR• NODPRI		k
KS	IV	SIGNED ARC ARRAY		
L	IV	ARC LOWER BOUNDS	l_k	
LDFN	I	LABELLING DONE FROM•••NUMBER OF NODES		
LMX	I	FLOW DIFFERENCE L MINUS X		
LSA	I	NODE TO SIGNED ARC TABLE INDEX		

MD	I	POINTER DUMMY	
MN	I	NEXT STAGE POINTER	
MP	I	PRESENT STAGE POINTER	
NARCIL	I	NUMBER OF ARCS CONNECTED TO NODE IL	
NEXT	I	NUMBER OF NODES IN NEXT STAGE	
NODES	I	NUMBER OF NODES IN THE NETWORK	n
NODLOC	IV	NODE LOCATION ARRAY	
NPSTAG	IM	NEXT AND PRESENT STAGE NODE NUMBERS	
OFL	I	ORIGIN FOR LABELLING	
PI	IV	NODE PRICES (PRICING VECTOR)	π_i $\underline{\pi}$
PRES	I	NUMBER OF NODES IN PRESENT STAGE	
S	IV	START NODES (ARC TO NODE TABLE)	s
SKJ	I	START NODE OF ARC KJ IN SUBR. CPIPI	s
SKN	I	START NODE OF ARC KN IN SUBR. NODPRI	s
STADES	A	STATE DESCRIPTION FOR ARC K	
STAGES	I	NUMBER OF LABELLING STAGES FOR ARC K	
STAIND	I	STATE INDICATOR FOR ARC K	
T	IV	TERMINAL NODES (ARC TO NODE TABLE)	t
TBT	I	TOTAL BREAKTHROUGHS	
TFL	I	TERMINAL FOR LABELLING	T
TKJ	I	TERMINAL NODE OF ARC KJ IN SUBR. CPIPI	t
TKN	I	TERMINAL NODE OF ARC KN IN SUBR. NODPRI	t
TNBT	I	TOTAL NONBREAKTHROUGHS	
U	IV	ARC UPPER BOUNDS (ARC CAPACITIES)	u_k
UMX	I	FLOW DIFFERENCE U MINUS X	
X	IV	ARC FLOWS (NETWORK FLOW. CIRCULATION)	x_k, $\underline{x}$
XML	I	FLOW DIFFERENCE X MINUS L	
XMU	I	FLOW DIFFERENCE X MINUS U	

14.10 Source program listing

```
      SUBROUTINE MULTAR
C     OUT-OF-KILTER ALGORITHM TO FIND MINIMAL-COST NETWORK
C     FLOW FOR UPPER AND LOWER BOUND CAPACITATED NETWORK
C     MULTIPLE ARCS ALLOWED BETWEEN NODE PAIRS
      INTEGER TBT,TNBT,FECIRC,PI,ARCS,STAIND
      COMMON /BKKIL/ L(1200),U(1200),C(1200),X(1200),
     1CBAR(1200),S(1200),T(1200),PI(400),KLABEL(400),
     2ELABEL(400),KS(2400),NODLOC(401),DIF,FECIRC,ARCS,NODES,
     3STAIND,STADES,K,NEXT,OFL,TFL,INFINY,TBT,TNBT,LDFN,STAGES
C                    INFINITY IS LARGEST INTEGER IN MACHINE
      INFINY=34359738367
      TBT=0
      TNBT=0
      FECIRC=-1
C          BRING EACH ARC K INTO KILTER
      DO 100 K=1,ARCS
C                    FIND THE STATE OF ARC K
    1 CALL FINDST
C                    IF ARC K IS IN KILTER, NO NEED TO LABEL
      IF(STAIND.EQ.0) GO TO 100
C                    LABELING PROCEDURE SEARCHES FOR A PATH
      CALL LABLEM
C                    TEST FOR BREAKTHROUGH
      IF(NEXT.EQ.0) GO TO 2
C                    BREAKTHROUGH, CHANGE THE ARC FLOWS
      CALL FLOCHM
      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)
    4 FORMAT( 37H NO FEASIBLE SOLUTION TO THIS PROBLEM
     18H EXISTS.)
      GO TO 6
C                   COMPUTE NEW OPTIMIZING COST, CBAR
    3 CALL CPIPI
      GO TO 1
  100 CONTINUE
      FECIRC=1
      WRITE(6,5)
    5 FORMAT( 37H ALGORITHM TERMINATES WITH FEASIBLE,
     117HOPTIMAL SOLUTION.)
    6 WRITE(6,7) TBT,TNBT
    7 FORMAT(14H BREAKTHROUGHS,I4,3X,16HNONBREAKTHROUGHS,
     1I4)
      RETURN
      END
```

```
      SUBROUTINE FINDST
C     FINDS THE STATE OF ARC K, AND CHOOSES APPROPRIATE CASE
      INTEGER XML,XMU,UMX,X,U,CBAR,STAIND,DIF,OFL,TFL,S,T
      COMMON /BKKIL/ L(1200),U(1200),C(1200),X(1200),
     1CBAR(1200),S(1200),T(1200),PI(400),KLABEL(400),
     2ELABEL(400),KS(2400),NODLOC(401),DIF,FECIRC,ARCS,NODES,
     3STAIND,STADES,K,NEXT,OFL,TFL,INFINY,TBT,TNBT,LDFN,STAGES
      DATA ALF,BET,GAM,ALF1,ALF2,BET1,BET2,GAM1,GAM2
     1/5HALPHA,4HBETA,5HGAMMA,6HALPHA1,6HALPHA2,5HBETA1,
     25HBETA2,6HGAMMA1,6HGAMMA2/
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) 10,6,20
    2 IF(XMU) 4,7,21
    4 IF(XML) 11,7,7
    3 IF(XMU) 12,8,22
C          IN KILTER STATES
    6 STADES=ALF
      GO TO 5
    7 STADES=BET
      GO TO 5
    8 STADES=GAM
    5 STAIND=0
      RETURN
C          OUT OF KILTER STATES
C                     ORIGIN FOR LABELING IS TERMINAL OF ARC K
   10 STADES=ALF1
      DIF=LMX
      GO TO 14
   11 STADES=BET1
      GO TO 13
   12 STADES=GAM1
   13 DIF=UMX
   14 OFL=T(K)
      TFL=S(K)
      STAIND=1
      RETURN
```

```
C                       ORIGIN FOR LABELING IS START OF ARC K
   20 STADES=ALF2
      GO TO 23
   21 STADES=BET2
   23 DIF=XML
      GO TO 24
   22 STADES=GAM2
      DIF=XMU
   24 OFL=S(K)
      TFL=T(K)
      STAIND=2
      RETURN
       END



      SUBROUTINE CPIPI
C     COMPUTES THE ENTIRE OPTIMIZING COST ARRAY, CBAR
      INTEGER ARCS,SKJ,S,T,TKJ,CBAR,C,PI
      COMMON /BKKIL/ L(1200),U(1200),C(1200),X(1200),
     1CBAR(1200),S(1200),T(1200),PI(400),KLABEL(400),
     2ELABEL(400),KS(2400),NODLOC(401),DIF,FECIRC,ARCS,NODES,
     3STAIND,STADES,K,NEXT,OFL,TFL,INFINY,TBT,TNBT,LDFN,STAGES
      DO 100 KJ=1,ARCS
      SKJ=S(KJ)
      TKJ=T(KJ)
  100 CBAR(KJ)=C(KJ)+PI(SKJ)-PI(TKJ)
      RETURN
      END
```

```
      SUBROUTINE LABLEM
C     NODE LABELING PROCEDURE
      INTEGER ELABEL,OFL,TFL,STAGES,PRES,S,T,X,CBAR,U,DIFL
      DIMENSION NPSTAG(400,2)
      COMMON /BKKIL/ L(1200),U(1200),C(1200),X(1200),
     1CBAR(1200),S(1200),T(1200),PI(400),KLABEL(400),
     2ELABEL(400),KS(2400),NODLOC(401),DIF,FECIRC,ARCS,NODES,
     3STAIND,STADES,K,NEXT,OFL,TFL,INFINY,TBT,TNBT,LDFN,STAGES
C                    INITIALIZATION
      LDFN=1
      STAGES=1
      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))
C                    TEST FOR FORWARD OR REVERSE ARC
      IF(KS(LSA).LT.0) GO TO 2
C          KL IS A FORWARD ARC WITH FAR NODE ITS TERMINAL
      JL=T(KL)
C                    TEST IF FAR NODE JL HAS BEEN LABELED
      IF(KLABEL(JL).NE.0) GO TO 100
C                    SCAN THE FORWARD ARC KL
      IF(CBAR(KL).LE.0) GO TO 3
C                    CASE 3.1.A. STATE ALPHA1
      IF(X(KL).GE.L(KL)) GO TO 100
      DIFL=L(KL)-X(KL)
      GO TO 9
C                    CASE 3.1.B. STATES BETA.BETA1.GAMMA1
    3 IF(X(KL).GE.U(KL)) GO TO 100
      DIFL=U(KL)-X(KL)
      GO TO 9
```

```
C           KL IS A REVERSE ARC WITH FAR NODE ITS START
    2 JL=S(KL)
C                       TEST IF NODE JL HAS BEEN LABELED
      IF(KLABEL(JL).NE.0) GO TO 100
C                       SCAN THE REVERSE ARC KL
      IF(CBAR(KL).LT.0) GO TO 5
C                       CASE 3.2.A. STATES BETA. BETA2. ALPHA2
      IF(X(KL).LE.L(KL)) GO TO 100
      DIFL=X(KL)-L(KL)
      GO TO 9
C                       CASE 3.2.B. STATE GAMMA2
    5 IF(X(KL).LE.U(KL)) GO TO 100
      DIFL=X(KL)-U(KL)
C                       ASSIGN EPSILON LABEL TO NODE JL
    9 ELABEL(JL)=MIN0(ELABEL(IL).DIFL)
C                       ASSIGN PATH LABEL TO NODE JL
      KLABEL(JL)=KS(LSA)
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
C                       STEP TO NEXT STAGE
      LDFN=LDFN+NEXT
      PRES=NEXT
      MD=MP
      MP=MN
      MN=MD
      STAGES=STAGES+1
      GO TO 1
      END
```

```
      SUBROUTINE FLOCHM
C     CHANGES THE FLOW IN ALL ARCS OF THE CYCLE JUST FOUND
      INTEGER EPSILN,ELABEL,TFL,OFL,DIF,S,T,X,STAIND
      COMMON /BKKIL/ L(1200),U(1200),C(1200),X(1200),
     1CBAR(1200),S(1200),T(1200),PI(400),KLABEL(400),
     2ELABEL(400),KS(2400),NODLOC(401),DIF,FECIRC,ARCS,NODES,
     3STAIND,STADES,K,NEXT,OFL,TFL,INFINY,TBT,TNBT,LDFN,STAGES
C                    COMPUTE THE FLOW CHANGE, EPSILON
      EPSILN=MIN0(ELABEL(TFL),DIF)
C                    BACKTRACK FROM THE TERMINAL FOR LABELING
      IA=TFL
C          MAIN PATH TRACING LOOP
C                    REMOVE SIGN FOR SUBSCRIPTING
    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                    ADD EPSILON TO FLOW IN FORWARD ARC KA
      X(KA)=X(KA)+EPSILN
C                    STEP TO THE NEXT NODE
      IA=S(KA)
      GO TO 3
C                    SUBTRACT EPSILON FROM FLOW IN
C                    REVERSE 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
      INTEGER DELTA,ARCS,SKN,TKN,S,T,CBAR,X,U,PI,FECIRC
      COMMON /BKKIL/ L(1200),U(1200),C(1200),X(1200),
     1CBAR(1200),S(1200),T(1200),PI(400),KLABEL(400),
     2ELABEL(400),KS(2400),NODLOC(401),DIF,FECIRC,ARCS,NODES,
     3STAIND,STADES,K,NEXT,OFL,TFL,INFINY,TBT,TNBT,LDFN,STAGES
      DELTA=INFINY
C          COMPUTE DELTA, THE NODE PRICE INCREASE
      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.0).AND.(KLABEL(TKN).EQ.0).AND.
     1(CBAR(KN).GT.0).AND.(X(KN).LE.U(KN))) GO TO 1
C                    FIND IF ARC KN IS A MEMBER OF SET A2
      IF((KLABEL(TKN).NE.0).AND.(KLABEL(SKN).EQ.0).AND.
     1(CBAR(KN).LT.0).AND.(X(KN).GE.L(KN))) GO TO 2
      GO TO 100
    1 DELTA=MIN0(CBAR(KN),DELTA)
      GO TO 100
    2 DELTA=MIN0(-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.0) PI(IN)=PI(IN)+DELTA
  200 CONTINUE
      RETURN
C                    NO FEASIBLE CIRCULATION
    9 FECIRC=0
      RETURN
      END
```

```
      SUBROUTINE NODARC
C     BUILDS NODE TO ARC TABLE FROM ARC TO NODE TABLE
      INTEGER ARCS,S,T
      COMMON /BKKIL/ L(1200),U(1200),C(1200),X(1200),
     1CBAR(1200),S(1200),T(1200),PI(400),KLABEL(400),
     2ELABEL(400),KS(2400),NODLOC(401),DIF,FECIRC,ARCS,NODES,
     3STAIND,STADES,K,NEXT,OFL,TFL,INFINY,TBT,TNBT,LDFN,STAGES
      LSA=1
C          TAKE EACH NODE IN TURN
      DO 100 IM=1,NODES
C                    NODE IM IS AT LOCATION LSA
C                    IN THE NODE TO ARC TABLE
      NODLOC(IM)=LSA
C                    SEARCH ARC TO NODE TABLE FOR ARCS
C                    CONNECTED TO NODE IM.
      DO 200 KM=1,ARCS
C                    TEST FOR RADIATING ARCS
      IF(S(KM).NE.IM) GO TO 2
      KS(LSA)=+KM
      GO TO 3
C                    TEST FOR IMPINGING ARCS
    2 IF(T(KM).NE.IM) GO TO 200
      KS(LSA)=-KM
C                    INCREMENT THE NODE TO ARC TABLE INDEX
    3 LSA=LSA+1
  200 CONTINUE
  100 CONTINUE
C                    FINAL SCAN LIMIT IS NECESSARY FOR
C                    THE LAST NODE
      NODLOC(NODES+1)=LSA
      RETURN
      END
```

15 References

1 'American Standard FORTRAN, X3.9-1966,' American Standards Association, March 7, 1966.

2 'IBM7090/7094 IBSYS Operating System, Version 13, FORTRAN IV Language, File No. 7090-25, Form C28-6390-3,' International Business Machines Corporation, April 1966.

3 FULKERSON,D.R.: 'An out-of-kilter method for minimal-cost flow problems,' J. Soc. ind. appl. Math., 1961, 9, pp18-27.

4 FORD, L.R.Jr. and FULKERSON, D.R.: 'Flows in Networks' (Princeton University Press, 1962).

5 BERGE, C. and GHOUILA-HOURI, A.: 'Programming, Games and Transportation Networks' (Methuen and Co. Ltd., 1965).

6 HADLEY, G.: 'Linear Programming' (Addison-Wesley, 1965).

7 'American Standard Flowchart Symbols for Information Processing, X3.5-1966,' American Standards Association, June 8, 1966.

16 Similar programs

1 CLASEN, R.J.: 'RS OKF1, Out of Kilter Network Flow Routine One,' Share Program Library, number 1084; The RAND Corporation, March 1, 1961. FORTRAN II, FAP.

2 BRIGGS, W.A.: 'Algorithm 248, NETFLOW(H),' Marathon Oil Company, Communs Ass. comput. Mach., 1965, 8, pp103-104. ALGOL.

3 BRAY, T.A. and WITZGALL, C.: 'Algorithm 336, NETFLOW(H),' Boeing Scientific Research Laboratories, Communs Ass. comput. Mach., 1968, 11, pp631-632. ALGOL.

Program title: Conversational Mode Load Flow Program

Program name: CMLFP

Author: D.G. Stone

Source language: FORTRAN IV. See reference 1.

Machine: DEC PDP-9, basic configuration.
Core memory size is 8192_{10} words.
Word length is 18 bits.
300 character per second paper tape reader.
50 character per second paper tape punch.
Teletype model KSR33.

Monitor: PDP-9 Advanced Software Input/Output Monitor, issued October 12, 1967.

Date written: 1 June 1968.

Source program length:

	text lines	FORTRAN statements
Program CMLFP	215	125
Subroutine GANDS	90	50
Subroutine BUSTYP	23	12
Subroutine ZYZ	9	6
totals	337	193

Media: Standard IOPS ASCII, eight-hole, fan-fold paper tape; one inch wide, with 8½ inch long fan-folds.

Required program: PSDAP, Power system data assembly program.

Location: Centre for Computing and Automation
Electrical Engineering Building
Imperial College
Exhibition Road
LONDON S.W. 7.
U.K.

TABLE OF CONTENTS

1 Purpose
2 Method
3 Description
4 Usage
4.1 Tapes required
4.2 Loading instructions
4.3 Operating instructions
4.3.1 Command repertoire
4.3.2 Typical initial sequence of commands
4.3.3 Notes on operation
4.3.4 Recovery from errors
4.4 Messages
4.5 Input data preparation
4.6 Output
4.7 Run requirements diagram
5 Example
5.1 Teletype output
5.2 Operational data tape listing
5.3 AEP14 system diagram
6 Restrictions
7 Accuracy
8 Data checking
9 Storage
10 Execution time
11 Program details
11.1 Main program CMLFP
11.2 Subroutine GANDS
11.3 Subroutine BUSTYP
11.4 Subroutine ZYZ
11.5 Flowchart CMLFP
11.6 Flowchart GANDS

11.7 Variables list

11.8 Source program listings

12 Machine dependencies

13 Program modification

14 Checkout status

15 References

1 Purpose

Conversational mode simulation of an electric power system on a small digital computer.

2 Method

The Gauss-Seidel iterative solution of the power system network equations in nodal form, as presented by Glimn and Stagg in reference 4.

3 Description

The nodal equations of the network are expressed in rectangular coordinates of complex impressed power and bus voltage, precisely as implemented in the program. The indexing scheme and variable names are those of the program. Variables not defined here may be found in subsection 11.7 Variables list. All quantities are in per unit on a line-to-line KV base and a total three-phase MVA base.

At bus i the complex power impressed on the network, written as the sum of real and reactive components, is $S_i = p_i + jq_i$. V_i, the complex bus voltage is $u_i + jw_i$ in rectangular coordinates and $v_i\, e^{j\theta_i}$ in polar coordinates. Bus i has a self-admittance per phase of $Y_i^b = g_i^b + jb_i^b$, and a corresponding self-impedance $Z_i^b = r_i^b + jx_i^b = 1/Y_i^b$. The series admittance of branch k is $Y_k = g_k + jb_k$. Note that Y_{ij}, the mutual admittance between bus i and bus j is not used in this formulation. $Y_{ij} = -Y_k$. B_i is the set of all branches, both radiating and impinging, connected to bus i. For a particular branch k, j is the far bus at the other end of the branch from bus i.

An n-bus network is described by n nonlinear simultaneous equations

$$S_i = Y_i^{b*} V_i V_i^* + V_i \sum_{k \in B_i} Y_k^* V_j^* \qquad i = 1,2, \ldots, n$$

in the complex variables S and V. Rewriting them in component form produces 2n nonlinear, simultaneous algebraic equations in real variables. Since there are four variables per bus, the equation set is underspecified. A power system operational condition can be defined by giving two of the variables at each bus numerical values, and solving the equations for the remaining 2n variables. Three different pairs of independent variables are selected, defining the three bus types described below.

Solution of the network equations begins with an initial set of assumed voltages and proceeds in an iterative fashion, correcting the initial voltages bus by bus. Reference 4 describes the procedure in detail.

For each bus type, the network equations are written in a form that reflects the choice of independent variable.

1 Slack generator bus

One bus must be designated to maintain the real power balance by picking up the unassigned network losses. Its voltage magnitude and angle are chosen and fixed; the former at some desired value and the latter at zero, as an angular reference. Thus the real and imaginary components of bus voltage are known and

$$p_i = g_i^b v_i^2 - u_i \sum_{k \in B_i} (g_k u_j - b_k w_j) - w_i \sum_{k \in B_i} (b_k u_j + g_k w_j)$$

$$q_i = -b_i^b v_i^2 - w_i \sum_{k \in B_i} (g_k u_j - b_k w_j) + u_i \sum_{k \in B_i} (b_k u_j + g_k w_j)$$

2 Load bus

At this type of bus the given complex power generation and demand have been netted to fix the complex impressed power, S_i. The complex bus voltage is not controlled, and is therefore a dependent variable. Accordingly, the network equation at bus i is re-written

$$V_i = \frac{1}{Y_i^b}\left[\frac{S_i^*}{V_i^*} - \sum_{k\in B_i} Y_k V_j\right]$$

Note that the equation is implicit in V_i. The first value of V_i calculated at each iteration is re-inserted into the R.H.S. expression and an improved value obtained before moving on to the next bus. In component form

$$u_i = \frac{1}{v_i^2}\left[r_i^b(p_i u_i + q_i w_i) - x_i^b(p_i w_i - q_i u_i)\right]$$

$$+ r_i^b \sum_{k\in B_i}(g_k u_j - b_k w_j) - x_i^b \sum_{k\in B_i}(b_k u_j + g_k w_j)$$

$$w_i = \frac{1}{v_i^2}\left[r_i^b(p_i w_i - q_i u_i) + x_i^b(p_i u_i + q_i w_i)\right]$$

$$+ r_i^b \sum_{k\in B_i}(b_k u_j + g_k w_j) + x_i^b \sum_{k\in B_i}(g_k u_j - b_k w_j)$$

3 Voltage-controlled bus

Here the specified quantities are the real impressed power p_i and v_i, the bus voltage magnitude. First q_i is calculated by evaluating the expression in 1 with the current values of bus voltage. Next this new value of q_i is used in calculating new bus voltage components from the expressions in 2. The corresponding bus voltage angle θ_i, effectively the second dependent variable, is now taken as correct. However the voltage magnitude must be adjusted to the set value by modifying each component by a fixed fraction

$$u_i^{new} = u_i \frac{v_i^{set}}{(u_i^2 + w_i^2)^{\frac{1}{2}}}, \qquad w_i^{new} = w_i \frac{v_i^{set}}{(u_i^2 + w_i^2)^{\frac{1}{2}}}$$

Instead of replacing the old bus voltage components at bus i with the values just calculated, their difference is computed. This voltage correction is then amplified by an acceleration factor, in order to improve convergence.

Tap-changing transformers are modelled by a pi equivalent circuit, as described in reference 3. Let n_k be the present p.u. tap ratio and n_k^{new} a new tap ratio setting for transformer branch k. Then

$$Y_k^{new} = Y_k \frac{n_k^{new}}{n_k}$$

is the new series admittance of branch k, Y_k being the present value. In CMLFP, transformers are always oriented with their low voltage bus as the terminal bus j of the transformer branch k = (i,j). Only the self-admittance of the low voltage, tap-changing bus need be modified:

$$Y_j^{bnew} = Y_j^b + Y_k \frac{((n_k^{new})^2 - n_k^2)}{n_k}$$

The calculation of transformer branch power flows is complicated by off-nominal taps. The shunt elements in the pi equivalent circuit are then non-zero, and must be accounted for in the power flow computation. By using a factor, a, power flows in all types of branches can be calculated with a single pair of expressions:

$$p_i^k = v_i^2 g_k a - g_k(u_i u_j + w_i w_j) + b_k(u_i w_j - u_j w_i)$$

$$q_i^k = -v_i^2 b_k a + g_k(u_i w_j - u_j w_i) + b_k(u_i u_j + w_i w_j)$$

At a transformer high voltage bus, $a = 1/n_k$; while at its low voltage bus $a = n_k$. Setting $a = 1$ renders the expressions valid for the determination of transmission line power flows. p_i^k means the real power flow into branch k at bus i. As the transmission line pi model shunt elements b_k^s are not stored by CMLFP, the reactive line flows are somewhat in error. Replacing b_k in the first term of the q_i^k expression by $b_k + b_k^s$ would correct this deficiency.

4 Usage

4.1 Tapes required

1 Binary object program tapes containing the compiled versions of:

Program CMLFP

Subroutine GANDS

Subroutine BUSTYP

Subroutine ZYZ

2 A power system constant data (CD) tape punched by PSDAP.

3 An operational data (OD) tape for the same power system, prepared according to the instructions given in section 4.5.

4 Linking Loader, and library tapes.

4.2 Loading instructions

Collateral reference: subsection ~~4.6~~ 5.1

1 Turn the console POWER ON-OFF switch to ON.

2 Set the address switches to 17720_8.

3 Put the Linking Loader tape in the paper tape reader.

4 Push the reader feed button momentarily, advancing the tape a little, but not past the blank leader.

5 Depress the console I/O RESET switch.

6 Depress the console READ IN switch.

7 The tape is read in, and then the machine types

LOADER

>

8 Put the CMLFP binary object program tape in the paper tape reader, touching the reader feed button again.

9 Type 3 commas (one for each subroutine) and then press the ALT MODE key.

10 The tape is read in, and as each subroutine is loaded its name and bottom memory location are typed. See subsection 5.1.

11 ↑P is output when the physical end of tape passes through the reader.

12 Put the library tape in the paper tape reader, touch the feed button.

13 Holding down the CTRL key on the teletype keyboard, type the letter P.

14 Machine echoes ↑P,and reads in the library tape, continuing the loader memory map as it does so.

15 If the library is on more than one tape, repeat steps 11 to 14.

16 When the machine types ↑S, loading is successful and complete.

4.3 Operating instructions

4.3.1 Command repertoire

COMMAND				DESCRIPTION
Task code	Integr	Cha1	Cha2	
12	3456	789012	345678	← print position
BR ↵				List the bus results and data.
CD ↵				Input the constant data tape of a new power system.
DE	13	15.6	7.0 ↵	Change the demand at bus 13 to 15.6 MW and 7.0 MVAR.
GE	2	30.↵		Set the real power generation at bus 2 to 30.0 MW.
GS ↵				Call the Glimn and Stagg algorithm to solve the nodal equations.
IL	80 ↵			Set the iterations limit to 80.
OD ↵				Read in a new operational data tape to change the power system conditions.
PF ↵				Compute the branch power flows.
SB	1 ↵			Select bus 1 as the slack generator bus.
TT	16	.985↵		Set the transformer tap ratio on branch number 16 to 0.985.
VM	6	1.06 ↵		Set the voltage magnitude at bus 6 to 1.06 p.u.

CMLFP commands are typed according to a fixed format. The user enters them into the computer by pressing the RETURN key, symbolized by ↵ . The command task codes are two-letter mnemonics, typed in print positions 1 and 2. The next four print positions hold an integer index, which must appear right-justified in the field. Then come two six-position fields to hold changes in power system variables. Numbers entered in these fields require a decimal point, but do not have to be justified. Up to 5 significant digits can be

accommodated, otherwise spaces may be left for clarity and fewer digits entered. The right-hand field does not have to be filled out by blanks and may be terminated early by ⟩ . Blank fields will be entered into the machine as zero.

4.3.2 Typical initial sequence of commands

1 To begin execution, hold down the CTRL key and type S.

2 Machine types back ↑S and then the conversational word
NEXT.
signifying that CMLFP is ready to accept commands from the user via the teletype. The only valid commands at this point are CD, IL, and SB.

3 Type CD ⟩

4 Machine responds with
PAUSE ØØØØØ1
indicating its readiness to read in the constant data (CD) tape.

5 Place the CD tape in the reader, tap the reader feed button, hold down CTRL and type P.

6 Machine echoes ↑P, reads in the CD tape, and types out
PAUSE ØØØØØ2
requiring the user to

7 Place the operational data (OD) tape in the reader, touch the feed button, and type P with the CTRL key held down.

8 Machine types ↑P, reads in the OD tape, and types
NEXT.

9 Set a suitable iterations limit for the solution algorithm by typing
IL␣␣50 ⟩
␣ symbolizes a blank space.

10 Select the slack generator bus, for example
SB␣␣␣1⟩

11 All commands are now valid. Of course BR and PF will not display correct results as we have not yet computed a solution.

12 At this point it is usual to specify changes via the commands TT, VM, GE, and DE.

13 Now solve the nodal equations by typing
GS ↵

14 While the calculation is in progress no commands may be given. When convergence of the solution is detected, the number of iterations taken is typed out, followed by NEXT. If NEXT. appears without a number, the system solution has not yet converged to the specified accuracy, but has run for the iterations limit.

15 BR ↵
will obtain for the user a listing of the bus results and data.

16 If these are as desired and he wishes to observe the branch power flows, he may type
PF ↵

4.3.3 Notes on operation

1 When all commands are valid, they may be issued in any sequence, or repeated any number of times.

2 Giving the CD command implies a repetition of steps 4 to 10 in the previous subsection, with the possible omission of step 9.

3 Until the slack bus is chosen, commands GS and VM are invalid.

4 Results identification and comments may be typed by the user, provided he avoids print positions 1 and 2.

5 The total number of iterations to converge is NL + I, where L is the iterations limit, N is the number of unsuccessful GS commands, and I the number of iterations printed out on the final, successful call to GS.

6 Discrete transformer tap settings must be manually converted by the operator to p.u. tap ratios.

4.3.4 Recovery from errors

1 Any operation can be safely aborted by holding CTRL and typing S. The machine will echo

↑S

NEXT.

2 If the user forgets to load the tape reader properly he will get an

.OTS11

↑P

message from the machine. Recover as in 1.

3 Recovery from some mistakes is automatic. See section 8.

4 Pressing the RUBOUT key n times deletes the previous n characters in a command. This computer software feature is especially useful if a non-numeric character gets typed in a numeric field. This CMLFP will not tolerate, giving wrong results even after attempted correction. RUBOUT prints on the teletype as a back slash.

5 If it is suspected that the data has been irretrievably corrupted by a slip, repeating the initial command sequence is a way of rectifying the errors quickly.

6 When the console STOP light comes on, the teletype is dead and the user has lost control. He can reload CMLFP, or alternatively, attempt a restart procedure.

4.4 Messages

One conversational word, NEXT. is the only message generated by CMLFP. It signals completion of the commands OD, GS, BR, and PF as these take some time to execute. NEXT. does not follow CD, because at that time the only valid operation is the loading of the OD tape. If the machine types NEXT. immediately following the entry of any command, that command is in error. After a correct TT, SB, VM, GE, IL, or DE command NEXT. is omitted, since these commands execute virtually instantaneously.

Object Time System error messages are listed in reference 2.

4.5 Input data preparation

The CD tape punched by PSDAP will have the correct format for input to CMLFP. This section describes the layout of the OD (operational data) tape. The AEP14 Standard Test Study OD tape is listed in subsection 5.2.

Line 1 of the text is the operational data study title, and line 2 is a heading for the generator bus numbers. Neither of these lines is processed, but they must be present on the tape. Line 3 lists the generator bus numbers, including all voltage-controlled busses. They appear 10 numbers per line, right-justified in 5-position data fields as specified by FORMAT statement 902. If there are more than 10 generator busses, the additional ones may be placed on a second line. Lines 4 and 5 are gain non-processed headings, inserted to improve the readability of the tape listing.

Then follow the operational conditions, one bus to a text line, for input by FORMAT 925. BN is the bus number. The remaining headings are program variable names defined in 11.7 Variables list. CMLFP does not process the BVANG variable, so no entries in that column need appear. However, an entry must be made for each bus, under BVMAG. Leaving a space after BN, the

next 10 positions are for non-processed bus names, if desired. Blank data items are converted to zeros. Transformer headings and data are not input from the tape. The user will enter off-nominal taps via the teletype keyboard.

4.6 Output

The one output device used by CMLFP is the teletype. Two commands initiate listing of data and results; BR and PF.

The bus results (BR) are arranged one bus per text line with headings

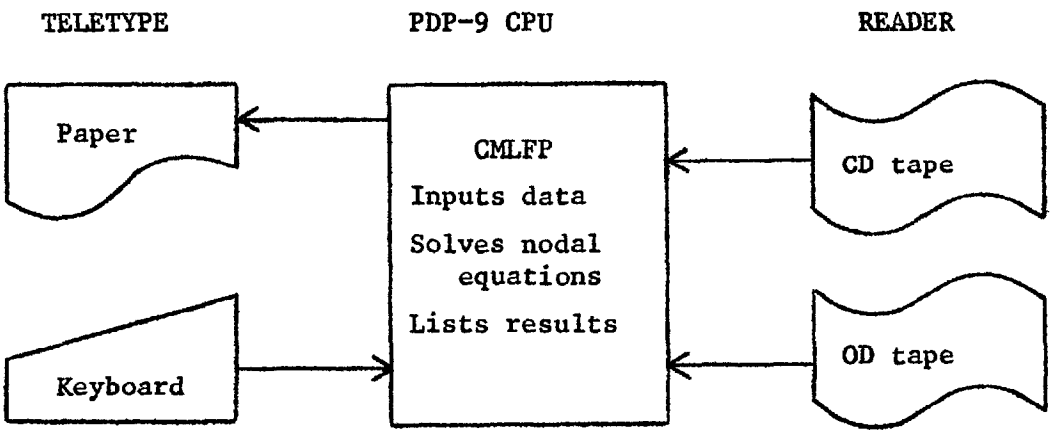
BUS	bus number
PL	real power demand, MW (imports negative)
QL	reactive power demand, MVAR (inductive positive)
PG	real power generation, MW
QG	reactive power generation, MVAR
V	bus voltage magnitude, p.u.
THETA	bus voltage angle, degrees

The headings for the branch power flows (PF) are as follows

LN	branch number
SB	start bus number
EB	end bus number
PS	real power sent from the start bus, MW
PR	real power received at the end bus, MW
QS	reactive power sent, MVAR
QR	reactive power received, MVAR

Real power loss in a line is PS-PR. The reactive line power flows do not include the contribution of the line shunt susceptance since this latter quantity has not been stored.

4.7 Run requirements diagram CMLFP



5 Example

Subsection 5.1 Teletype output logs the operation of CMLFP on our example, the American Electric Power 14 Bus Test System. To begin with, the user follows the operating sequence given in subsection 4.3.2. After reading in the power system data tapes CD and OD, he chooses an iterations limit of 20 and selects bus 1 as the slack bus. Next he sets the transformer tap ratios on branches 16, 17 and 18. The conditions of the test study are now fulfilled, and he calls the solution algorithm by typing GS ↵.

Realizing that he has set the iterations limit too low for complete convergence in one GS call, the user relaxes it to 50. He continues the solution by entering GS ↵ again. 14 additional iterations are performed, and convergence reached in a total of 34 iterations. The user types a line of text identifying the results, and then lists them by issuing the commands BR and PF. Subsection 5.3 presents this solution as a diagram.

The next case examines the effect of doubling the load at bus 13. Just after beginning the solution, the operator remembers that he meant to divide the demand increase between generators 1 and 2. So he abandons the solution and increases the real power generation on the bus 2 generator to 47.0 MW. On listing the bus results after convergence has been reached, he notices that the reactive power generation at bus 6 is very near its upper limit of 24 MVAR, and that the voltage magnitude at bus 13 is lower than before. He corrects this condition by raising the voltage on bus 6 and reducing the turns ratio on branch 18. Note that in setting v_6 to 1.08 he slipped and hit comma instead of decimal point, a disastrous error. Nevertheless recovery was possible by pushing RUBOUT and continuing the entry.

5.1 Teletype output

```
LOADER

>,,,
 CMLFP   15064
 GANDS   13555
 BUSTYP  13407
 ZYZ     13330
↑P↑P
 ABS     13312
 IABS    13276
 SQRT    13210
 ATAN    13175
 .ED     13110
 .EC     13044
 .DA     12775
↑P↑P
 BCDIO   10002
 .SS     07723
 GOTO    07675
 PAUSE   07661
 SPMSG   07566
 FIOPS   07032
 OTSER   06724
 INTEGE  06540
 REAL    05605
↑S↑S
    NEXT.

CD

PAUSE 000001
↑P
PAUSE 000002
↑P    NEXT.
IL  20
SB   1
TT  16 .978
TT  17 .969
TT  18 .932
GS
    NEXT.
IL  50
GS
     14
    NEXT.
```

```
  AEP14  STANDARD TEST STUDY, 34 ITERATIONS
    NEXT.
BR
    BUS     PL       QL       PG       QG        V       THETA
      1    0.00     0.00   232.39   -17.16    1.060     0.000
      2   21.70    12.70    40.00    42.08    1.045    -4.985
      3   94.20    19.00     0.00    23.33    1.010   -12.730
      4   47.80    -3.90     0.00     0.00    1.019   -10.344
      5    7.60     1.60     0.00     0.00    1.021    -8.778
      6   11.20     7.50     0.00    13.89    1.070   -14.831
      7    0.00     0.00     0.00     0.00    1.062   -13.669
      8    0.00     0.00     0.00    17.01    1.090   -13.669
      9   29.50    16.60     0.00     0.00    1.057   -15.317
     10    9.00     5.80     0.00     0.00    1.052   -15.516
     11    3.50     1.80     0.00     0.00    1.057   -15.304
     12    6.10     1.60     0.00     0.00    1.055   -15.669
     13   13.50     5.80    -0.00     0.00    1.051   -15.734
     14   14.90     5.00    -0.00     0.00    1.036   -16.497
    NEXT.
PF
    LN  SB  EB      PS       PR       QS       QR
     1   1   2   156.9    152.6   -17.5    -30.6
     2   1   5    75.5     72.8     6.0     -5.4
     3   2   3    73.2     70.9     5.9     -3.8
     4   2   4    56.3     54.6    -0.3     -5.5
     5   2   5    41.3     40.5     2.3     -0.4
     6   3   4   -23.3    -23.7     4.5      3.5
     7   4   5   -62.5    -63.0    15.8     14.1
     8   6  11     6.4      6.4     3.7      3.6
     9   6  12     7.7      7.6     2.5      2.4
    10   6  13    17.3     17.1     7.3      6.9
    11   9  10     6.1      6.1     4.1      4.0
    12   9  14    10.0      9.9     3.5      3.2
    13  10  11    -2.9     -2.9    -1.7     -1.8
    14  12  13     1.5      1.5     0.8      0.8
    15  13  14     5.1      5.0     1.8      1.8
    16   4   7    29.4     29.4    -9.1    -11.0
    17   4   9    16.3     16.3    -0.3     -1.7
    18   5   6    42.6     42.6    11.8      7.1
    19   7   8     0.0      0.0   -16.6    -17.0
    20   7   9    29.4     29.4     5.6      4.7
    NEXT.
DE  13 27.0  11.6
GS
↑S
    NEXT.

GE   2 47.
GS
     28
    NEXT.
```

```
BR
    BUS     PL       QL       PG       QG        V       THETA
      1    0.00     0.00   240.34   -17.17    1.060     0.000
      2   21.70    12.70    47.00    44.17    1.045    -5.106
      3   94.20    19.00     0.00    24.59    1.010   -13.081
      4   47.80    -3.90     0.00     0.00    1.016   -10.859
      5    7.60     1.60     0.00     0.00    1.018    -9.292
      6   11.20     7.50     0.00    23.25    1.070   -16.580
      7    0.00     0.00     0.00     0.00    1.061   -14.589
      8    0.00     0.00     0.00    17.72    1.090   -14.589
      9   29.50    16.60     0.00     0.00    1.056   -16.435
     10    9.00     5.80     0.00     0.00    1.051   -16.744
     11    3.50     1.80     0.00     0.00    1.057   -16.787
     12    6.10     1.60     0.00     0.00    1.050   -17.665
     13   27.00    11.60    -0.00     0.00    1.039   -17.870
     14   14.90     5.00    -0.00     0.00    1.031   -18.066
    NEXT.
VM    6 1.\.08
TT   18 .900
GS
     24
    NEXT.
BR
    BUS     PL       QL       PG       QG        V       THETA
      1    0.00     0.00   240.43   -15.38    1.060     0.000
      2   21.70    12.70    47.00    47.79    1.045    -5.116
      3   94.20    19.00     0.00    25.83    1.010   -13.110
      4   47.80    -3.90     0.00     0.00    1.014   -10.853
      5    7.60     1.60     0.00     0.00    1.014    -9.249
      6   11.20     7.50     0.00    18.11    1.080   -16.760
      7    0.00     0.00     0.00     0.00    1.062   -14.588
      8    0.00     0.00     0.00    17.20    1.090   -14.588
      9   29.50    16.60     0.00     0.00    1.059   -16.427
     10    9.00     5.80     0.00     0.00    1.056   -16.764
     11    3.50     1.80     0.00     0.00    1.064   -16.880
     12    6.10     1.60     0.00     0.00    1.060   -17.817
     13   27.00    11.60    -0.00     0.00    1.048   -17.999
     14   14.90     5.00    -0.00     0.00    1.036   -18.107
    NEXT.
```

5.2 Operational data tape listing

OD AEP14 STANDARD TEST STUDY
GENERATOR BUS NUMBERS
1 2 3 6 8
BUS OPERATIONAL DATA

BN	RDEM	XDEM	RGEN	XGEN	BVMAG	BVANG
1			0.	0.	1.06	0.
2	21.7	12.7	40.		1.045	
3	94.2	19.0			1.01	
4	47.8	-3.9			1.	
5	7.6	1.6			1.	
6	11.2	7.5			1.07	
7	0.	0.			1.	
8	0.	0.			1.09	
9	29.5	16.6			1.	
10	9.0	5.8			1.	
11	3.5	1.8			1.	
12	6.1	1.6			1.	
13	13.5	5.8			1.	
14	14.9	5.0			1.	

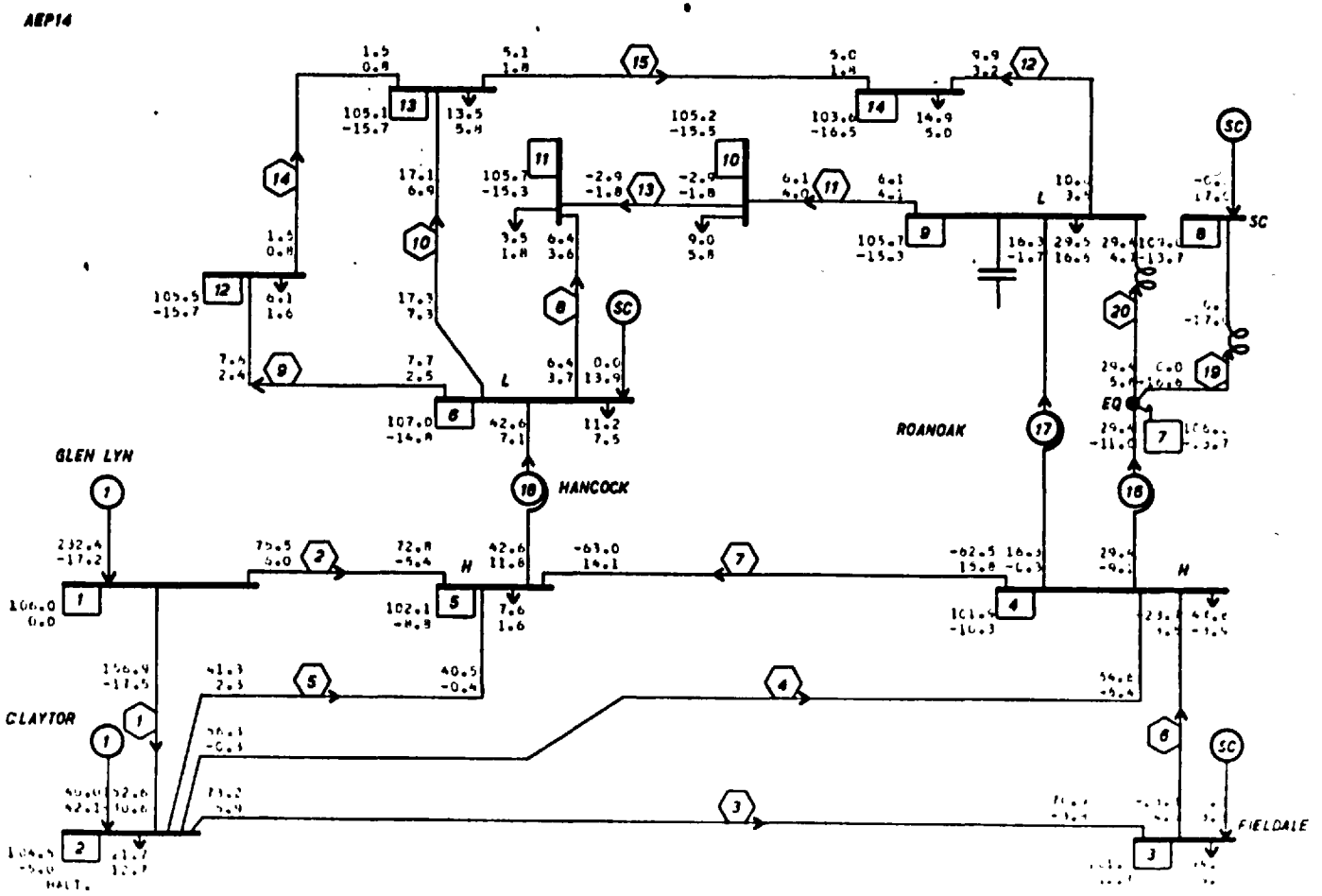
TRANSFORMER TAP SETTINGS

TN	TAPSET
16	.978
17	.969
18	.932

5.3 AEP14 System diagram

The bus numbers are in small rectangles attached to each bus, while the line numbers are enclosed by small hexagons for ease of reference. SC denotes synchronous capacitor, EQ an equivalent bus, L low voltage, and H high voltage. The data is plotted automatically by an expanded version of CMLFP.

Beside the bus number is printed the bus voltage magnitude in percent, and the bus voltage angle in degrees. Under the load arrows are the real power demand in MW, and the reactive power demand in MVAR. The generators and synchronous capacitors have their real and reactive power outputs listed to their left. Sending and receiving end branch power flows are positive when they have the same direction as the branch orientation.


AEP14
GLEN LYN
HANCOCK
ROANOAK
CLAYTOR
FIELDALE
SC
EQ
L
H

6 Restrictions

1 Upper limits on problem size are given in the PSDAP documentation.

2 The power system OD tape is required to conform to subsection 4.5

3 Notes 2 and 3 of subsection 4.3.3 describe temporary command restrictions.

4 The user must note power system operational constraints and satisfy them manually.

5 A badly chosen slack bus can cause slow convergence.

6 Input of a new CD tape resets the transformer taps to their nominal values, whereas a new OD tape does not change the taps.

7 Accuracy

All calculations are done in 6 digit, floating point arithmetic. The solution is not checked for power mismatch. However the bus voltage convergence criterion of 10^{-6} ensures convergence of bus powers to within the digits printed out.

8 Data checking

The first two items in a command are the only data elements checked by CMLFP. If the command task code does not match one of the 11 in the repertoire, NEXT. is printed by the program as an error indication. There is protection against trying to change taps on a branch other than a transformer. The commands SB, VM, and GE deal exclusively with generator busses, so that if a load bus is referred to, NEXT. is typed and no data is modified.

9 Storage

The total number of core storage locations filled by CMLFP and its three subroutines is $2247_{10} = 4307_8$. The three COMMON blocks occupy an additional $922_{10} = 1632_8$ locations. Moreover, the necessary subroutines from the library tape up another $2899_{10} = 5523_8$ machine memory words.

10 Execution time

For AEP14, the CD tape took 12 seconds to read in, and the OD tape 7 seconds. 38 seconds were required by the solution algorithm to iterate to convergence from a flat start. The bus results were listed in 1 minute and 25 seconds. PDP-9 FORTRAN arithmetic operations are done by software, so solution times are longer than expected. Nevertheless, while the machine is busy the user may occupy himself with data checking, hand calculation, and consideration of his power system problem. A 28-bus system converged in 91 iterations, spending 3 minutes and 18 seconds in subroutine GANDS.

11 Program details

11.1 Main program CMLFP

The basic form of this program consists of task selection and subsequent transfer to task-related subroutines. Unfortunately the body of CMLFP is cluttered up with five of these subroutines, obscuring its logical structure. This imbedding of the subroutines effected a significant economy of core memory locations, and allowed more power system features to be included.

The three COMMON blocks contain, in order, the power system parameters, the system constant data, and the system operational data. Array TL in the DATA statement holds the 11 alphameric task codes of the command repertoire.

The next task message NEXT. is placed at the beginning of the program so that when the ↑S monitor system feature is used to recover from operating mistakes, the program can immediately accept any command. Statements 51 and 907 input all commands; incidentally achieving economy in program space and consistency in command form. Subroutine BUSTYP is called so that the bus type entered in by the user can be tested in command sections SB, VM, and GE.

DO-loop 52 scans the task code list TL until a match is detected with the desired command held in variable TASK. Index ML, the statement numbers in the computed GO TO, and the task codes in TL are all arranged so that transfer to the correct program segment is made. If a match is not detected, DO 52 is satisfied, and NEXT. typed out to indicate a wrong task code entry by the user.

PAUSE's 1 and 2 remind the user to place the appropriate tape in the paper tape reader. The power system constant data, pre-assembled by PSDAP on the CD tape, is read in exactly as punched, and the transformer tapping ratios set to their nominal value of 1. By means of DO-loop 200, the bus operational data tape is read, one bus at a time. Real and reactive demands are stored in MW and MVAR's, then netted with the assigned generations and the result converted to p.u. to give the initial value of the bus impressed power. Next note the lack of provision for a non-flat start in bus voltage components. This omission saves core space by avoiding the use of trigonometric library subroutines.

Memory savings are also made by keeping no tables for directly referencing the two busses of each branch. Thus DO-loop 300 must scan the signed branch table for + K to find the end bus (LV bus) of branch K in table JA. This search and the computation that follows it appears to be instantaneous to the human operator. Statement 32 calculates the index for table TAP from the branch index entered by the user; and is the reason lines must preceed transformers in the branch numbering. The IF statement protects against wrong branch index entries by the user. The scheme for tap changing is that of reference 3.

Statements 7, 8, and 9 protect against referencing a non-voltage-controlled bus; as described in section 8. When a new slack bus is selected, its voltage components must be set to reference values. Also, if the user sets a new voltage magnitude

for the slack bus, the real component of slack bus voltage is reset to that value.

Since the complex power at each bus is not stored, changing the bus demand power is clumsy to program.

$$p_i^{new} = g_i^{old} - d_i^{new}$$

$$= p_i^{old} + d_i^{old} - d_i^{new}$$

Here CHA1 is the new real power demand at bus IB. RDEM is the array holding present values of demanded power. Both quantities are in MW, so the program statement includes conversion to p.u.

The branch power flows are computed and tabulated branch by branch in DO-loop 600. The signed branch table KS is scanned in DO 611 so that I and J the start and end busses of branch K may be found. The main body of DO 600 is used twice per branch, first for the sending end (IE = 0) and second for the receiving end (IE = 1). When the branch is a transformer, the factor AA must be set correctly for each end according to the method described in section 3. The end logic section interchanges the bus indicies, sets the end indicator IE to receive, and stores the sending end complex power for output after the receiving end power has been computed.

The main program of CMLFP calls subroutines GANDS, BUSTYP, and ZYZ.

11.2 Subroutine GANDS

The Gauss-Seidel iterative technique of reference 4 is implemented without changes as subroutine GANDS. DO-loop 501 forms the branch current sums by scanning all the branches connected to bus i. These branches and their corresponding far bus numbers have been collected together for high-speed reference in the signed branch table KS, and the far bus table JA. Their position in the tables

is defined by the scan limits JSTART and JFINIS, which in turn are calculated from the bus location table, IBSLOC. The IABS function removes the sign from KS to give the branch index K.

At each load bus the voltage calculation is performed twice; first with repetition indicator M=-1, and second with M=0. Temporary bus voltage components UIT and WIT were created because the present ones, U(I) and W(I) are needed for acceleration later in the DO-loop. The expression for VISQ is written without using the power 2 so as to avoid loading a library subroutine. IF(IBTYPE+M) tests the bus type and the load bus repetition indicator simultaneously. As the IF expression is evaluated; M=0 and IBTYPE=-1 after the first calculation of a load bus. So control transfers to statement 6, and the load bus voltage computation is repeated. Note that the precalculated branch current sums are valid on the load bus repetition. On the second evaluation of the IF, M is 1 and control passes to statement 5.

There the imaginary component of the present bus voltage is saved for convergence testing. The amplified difference (UIT - U(I))*1.6 between the present bus voltage component and the newly computed value is, in effect, and over-correction to the present value. The convergence test is applied only to the imaginary component of bus voltage, since it will be the more sensitive component under constant voltage operation and relatively small bus voltage angular differences. Each bus is tested when calculated, and if it fails, the convergence indicator IDRCON is changed from 1 to 0, signalling non-convergence in the current iteration.

At the end of each complete iteration, IDRCON itself is tested. If it is positive, all busses have converged and control exits from the iterations loop, DO 520. The user is informed of this fact via statement 14.

The program path taken while computing a voltage-controlled bus is somewhat tortuous in order to use program statements written

for the other bus types. Initially there will be a transfer to statement 3 since now IBTYPE=1. There the impressed reactive power Q(I) required to maintain the given bus voltage will be determined. Unless the present bus is the slack generator bus, testing for bus type sends control back to statement 6. The bus voltage components are then calculated, using the value of Q(I) just found. This time the bus type and repetitions test results in a transfer to statement 7; since M=0 and IBTYPE=1. IG, the generator bus index, has a value from the early call to subroutine BUSTYP. Before acceleration, the bus voltage components are adjusted so that the voltage magnitude is at the controlled value.

Slack bus calculations are straightforward, control first going to statement 3, and then to statement 15. The final term in the expression for P(I) is included for completeness, but is not strictly necessary, as W(I)=0. Since the slack bus is a voltage reference, its voltage components do not change. Therefore the modification, acceleration, and convergence-checking operations are redundant, and thus omitted.

Subroutine GANDS makes calls to subroutines BUSTYP and ZYZ.

11.3 Subroutine BUSTYP

The generator bus numbers and their bus voltage magnitude settings are stored in short tables IGENB and BVMSET. In addition, the three bus types are coded numerically:

load bus	-1
slack bus	0
generator bus	+1

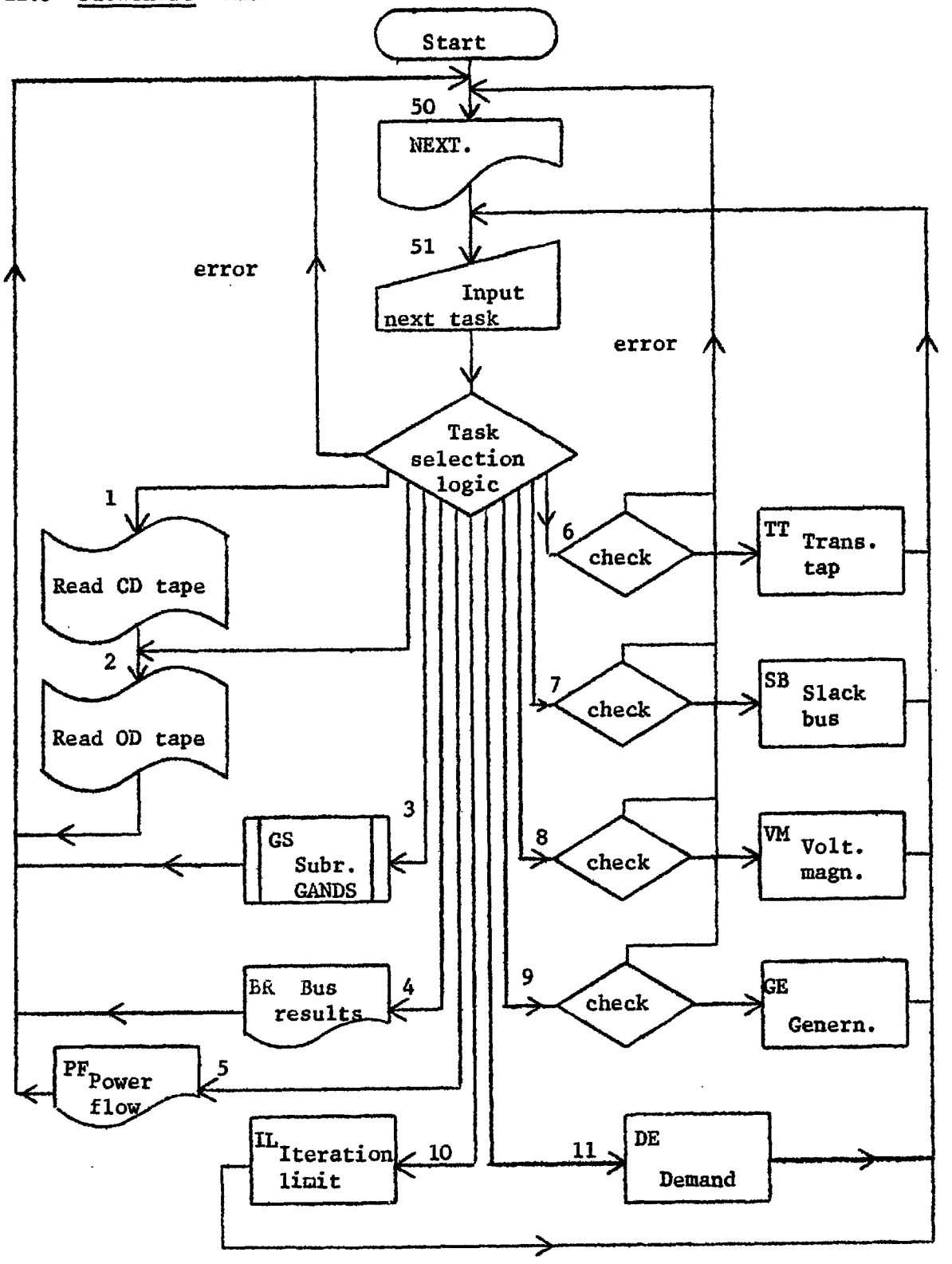
When presented with a bus number IB, the subroutine scans the array IGENB to check if it is a generator bus. If so, early exit from DO-loop 800 fixes IG, the short table index. The slack bus is also tested for, and indicator IBTYPE set appropriately.

11.4 Subroutine ZYZ

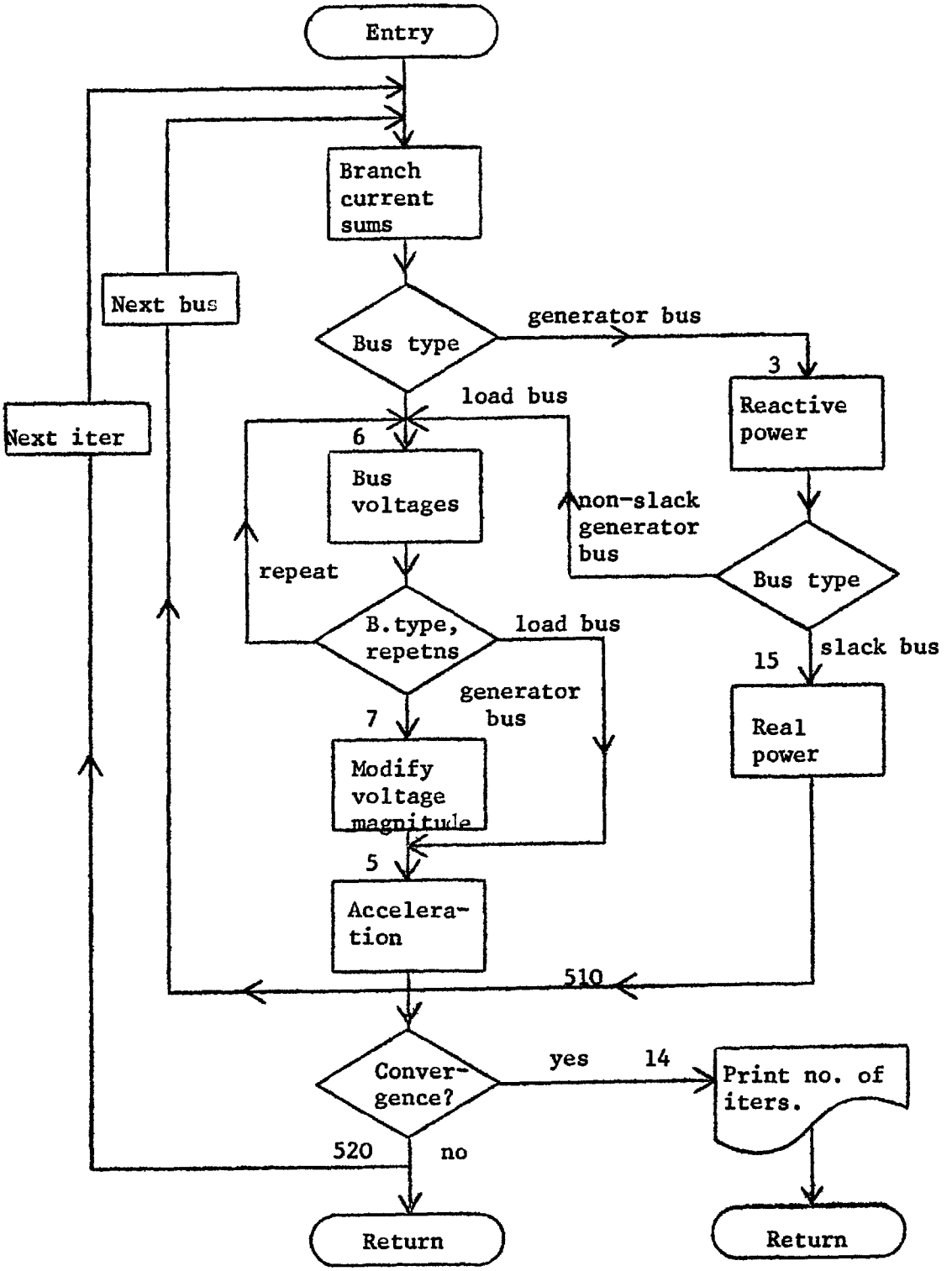
Since Y and Z are reciprocals, the expressions to convert one to the other have the same form, and can be used in either direction simply by exchanging the roles of the variables.

$$Y = \frac{1}{Z} = \frac{1}{r + jx} = \frac{r}{z^2} - \frac{jx}{z^2} = g + jb$$

11.5 Flowchart CMLFP



11.6 Flowchart Subroutine GANDS



11.7
VARIABLES LIST

*

CMLFP CONVERSATIONAL MODE LOAD FLOW PROGRAM

*

NAME OF FORTRAN VARIABLE	DESCRIPTION I INTEGER, R REAL, A ALPHABETIC, V VECTOR ARRAY		
AA	R	ADMITTANCE ADJUSTING FACTOR, LINE FLOWS	a
B	R	OUTPUT SUSC. OR REAC. IN SUBR. ZYZ	b
B	RV	BRANCH SERIES SUSCEPTANCE, P.U.	b_k b_i^b
BB	R	BUS SELF-SUSCEPTANCE, P.U.	
BRAK1	R	INTERMEDIATE VARIABLE IN SUBR. GANDS	
BRAK2	R	INTERMEDIATE VARIABLE IN SUBR. GANDS	
BVANG	R	BUS VOLTAGE ANGLE, DEGREES	θ_i
BVMSET	RV	BUS VOLTAGE MAGNITUDE SETTING, P.U.	v_i^{set}
BVMAG	R	BUS VOLTAGE MAGNITUDE, P.U.	v_i
CHA1	R	CHANGE ONE	
CHA2	R	CHANGE TWO	
FACTOR	R	TAP-CHANGING ADMITTANCE MOD. FACTOR	
G	R	OUTPUT COND. OR RES. IN SUBR ZYZ	g
G	RV	BRANCH SERIES CONDUCTANCE, P.U.	g_k g_i^b
GB	R	BUS SELF-CONDUCTANCE, P.U.	
I	I	BUS INDEX	i
IB	I	BUS NUMBER, DATA CHANGE NUMBER	i
IBSLOC	IV	BUS LOCATION TABLE	
IBTYPE	I	BUS TYPE INDICATOR	
IDRCON	I	CONVERGENCE INDICATOR	
IE	I	SEND-RECEIVE SWITCH	
IG	I	GENERATOR BUS INDEX	
IGENB	IV	LIST OF GENERATOR BUSSES	
ISLACK	I	SLACK BUS NUMBER	
ITERS	I	ITERATIONS INDEX	
ITLIM	I	ITERATIONS LIMIT	
J	I	BUS INDEX	j
JA	IV	END BUS TABLE	
JFINIS	I	FINAL BUS SCAN INDEX	
JSCAN	I	BUS SCAN INDEX	
JSTART	I	BUS SCAN INDEX START	
K	I	BRANCH INDEX	k
KS	IV	SIGNED BRANCH TABLE	
KT	I	TRANSFORMER INDEX	
LINES	I	NUMBER OF LINES IN THE NETWORK	
LSB	I	LOCATION SIGNED BRANCH	
M	I	LOAD BUS REPETITION INDICATOR, S. GANDS	
M	I	TEMPORARY STORAGE VARIABLE	
ML	I	TASK SELECTION INDEX	
MVABSE	I	MVA BASE OF POWER SYSTEM DATA	
NBUP1	I	NUMBER OF BUSSES PLUS ONE	
NBUSS	I	NUMBER OF BUSSES IN THE NETWORK	

NGBUS	I	NUMBER OF VOLTAGE-CONTROLLED BUSSES	
NRBRAN	I	NUMBER OF BRANCHES IN THE NETWORK	
NRBT2	I	NUMBER OF BRANCHES TIMES TWO	
NXFMRS	I	NUMBER OF TRANSFORMERS IN THE NETWORK	
P	RV	REAL POWER IMPRESSED ON THE NETWORK, P.U.	p_i p_i^k
PI	R	REAL POWER FLOW FROM BUS I, MW	
PR	R	REAL POWER RECEIVED, MW	
PS	R	REAL POWER SENT, MW	
Q	RV	REACTIVE POWER IMPRESSED, P.U.	q_i q_i^k
QI	R	REACTIVE POWER FLOW FROM BUS I, MVAR	
QR	R	REACTIVE POWER RECEIVED, MVAR	
QS	R	REACTIVE POWER SENT, MVAR	
R	R	INPUT RES. OR COND. IN SUBR. ZYZ	r
RB	RV	BUS SELF-RESISTANCE, P.U.	r_i^b
RDEM	RV	REAL POWER DEMAND, MW	d_i
RGEN	R	REAL POWER GENERATED, MW	g_i
RMVABS	R	MVA BASE OF POWER SYSTEM DATA	
SIG1	R	BRANCH CURRENT SUM ONE, P.U.	
SIG2	R	BRANCH CURRENT SUM TWO, P.U.	
TAPNEW	R	NEW VALUE OF THE TRANSFORMER TAP, P.U.	n_k^{new}
TAP	RV	TRANSFORMER TAP, P.U. TURNS RATIO	n_k
TASK	A	NEXT TASK SELECTION VARIABLE	
TL	AV	TASK CODE LIST	
U	RV	REAL COMPONENT OF BUS VOLTAGE, P.U.	u_i
UIT	R	TEMPORARY REAL BUS VOLTAGE COMP., P.U.	
VISQ	R	BUS VOLTAGE SQUARED, P.U.	
VMOD	R	BUS VOLTAGE MODIFICATION FACTOR	
VSM	R	PRECOMPUTED VOLTAGE EXPRESSION, LINE FLOWS	
VSP	R	PRECOMPUTED VOLTAGE EXPRESSION, LINE FLOWS	
W	RV	IMAGINARY COMPONENT OF BUS VOLTAGE, P.U.	w_i
WIOLD	R	OLD IMAGINARY BUS VOLTAGE COMPONENT, P.U.	
WIT	R	TEMPORARY IMAGINARY BUS VOLTAGE COMP., P.U.	
X	R	INPUT REAC. OR SUSC. IN SUBR. ZYZ	x
XB	RV	BUS SELF-REACTANCE, P.U.	x_i^b
XDEM	RV	REACTIVE POWER DEMAND, MW	
XGEN	R	REACTIVE POWER GENERATED, MW	
ZSQD	R	INPUT IMP. OR ADM. SQUARED IN SUBR. ZYZ	

11.8 Source program listings

```
C          CMLFP
C          CONVERSATIONAL MODE LOAD FLOW PROGRAM
C          SOLVES NODAL EQUATIONS AND COMPUTES
C          BRANCH POWER FLOWS
C               PROGRAM            D.G. STONE
C               LANGUAGE           FORTRAN IV
C               MACHINE            PDP-9, 8K CORE MEMORY
C               MONITOR            I/O MONITOR, PAPER TAPE
C               DATE WRITTEN       1 JUNE 1968
C          MAXIMUM PROBLEM SIZE
C               BRANCHES  40
C               BUSSES  30
C               VOLTAGE-CONTROLLED BUSSES  20
C               TAP-CHANGING TRANSFORMERS  10
           DIMENSION TL(11)
           COMMON /BKSPAR/ LINES,NXFMRS,NDEMS,NBUSS,NGBUS,
     1RMVABS,NRBRAN,NRBT2,ITLIM,ISLACK
           COMMON /BKSCD/ IBSLOC(31),JA(80),KS(80),
     1RB(30),XB(30),G(40),B(40)
           COMMON /BKSOPD/ IGENB(20),P(30),Q(30),U(30),
     1W(30),BVMSET(20),TAP(10),RDEM(30),XDEM(30)
           DATA TL(1),TL(2),TL(3),TL(4),TL(5),TL(6),
     1TL(7),TL(8),TL(9),TL(10),TL(11)/2HCD,2HOD,2HGS,2HBR,
     22HPF,2HTT,2HSB,2HVM,2HGE,2HIL,2HDE/
C               NEXT TASK MESSAGE
50         WRITE(1,906)
906        FORMAT(5X,5HNEXT.)
C               INPUT THE NEXT TASK AND CHANGES
51         READ(2,907) TASK,IB,CHA1,CHA2
907        FORMAT(A2,I4,2F6.3)
C               FIND BUS TYPE FOR BUS DATA CHANGES
           CALL BUSTYP(IB,IBTYPE,IG)
C               TASK SELECTION LOGIC
           DO 52 ML=1,11
           IF(TASK.EQ.TL(ML)) GO TO (1,2,3,4,5,
     16,7,8,9,10,11),ML
52         CONTINUE
           GO TO 50
C               INPUT THE POWER SYSTEM CONSTANT DATA
1          PAUSE 1
C               SKIP TWO LINES ON THE TAPE
           READ(3,904)
904        FORMAT(/1X)
C               READ POWER SYSTEM PARAMETERS
           READ(3,902) LINES,NXFMRS,NBUSS,NGBUS,
     1MVABSE,NRBRAN,NRBT2,NBUP1
902        FORMAT(10I5)
           RMVABS=MVABSE
C               READ NETWORK TOPOLOGY TABLES
```

```
      READ(3,902)(IBSLOC(I),I=1,NBUP1)
      READ(3,902)(JA(LSB),KS(LSB),LSB=1,NRBT2)
C           READ BUS SELF IMPEDANCES
      READ(3,903)(RB(I),XB(I),I=1,NBUSS)
903   FORMAT(5X,2E14.6)
C           READ BRANCH ADMITTANCES
      READ(3,903) (G(K),B(K),K=1,NRBRAN)
C           INITIALIZE THE TRANSFORMER
C           TAP SETTINGS
      DO 201 KT=1,NXFMRS
201   TAP(KT)=1.0
C           INPUT THE POWER SYSTEM OPERATIONAL DATA
2     PAUSE 2
C           SKIP TWO LINES ON THE TAPE
      READ(3,904)
C           READ GENERATOR BUS NUMBERS
      READ(3,902)(IGENB(IG),IG=1,NGBUS)
      READ(3,904)
C           INPUT AND ASSEMBLE THE BUS
C           OPERATIONAL DATA
      DO 200 I=1,NBUSS
      READ(3,925) IB,RDEM(IB),XDEM(IB),RGEN,XGEN,BVMAG
925   FORMAT(I3,12X,4F6.1,F6.3)
C           COMPUTE THE P.U. IMPRESSED POWER
      P(IB)=(RGEN-RDEM(IB))/RMVABS
      Q(IB)=(XGEN-XDEM(IB))/RMVABS
C           CALCULATE STARTING VALUES FOR THE
C           BUS VOLTAGE COMPONENTS
      U(IB)=BVMAG
      W(IB)=0.
C           STORE GENERATOR BUS VOLTAGE
C           MAGNITUDES IN SHORT TABLE
      CALL BUSTYP(IB,IBTYPE,IG)
      IF(IBTYPE.GE.0) BVMSET(IG)=BVMAG
200   CONTINUE
      GO TO 50
C           MAKE CHANGES IN OPERATIONAL CONDITIONS
C           OR PROGRAM PARAMETERS
C           CHANGE TRANSFORMER TAPS
6     K=IB
      TAPNEW=CHA1
C           MODIFIES NETWORK ADMITTANCES ACCORDING
C           TO THE VALUE OF THE NEW TAP
      DO 300 LSB=1,NRBT2
C           SEARCH THE SIGNED BRANCH TABLE FOR +K
      IF(KS(LSB)-K) 300,31,300
C           LOCATE THE LV BUS OF TRANSFORMER K
31    J=JA(LSB)
      GO TO 32
300   CONTINUE
```

```
C               NKYK IS STORED AS G(K), B(K) AND
C               FORMULAS ARISE FROM THIS FACT
C               COMPUTE TRANSFORMER INDEX FOR SHORT TABLE
32          KT=K-LINES
            IF(KT.LE.0.OR.KT.GT.NXFMRS) GO TO 50
C               INVERT BUS SELF-IMPEDANCE
            CALL ZYZ(RB(J),XB(J),GB,BB)
C               CHANGE LV BUS SELF IMPEDANCE
            FACTOR=(TAPNEW*TAPNEW-TAP(KT)*TAP(KT))/TAP(KT)
            GB=GB+G(K)*FACTOR
            BB=BB+B(K)*FACTOR
C               RE-INVERT
            CALL ZYZ(GB,BB,RB(J),XB(J))
C               CHANGE THE BRANCH ADMITTANCE
            G(K)=G(K)*TAPNEW/TAP(KT)
            B(K)=B(K)*TAPNEW/TAP(KT)
            TAP(KT)=TAPNEW
            GO TO 51
C               CHANGE SLACK BUS POSITION
7           IF(IBTYPE.LT.0) GO TO 50
            ISLACK=IB
C               SET SLACK BUS VOLTAGE COMPONENTS
12          U(ISLACK)=BVMSET(IG)
            W(ISLACK)=0.
            GO TO 51
C               CHANGE BUS VOLTAGE MAGNITUDE
8           IF(IBTYPE.LT.0) GO TO 50
            BVMSET(IG)=CHA1
C               TEST FOR SLACK BUS
            IF(IBTYPE.EQ.0) GO TO 12
            GO TO 51
C               CHANGE BUS GENERATION
9           IF(IBTYPE.LT.0) GO TO 50
            P(IB)=(CHA1-RDEM(IB))/RMVABS
            GO TO 51
C               CHANGE ITERATIONS LIMIT
10          ITLIM=IB
            GO TO 51
C               CHANGE BUS DEMAND
11          P(IB)=P(IB)+(RDEM(IB)-CHA1)/RMVABS
            RDEM(IB)=CHA1
            Q(IB)=Q(IB)+(XDEM(IB)-CHA2)/RMVABS
            XDEM(IB)=CHA2
            GO TO 51
C               GLIMN AND STAGG NODAL ITERATIONS
3           CALL GANDS
            GO TO 50
C               OUTPUT BUS DATA AND RESULTS
C               PRINT HEADING
4           WRITE(1,941)
```

```
941       FORMAT(4X,3HBUS,3X,2HPL,5X,2HQL,5X,2HPG,
     15X,2HQG,6X,1HV,6X,5HTHETA)
C               CALCULATIONS FOR OUTPUT
          DO 400 I=1,NBUSS
          RGEN=P(I)*RMVABS+RDEM(I)
          XGEN=Q(I)*RMVABS+XDEM(I)
          BVANG=57.2958*ATAN(W(I)/U(I))
          BVMAG=SQRT(U(I)*U(I)+W(I)*W(I))
          WRITE(1,940) I,RDEM(I),XDEM(I),RGEN,XGEN,
     1BVMAG,BVANG
940       FORMAT(5X,I2,4F7.2,2F8.3)
400       CONTINUE
          GO TO 50
C               COMPUTE AND TABULATE THE BRANCH
C               POWER FLOWS
C               PRINT HEADING
5         WRITE(1,960)
960       FORMAT(5X,2HLN,6H SB EB,4X,2HPS,5X,2HPR,5X,2HQS,
     15X,2HQR)
C               MAIN LOOP TAKES EACH BRANCH IN TURN
          DO 600 K=1,NRBRAN
C               LOCATE BUSSES FOR BRANCH K
          DO 611 LSB=1,NRBT2
C               DETECT END BUS OF BRANCH K
          IF(KS(LSB).EQ.K) J=JA(LSB)
C               DETECT START BUS OF BRANCH K
          IF(KS(LSB).EQ.(-K)) I=JA(LSB)
611       CONTINUE
C               SET SEND-RECEIVE SWITCH
          IE=0
C               NORMAL BRANCH ADMITTANCES
          AA=1.
C               TEST FOR K A TRANSFORMER
62        IF(K.LE.LINES.OR.K.GT.LINES+NXFMRS) GO TO 61
C               ADJUST ADMITTANCES FOR OFF
C               NOMINAL TAPS
          KT=K-LINES
C               SENDING END IS HV SIDE
          AA=1./TAP(KT)
C               RECEIVING END IS LV SIDE
          IF(IE.EQ.1)  AA=TAP(KT)
C               PRECOMPUTE QUANTITIES
61        VSP=U(I)*U(J)+W(I)*W(J)
          VSM=U(I)*W(J)-U(J)*W(I)
          VISQ=U(I)*U(I)+W(I)*W(I)
C               REAL POWER, MW
          PI=(VISQ*G(K)*AA-G(K)*VSP+B(K)*VSM)*RMVABS
C               REACTIVE POWER, MVAR
          QI=(-VISQ*B(K)*AA+G(K)*VSM+B(K)*VSP)*RMVABS
C               END LOGIC
```

```
            IF(IE.EQ.1) GO TO 63
            M=I
            I=J
            J=M
            IE=1
            PS=PI
            QS=QI
            GO TO 62
63          PR=-PI
            QR=-QI
C                OUTPUT THE BRANCH POWER FLOW
            WRITE(1,961) K,J,I,PS,PR,QS,QR
961         FORMAT(4X,3I3,4F7.1)
600         CONTINUE
            GO TO 50
            END
```

```
      SUBROUTINE GANDS
C     SOLVES POWER SYSTEM NODAL EQUATIONS BY GLIMN
C     AND STAGG METHOD
C     QUANTITIES EXPRESSED IN RECTANGULAR COORDINATES
      COMMON /BKSPAR/ LINES,NXFMRS,NDEMS,NBUSS,NGBUS,
     1RMVABS,NRBRAN,NRBT2,ITLIM,ISLACK
      COMMON /BKSCD/ IBSLOC(31),JA(80),KS(80),
     1RB(30),XB(30),G(40),B(40)
      COMMON /BKSOPD/ IGENB(20),P(30),Q(30),U(30),
     1W(30),BVMSET(20),TAP(10),RDEM(30),XDEM(30)
C          ITERATIONS LOOP
      DO 520 ITERS=1,ITLIM
C          PRESET THE CONVERGENCE INDICATOR
      IDRCON=1
C          MAIN LOOP SOLVES EACH BUS IN TURN
      DO 510 I=1,NBUSS
C          BRANCH CURRENT SUMS CALCULATION
C          COMMON TO ALL BUS TYPES
      SIG1=0.
      SIG2=0.
C          CALCULATE SCAN INDEX LIMITS
      JSTART=IBSLOC(I)
      JFINIS=IBSLOC(I+1)-1
C          SCAN BUSSES CONNECTED TO BUS I
      DO 501 JSCAN=JSTART,JFINIS
C          FIND FAR BUS NUMBER
      J=JA(JSCAN)
C          FIND BRANCH NUMBER
      K=IABS(KS(JSCAN))
C          FORM BRANCH CURRENT SUMS
      SIG1=SIG1+G(K)*U(J)-B(K)*W(J)
      SIG2=SIG2+B(K)*U(J)+G(K)*W(J)
501   CONTINUE
C          INITIALIZE THE LOAD BUS REPETITION INDR.
      M=-1
C          INITIALIZE TEMPORARY BUS VOLTAGE
      UIT=U(I)
      WIT=W(I)
      VISQ=U(I)*U(I)+W(I)*W(I)
C          FIND BUS TYPE
      CALL BUSTYP(I,IBTYPE,IG)
      IF(IBTYPE.GE.0) GO TO 3
C          VOLTAGE CALCULATIONS
C          LOAD BUSSES AND GENERATOR BUSSES
6     BRAK1=P(I)*UIT+Q(I)*WIT
      BRAK2=P(I)*WIT-Q(I)*UIT
C          REAL BUS VOLTAGE COMPONENT
      UIT=(RB(I)*BRAK1-XB(I)*BRAK2)/VISQ
     1+RB(I)*SIG1-XB(I)*SIG2
C          IMAGINARY BUS VOLTAGE COMPONENT
```

```
          WIT=(RB(I)*BRAK2+XB(I)*BRAK1)/VISQ
     1+RB(I)*SIG2+XB(I)*SIG1
C              INCREMENT THE LOAD BUS REPETITION INDR.
          M=M+1
          VISQ=UIT*UIT+WIT*WIT
C              BUS TYPE AND REPETITIONS TEST
          IF(IBTYPE+M) 6,5,7
C              BUS INJECTED POWER CALCULATIONS
C              GET ADMITTANCE FROM STORED IMPEDANCE
3         CALL ZYZ(RB(I),XB(I),GB,BB)
C              CALCULATE REACTIVE POWER FOR ALL GENERATORS
          Q(I)=-VISQ*BB-W(I)*SIG1+U(I)*SIG2
C              BUS TYPE TEST
          IF(IBTYPE) 15,15,6
C              CALCULATE REAL POWER FOR SLACK GENERATOR
15        P(I)=VISQ*GB-U(I)*SIG1-W(I)*SIG2
          GO TO 510
C              MODIFY VOLTAGE MAGNITUDE FOR GENERATOR BUSSES
7         VMOD=BVMSET(IG)/SQRT(VISQ)
          UIT=UIT*VMOD
          WIT=WIT*VMOD
C              ACCELERATE THE BUS VOLTAGE COMPONENTS
C              TO GET FINAL VALUES FOR THIS BUS
C              SAVE OLD VOLTAGE COMPONENT
5         WIOLD=W(I)
          U(I)=U(I)+(UIT-U(I))*1.6
          W(I)=W(I)+(WIT-W(I))*1.6
C              CONVERGENCE TEST ON EACH BUS VOLTAGE
          IF(ABS(W(I)-WIOLD).GT.1.0E-6) IDRCON=0
C              BOTTOM OF THE MAIN LOOP
510       CONTINUE
C              TEST FOR OVERALL CONVERGENCE
          IF(IDRCON) 520,520,14
520       CONTINUE
          RETURN
C              PRINT NUMBER OF ITERATIONS TO CONVERGE
14        WRITE(1,968) ITERS
968       FORMAT(5X,I3)
          RETURN
          END
```

```
      SUBROUTINE BUSTYP(IB,IBTYPE,IG)
C     IDENTIFIES THREE TYPES OF BUSSES
C     MAKES CORRESPONDENCE BETWEEN BUS NUMBER
C     IB AND GENERATOR BUS TABLE INDEX IG
      COMMON /BKSPAR/ LINES,NXFMRS,NDEMS,NBUSS,NGBUS,
     1RMVABS,NRBRAN,NRBT2,ITLIM,ISLACK
      COMMON /BKSOPD/ IGENB(20),P(30),Q(30),U(30),
     1W(30),BVMSET(20),TAP(10),RDEM(30),XDEM(30)
C          BUS IDENTIFICATION LOGIC
C          SCAN GENERATOR BUS LIST
      DO 800 IG=1,NGBUS
      IF(IGENB(IG).EQ.IB) GO TO 1
800   CONTINUE
C          SET BUS TYPE TO LOAD BUS
      IBTYPE=-1
      RETURN
C          SET BUS TYPE TO GENERATOR BUS
C          IG CORRESPONDING TO IB IS NOW AVAILABLE
1     IBTYPE=1
C          TEST AND SET FOR SLACK GENERATOR BUS
      IF(IGENB(IG).EQ.ISLACK) IBTYPE=0
      RETURN
      END



      SUBROUTINE ZYZ(R,X,G,B)
C     INVERTS EITHER IMPEDANCE OR ADMITTANCE
C     R AND X ARE INPUT VARIABLES, G AND B ARE OUTPUT
C     VARIABLES, EITHER IMPEDANCE OR ADMITTANCE
      ZSQD=R*R+X*X
      G=R/ZSQD
      B=-X/ZSQD
      RETURN
      END
```

12 Machine dependencies

The device assignment numbers in the FORTRAN input/output statements are

1 teletype output
2 teletype keyboard input
3 paper tape reader

Teletype KSR33 prints at 10 characters per second and has 72 print positions, 50 of which are used by CMLFP. The first character for output by a FORMAT statement is not printed, but is used for output device control. CMLFP contains no such character other than blank, which signifies normal one-line vertical spacing.

Computations are done in 6 digit single-precision floating point arithmetic, with two machine words per FORTRAN real variable. An integer variable takes one machine word. No failures of the convergence criterion were observed, despite the fact that its value is equivalent to a single unit in the least significant digit of the voltage component being tested.

Complex variable features are not available in PDP-9 FORTRAN, so the entire program is written in component form. Both arithmetic and logical IF statements are present in the program.

The machine-oriented details in the loading and operating instructions naturally depend on the computer used. Also, the monitor commands ↑P, ↑S, RUBOUT, and RETURN will have different counterparts on another machine.

13 Program modification

The PDP-9's 8K core memory is almost completely utilized by CMLFP. Marginal increases, or changes, in maximum problem size are possible by changing the dimensions of the arrays in the COMMON blocks. Some useful features could be added to CMLFP with only a

modest increase in available core storage, among them:

1 Variable convergence criterion and acceleration factor input from the teletype keyboard.
2 Keyboard selection of specific busses or branches to be output.
3 Ability to punch OD tapes of solutions for later input as non-flat starts.
4 Branch outage studies on-line from the keyboard.
5 When a command is incorrect, print ERROR. instead of NEXT.
6 Transmission line reactive power flows that include the line charging MVARs.

More flexible power system data structures would require many detailed changes. The logical structure of CMLFP main program may be made more evident by the removal of the imbedded subroutines. Program PSDAP can be incorporated into CMLFP as a subroutine.

14 Checkout status

CMLFP has been used by its author over a period of six months. Several demonstrations and practice sessions have been held for power system engineers. A data file of 12 systems is kept.

The tap-changing program segment was checked in detail by hand, using the 6-bus power system of reference 3. Also, the converged solutions for this system agreed to within machine accuracy.

The AEP14 test system solution checked satisfactorily but not precisely, because the results given with the test data are after only 17 iterations, and the convergence rates were not the same. The converged CMLFP solution at bus 6 was hand-checked on a 6-digit Curta calculator, using different (reference 6) expressions and intermediate data. Agreement to machine accuracy was observed.

There are 122 possible command-pairs, excluding repetitions. Only the most common operational mistakes have been protected against or tested. No attempt has been made to certify the program by

exhausting all its logical paths.

Parallel branches may be kept separate and CMLFP will compute the correct power flow in each. The program can also simulate an individual transmission line or transformer.

15 References

1 'American Standard FORTRAN, X3.9-1966,' American Standards Association, March 7, 1966.

2 'FORTRAN IV, Advanced Software System, Programmer's Reference Manual, DEC-9A-AF4B-D,' Digital Equipment Corporation, 1968.

3 WARD, J.B. and HALE, H.W.: 'Digital Computer Solution of Power-Flow Problems,' Trans. AIEE, 1956, vol. 75, pt. III, pp. 398-405.

4 GLIMN, A.F. and STAGG, G.W.: 'Automatic Calculation of Load Flows,' Trans. AIEE, 1957, vol. 76, pt. III, pp. 817-828.

5 LAUGHTON, M.A. and HUMPHREY DAVIES, M.W.: 'Numerical Techniques in Solution of Power-System Load-Flow Problems,' Proc. IEE, 1964, vol. 111, (9), pp. 1575-1588.

6 STEVENSON, W.D. Jr.: 'Elements of Power System Analysis' (McGraw-Hill, 1962).

Program title: Power System Data Assembly Program

Program name: PSDAP

Author: D.G. Stone

Source language: FORTRAN IV

Machine: *

Monitor: *

Date written: *

Source program length:

	text lines	FORTRAN statements
Program PSDAP	122	78
Subroutine ZYZ	9	6
totals	131	84

Media: *

Required routines: None.

Location: *

* as for CMLFP

TABLE OF CONTENTS

1 Purpose
2 Method
3 Description
4 Usage
  4.1 Tapes required
  4.2 Loading instructions
  4.3 Operating instructions
  4.4 Messages
  4.5 Input data preparation
  4.6 Output
  4.7 Run requirements diagram
5 Example
  5.1 Teletype output
  5.2 Basic data tape listing
  5.3 Constant data tape listing
6 Restrictions
7 Accuracy
8 Data checking
9 Storage
10 Execution time
11 Program details
  11.1 Main program PSDAP
  11.2 Subroutine ZYZ
  11.3 Flowchart, PSDAP
  11.4 Variables list
  11.5 Source program listing
  11.6 Network topology tables
12 Machine dependencies
13 Program modification
14 Checkout status
15 References

1 Purpose

To assemble power system data in the form required by the Conversational Mode Load Flow Program.

2 Method

All program operations are elementary. PSDAP does not make use of a numerical method.

3 Description

Network immittances and topology are stored as tables designed for efficient utilization of computer core memory and high speed of computation. The branch series admittance per phase in p.u., Y_k, is obtained from

$$Y_k = g_k + jb_k = \frac{1}{Z_k} \quad , \quad Z_k = r_k + jx_k$$

Most of the variables and indicies in this section are defined in 11.4 Variables list. As each branch is read in, its contribution to the bus self-admittances at each end is made. Finally, at bus i

$$Y_i^b = \sum_{k \in B_i} (Y_k + jb_k^s) + jb_i^s = g_i^b + jb_i^b$$

Transmission line total shunt susceptance per phase b_{TL} is an input quantity, from which the pi model leg is calculated

$$b_k^s = b_{TL}/2 \qquad \text{line k}$$

B_i is the set of all branches connected to bus i. The total susceptance per phase of the shunt equipment located at bus i is b_i^s. Rather than Y_i^b, the bus self-impedance Z_i^b is chosen as the stored variable for the nodal iterations.

$$Z_i^b = r_i^b + jx_i^b = \frac{1}{Y_i^b}$$

Let branch k have near bus i and far bus j. CMLFP has three operations requiring machine implementation of the network topology:

	given	find	speed
1 Notal iterations	i	k,j	high
2 Tap changing	k	j	low
3 Power flow	k	i,j	low

4 Usage

Concurrent reference to section 5 Example is suggested.

4.1 Tapes required

1 A binary object program tape containing the compiled versions of program PSDAP and subroutine ZYZ.

2 A power systems basic data (BD) tape, made up according to section 4.5 Input data preparation.

3 Linking loader, and subroutine library tapes.

4.2 Loading instructions

These are given in the CMLFP documentation. At step 9, type only one comma.

4.3 Operating instructions

1 Hold down the CTRL key on the teletype keyboard, and press the letter S.

2 Machine types back ↑S, and then the program title
PSDAP followed directly by
PAUSE ØØØØ1Ø

3 Put the power system basic data (BD) tape in the paper tape reader, pushing the reader feed button momentarily.

4 Hold CTRL, type P.

5 ↑P is echoed by the computer, followed by reading and typing of the power system title, and subsequently the entire BD tape is read in.

6 PSDAP then assembles the data in store, and indicates that it is ready to punch the power system constant data (CD) tape by typing
PAUSE ØØØØ11

7 Check that the punch is properly loaded with paper tape and that its POWER light is on.

8 Hold CTRL down, and press P.

9 The CD tape is punched. Machine types
FINISHED.
some line spaces, and then
PAUSE ØØØØ1Ø

10 Tear off the CD tape and run out some new leader by pushing the punch FEED button.

11 To prepare the CD tape for a new power system, repeat from step 3.

4.4 Messages

PSDAP generates no messages itself, other than those given in the previous subsection. Reference 2 in the CMLFP documentation contains a list of Object Time System error messages.

4.5 Input data preparation

Text-line 1 of the basic data (BD) tape contains the power system title, and must be made up of at least 60 characters, including blanks. Line 2 is a non-processed heading inserted to improve the readability of the BD tape listing. Line 3 holds the power system size parameters, read according to FORMAT 960. These are, in order:

1 transmission lines
2 transformers
3 generator sets (not read)
4 demands (not read)
5 busses, all types
6 voltage-controlled (generator) busses

7 series reactors

8 shunt equipments

9 MVA base

Lines 4 and 5 are again non-processed headings. They are followed by the branch data, one branch per text line, conforming to FORMAT 952. The branch data items on each line are:

1 branch number

2 start bus number

3 end bus number

4 total series resistance per phase, in per cent on the specified MVA base, and on the nominal KV base

5 total series reactance per phase

6 total shunt susceptance per phase (not the pi leg element in the transmission line model)

For a network of m branches and n busses, the branches must be numbered with the first m positive integers, and the busses with the first n. These numberings are not related by any additional rule. The branch orientation is arbitrary, except in the case of transformers, where the start bus must be its high voltage bus. Also the branch numbering must be such that the lines come first, followed by the transformers, and finally the series reactors. On the BD tape the branches must appear in numerical order.

If there is no shunt equipment, the final branch is the last text line on the BD tape; otherwise the final branch is followed by two non-processed heading lines and then the shunt equipment data. This latter follows FORMAT 958. The first item is the bus number, and the second the shunt susceptance per phase of the reactor or capacitor in per cent.

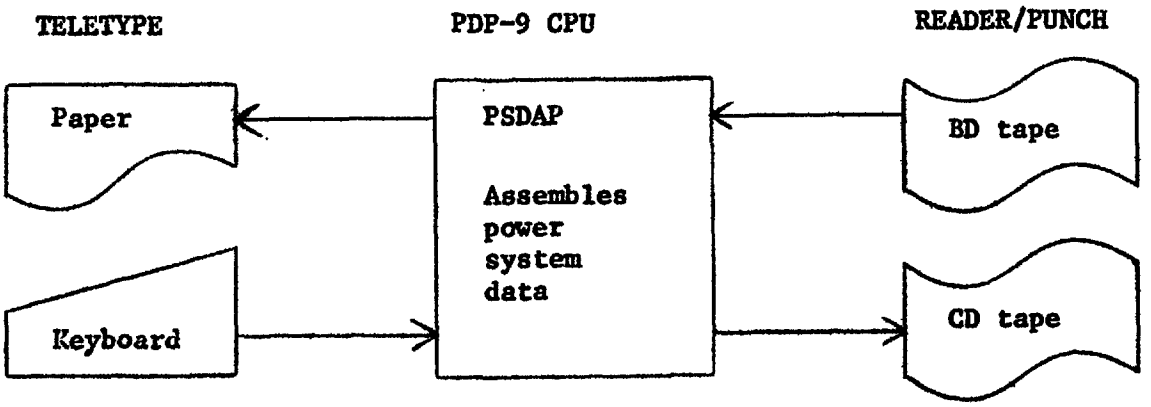
4.6 Output

Apart from that described in subsection 4.3, there is no teletype output from PSDAP. The power system constant data (CD)

tape produced by the program is punched in BCD so that a listing of it may be read by eye. This tape consists of, in order:

1 power system title
2 a blank text line
3 a line containing some power system parameters
4 lines comprising the bus location table, IBSLOC
5 tables JA and KS, pairwise
6 bus self-impedances, p.u.
7 branch series admittances, p.u.

4.7 Run requirements diagram



5 Example

Both CMLFP and PSDAP documentations have as their example AEP14, the American Electric Power 14 Bus Test System. It has 15 transmission lines, and 3 tap-changing transformers. One of the latter is modelled by an equivalent T which includes series reactors 19 and 20, and bus 7. There are 15+3+2=20 branches. Only busses 1 and 2 can generate real power, but there are in addition 3 synchronous capacitors, making a total of 5 voltage-controlled (generator) busses. A static shunt capacitance of 19.0 per cent is located at bus 9. AEP14's system diagram appears in the CMLFP documentation as subsection 5.3.

5.1 Teletype output

```
LOADER

>,
 PSDAP   16261
 ZYZ     16202
↑P↑P
 PPA.    15431
↑P↑P
 .DA     15362
↑P↑P
 BCDIO   12367
 .SS     12310
 PAUSE   12274
 SPMSG   12201
 FIOPS   11445
 OTSER   11337
 INTEGE  11153
 REAL    10220
↑S↑S

PSDAP


PAUSE 000010
↑P
AEP14 AMERICAN ELECTRIC POWER  14 BUS SYSTEM

PAUSE 000011
↑P
FINISHED.




PAUSE 000010
```

5.2 Basic data tape listing

```
AEP14 AMERICAN ELECTRIC POWER  14 BUS SYSTEM
 LINS XFMS GENS DEMS BUSS GBUS SERE SHRE MVAB
   15    3    2   11   14    5    2    1  100
SERIES BRANCH DATA, P.C. ON 100MVA BASE
 BR  SB  EB R     X      B
  1   1   2 1.94 5.92   5.28
  2   1   5 5.40 22.30  4.92
  3   2   3 4.70 19.80  4.38
  4   2   4 5.81 17.63  3.74
  5   2   5 5.70 17.39  3.40
  6   3   4 6.70 17.10  3.46
  7   4   5 1.34 4.21   1.28
  8   6  11 9.50 19.89  0.
  9   6  12 12.29 25.58  0.
 10   6  13 6.62 13.03  0.
 11   9  10 3.18 8.45   0.
 12   9  14 12.71 27.04  0.
 13  10  11 8.21 19.21  0.
 14  12  13 22.09 19.99  0.
 15  13  14 17.09 34.80  0.
 16   4   7 0.    20.91
 17   4   9 0.    55.62
 18   5   6 0.    25.20
 19   7   8 0.    17.62
 20   7   9 0.    11.00
SHUNT REACTANCE DATA
 BN          BSHUNT
  9           19.0
```

5.3 Constant data tape listing

```
AEP14 AMERICAN ELECTRIC POWER  14 BUS SYSTEM

   15     3    14     5   100    20    40    15
    1     3     7     9    14    18    22    25    26    30
   32    34    36    39    41
    2     1     5     2     1    -1     3     3     4     4
    5     5     2    -3     4     6     2    -4     3    -6
    5     7     7    16     9    17     1    -2     2    -5
    4    -7     6    18    11     8    12     9    13    10
    5   -18     4   -16     8    19     9    20     7   -19
   10    11    14    12     4   -17     7   -20     9   -11
   11    13     6    -8    10   -13     6    -9    13    14
    6   -10    12   -14    14    15     9   -12    13   -15
    1  0.145463E-01  0.469355E-01
    2  0.946188E-02  0.300697E-01
    3  0.294433E-01  0.925538E-01
    4  0.668057E-02  0.242767E-01
    5  0.731592E-02  0.266297E-01
    6  0.191339E-01  0.504142E-01
    7  0.000000E+00  0.511544E-01
    8  0.000000E+00  0.176200E+00
    9  0.874544E-02  0.395713E-01
   10  0.229891E-01  0.587134E-01
   11  0.441530E-01  0.977716E-01
   12  0.880715E-01  0.119080E+00
   13  0.422901E-01  0.670825E-01
   14  0.729185E-01  0.152179E+00
    1  0.499871E+01 -0.152538E+02
    2  0.102574E+01 -0.423592E+01
    3  0.113491E+01 -0.478111E+01
    4  0.168614E+01 -0.511648E+01
    5  0.170199E+01 -0.519256E+01
    6  0.198636E+01 -0.506967E+01
    7  0.686486E+01 -0.215680E+02
    8  0.195529E+01 -0.409375E+01
    9  0.152599E+01 -0.317614E+01
   10  0.309917E+01 -0.610004E+01
   11  0.390113E+01 -0.103662E+02
   12  0.142376E+01 -0.302899E+01
   13  0.188118E+01 -0.440164E+01
   14  0.248882E+01 -0.225222E+01
   15  0.113698E+01 -0.231520E+01
   16  0.000000E+00 -0.478240E+01
   17  0.000000E+00 -0.179791E+01
   18  0.000000E+00 -0.396825E+01
   19  0.000000E+00 -0.567537E+01
   20  0.000000E+00 -0.909091E+01
```

6 Restrictions

1 The power system basic data tape must be prepared as specified in subsection 4.5.

2 The FORTRAN arrays have been conservatively dimensioned, allowing a maximum problem size of:

- 30 busses, including voltage-controlled busses
- 40 branches, including lines, transformers, and series reactors
- 20 voltage-controlled busses
- 10 tap-changing transformers

7 Accuracy

The line parameters R, X, and B are commonly given to an engineering accuracy of 3 or 4 significant digits. Six decimal digit accuracy is maintained in the program by the PDP-9's single precision arithmetic package. All six digits are punched on the constant data tape by an E specification in FORMAT 965.

8 Data checking

None.

9 Storage

PSDAP, together with subroutine ZYZ occupies $797_{10} = 1435_8$ core memory locations. The arrays in the COMMON area overlay part of the Linking Loader at run time, taking $695_{10} = 1267_8$ locations. Library subroutines loaded to service PSDAP require $3058_{10} = 5762_8$ machine locations.

10 Execution time

Internal processing is almost instantaneous as far as the human operator is aware; so that total program running time is

determined by the input/output task. For AEP14, 15 seconds was required to read in the BD tape, and the corresponding CD tape was punched in 50 seconds.

11 Program details

The structure of PSDAP is presented in the form of a flow-chart, subsection 11.3, and its source program listing is in 11.5.

11.1 Main program PSDAP

The two skipped fields in FORMAT 960 contain parameters not required by CMLFP. FORMATs 953 and 954 skip non-processed headings on the BD tape. Variables NRBRAN, NRBT2, and NBUP1 are final values of DO-loop indicies.

As each branch is read in, (DO-loop 402) its start bus and end bus are stored, so that later in the program the network topology tables can be constructed. Also stored is the branch series admittance, calculated by subroutine ZYZ from the branch series impedance. The bus self-admittances are built up branch by branch so that when DO 402 is satisfied, they are complete, except for the addition of the shunt equipment susceptances. This is done, if needed, in DO 406. Once the bus self-admittances have been assembled, DO-loop inverts them using subroutine ZYZ to form the bus self-impedances.

Nested DO-loops 410 and 409 build the network topology tables IBSLOC, KS, and JA by scanning the start bus and end bus numbers of each branch. Subsection 11.6 gives the network topology tables for the example, AEP14. The signed branch table, KS, begins by listing the set of branches connected to bus 1; assigning radiating branches as positive and impinging branches as negative. JA, the far bus table, contains the corresponding busses. Next the branches (and their busses) connected to bus 2 are tabulated, and so on until the tables are complete. Each branch appears twice in the KS

table, once positive, once negative. A means for referencing these sets of branches is provided by IBSLOC, the bus location table. In the example, bus 4 begins in position 9 and has 5 branches connected to it, since bus 5 begins in position 14. The final scanning limit is necessary to indicate the number of branches connected to the last bus.

With all data assembly operations complete, a BCD tape of the power system constant data is punched out as described in subsection 4.6.

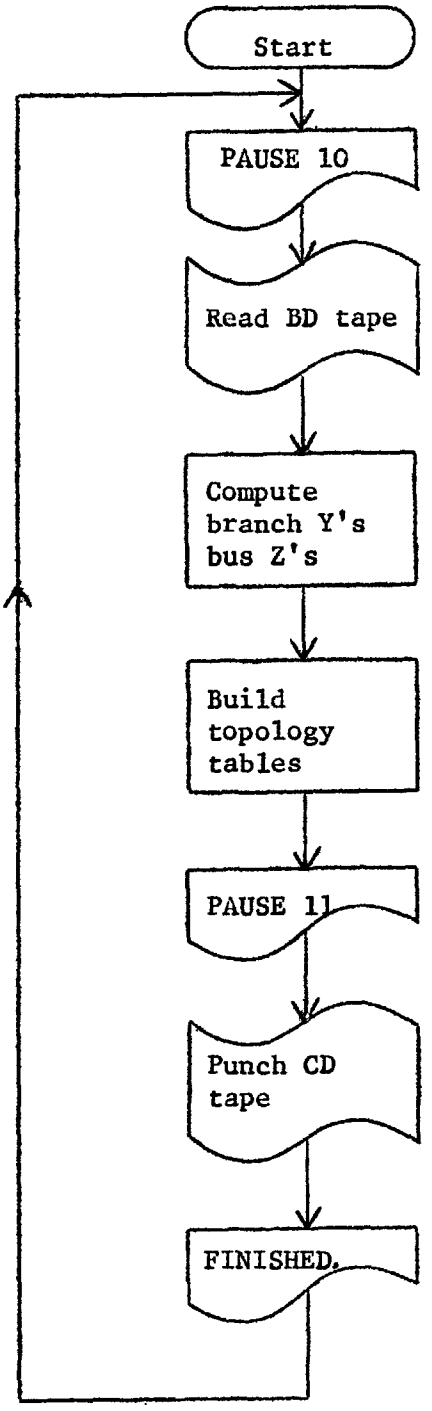
There is no STOP statement in the program, control transferring to statement 1 when a tape is finished.

ZYZ is the only subroutine called by PSDAP.

11.2 Subroutine ZYZ

Documented with CMLFP.

11.3 Flowchart PSDAP



11.4
VARIABLES LIST

*
PSDAP POWER SYSTEM DATA ASSEMBLY PROGRAM
*

NAME OF FORTRAN VARIABLE	DESCRIPTION I INTEGER, R REAL, A ALPHABETIC, V VECTOR ARRAY			
*				
B	RV	BRANCH SERIES SUSCEPTANCE, P.U.	b_k	b_i^b
BB	RV	BUS SELF-SUSCEPTANCE, P.U.		
BSHUNT	R	SHUNT SUSCEPTANCE ELEMENT, P.C.	b_i^s	
BTL	R	TOTAL LINE SHUNT SUSCEPTANCE, P.C.		b_{TL}
G	RV	BRANCH SERIES CONDUCTANCE, P.U.	g_k	
GB	RV	BUS SELF-CONDUCTANCE, P.U.		
I	I	BUS INDEX	i	
IB	I	BUS NUMBER	i	
IBSLOC	IV	BUS LOCATION TABLE		
IEB	IV	END BUS OF BRANCH K		
ISB	IV	START BUS OF BRANCH K		
J	I	BUS INDEX	j	
JA	IV	FAR BUS TABLE		
K	I	BRANCH INDEX	k	
KS	IV	SIGNED BRANCH TABLE		
LINES	I	NUMBER OF LINES IN THE NETWORK		
LB	I	BRANCH NUMBER		
LSB	I	LOCATION SIGNED BRANCH		
MVABSE	I	MVA BASE OF POWER SYSTEM DATA		
NBUP1	I	NUMBER OF BUSSES PLUS ONE		
NBUSS	I	NUMBER OF BUSSES IN THE NETWORK		
NGBUS	I	NUMBER OF VOLTAGE-CONTROLLED BUSSES		
NRBRAN	I	NUMBER OF BRANCHES IN THE NETWORK		
NRBT2	I	NUMBER OF BRANCHES TIMES TWO		
NSERC	I	NUMBER OF SERIES REACTORS		
NSHNT	I	NUMBER OF SHUNT EQUIPMENT ITEMS		
NXFMRS	I	NUMBER OF TRANSFORMERS IN THE NETWORK		
R	R	TOTAL BRANCH SERIES RESISTANCE, P.C.	r_k	
RB	RV	BUS SELF-RESISTANCE, P.U.		r_i^b
RMVABS	R	MVA BASE OF POWER SYSTEM DATA		
TITLE	AV	POWER SYSTEM TITLE		
X	R	TOTAL BRANCH SERIES REACTANCE, P.C.	x_k	
XB	RV	BUS SELF-REACTANCE, P.U.		x_i^b

11.5 Source program listing

```
C          PSDAP
C          POWER SYSTEM DATA ASSEMBLY PROGRAM
C          INPUTS THE POWER SYSTEM BASIC DATA, ASSEMBLES IT,
C          AND PUNCHES A PAPER TAPE FOR INPUT TO CONVERSATIONAL
C          MODE LOAD FLOW PROGRAM.
           COMMON TITLE(12),GB(30),BB(30),ISB(40),IEB(40),
     1G(40),B(40),RB(30),XB(30),IBSLOC(31),
     1KS(80),JA(80)
C               PRINT THE PROGRAM TITLE
           WRITE(1,959)
959        FORMAT(/1X,5HPSDAP/)
1          PAUSE 10
           WRITE(1,953)
C               READ AND PRINT THE POWER SYSTEM TITLE
           READ(3,950)(TITLE(I),I=1,12)
950        FORMAT(12A5)
           WRITE(1,951)(TITLE(I),I=1,12)
951        FORMAT(1X,12A5)
C               SKIP ONE HEADING LINE
           READ(3,953)
953        FORMAT(1X)
C               READ POWER SYSTEM PARAMETERS
           READ(3,960) LINES,NXFMRS,NBUSS,NGBUS,NSERC,
     1NSHNT,MVABSE
960        FORMAT(2I5,10X,5I5)
           NRBRAN=LINES+NXFMRS+NSERC
           NRBT2=NRBRAN*2
C               SKIP TWO HEADING LINES
           READ(3,954)
954        FORMAT(/1X)
C               ZERO THE BUS SELF-ADMITTANCES
           DO 403 I=1,NBUSS
           GB(I)=0.
403        BB(I)=0.
C               INPUT AND ASSEMBLE THE BRANCH DATA
           DO 402 K=1,NRBRAN
           READ(3,952) LB,ISB(LB),IEB(LB),R,X,BTL
952        FORMAT(I3,2I4,F5.2,F6.2,F7.2)
C               CONVERT FROM P.C. TO P.U.
           RMVABS=MVABSE
           R=R/~~RMVABS~~ 100.
           X=X/~~RMVABS~~ 100.
C               INVERT THE BRANCH SERIES IMPEDANCE TO GET
C               SERIES ADMITTANCE CIRCUIT ELEMENT VALUES
C               NOT THE MUTUAL ADMITTANCE
           CALL ZYZ(R,X,G(LB),B(LB))
C               PI MODEL SHUNT SUSCEPTANCE IS 1/2 THE TOTAL
C               LINE SHUNT SUSCEPTANCE
           BTL=BTL/(2.*RMVABS)
C               SUMMING THE BUS SELF ADMITTANCES
```

```
      I=ISB(LB)
      J=IEB(LB)
      GB(I)=GB(I)+G(LB)
      GB(J)=GB(J)+G(LB)
      BB(I)=BB(I)+B(LB)+BTL
      BB(J)=BB(J)+B(LB)+BTL
402   CONTINUE
C           TEST FOR SHUNT EQUIPMENT
      IF(NSHNT.EQ.0) GO TO 6
C           SKIP TWO HEADINGS
      READ(3,954)
C           INPUT AND ASSEMBLE THE SHUNT EQUIPMENT DATA
      DO 406 I=1,NSHNT
      READ(3,958) IB,BSHUNT
958   FORMAT(I3,11X,F7.2)
C           CONVERT FROM P.C. TO P.U.
      BSHUNT=BSHUNT/RMVABS
C           ADD TO BUS SELF ADMITTANCE
      BB(IB)=BB(IB)+BSHUNT
406   CONTINUE
C           INVERT THE BUS SELF ADMITTANCES TO GET SELF
C           IMPEDANCES FOR LOAD FLOW EXPRESSIONS
6     DO 408 I=1,NBUSS
      CALL ZYZ(GB(I),BB(I),RB(I),XB(I))
408   CONTINUE
C           CONSTRUCT THE NETWORK TOPOLOGY TABLES
      LSB=1
C           TAKE EACH BUS IN TURN
      DO 410 I=1,NBUSS
C           BUS I IS AT LOCATION LSB
      IBSLOC(I)=LSB
C           SCAN FOR BRANCH K CONNECTED TO BUS I
      DO 409 K=1,NRBRAN
C           TEST FOR RADIATING BRANCHES
      IF(ISB(K).NE.I) GO TO 12
      KS(LSB)=+K
      JA(LSB)=IEB(K)
      GO TO 15
C           TEST FOR IMPINGING BRANCHES
12    IF(IEB(K).NE.I) GO TO 409
      KS(LSB)=-K
      JA(LSB)=ISB(K)
C           INCREMENT THE KS, JA TABLE INDEX LSB
15    LSB=LSB+1
409   CONTINUE
410   CONTINUE
C           ADD THE FINAL SCANNING LIMIT
      NBUP1=NBUSS+1
      IBSLOC(NBUP1)=LSB
```

```
      PAUSE 11
      WRITE(1,953)
C           PUNCHING ASSEMBLED DATA TAPE FOR CMLFP
C           PUNCH THE POWER SYSTEM TITLE
      WRITE(5,951)(TITLE(I),I=1,12)
      WRITE(5,953)
C           PUNCH THE POWER SYSTEM PARAMETERS
      WRITE(5,964) LINES,NXFMRS,NBUSS,NGBUS,
     1MVABSE,NRBRAN,NRBT2,NBUP1
964   FORMAT(1X,10I5)
C           PUNCH NETWORK TOPOLOGY TABLES
      WRITE(5,964)(IBSLOC(I),I=1,NBUP1)
      WRITE(5,964)(JA(LSB),KS(LSB),LSB=1,NRBT2)
C           PUNCH BUS QUANTITIES
      WRITE(5,965)(I,RB(I),XB(I),I=1,NBUSS)
965   FORMAT(1X,I5,2E14.6)
C           PUNCH BRANCH QUANTITIES
      WRITE(5,965)(K,G(K),B(K),K=1,NRBRAN)
C           PRINT MESSAGE
      WRITE(1,963)
963   FORMAT(1X,9HFINISHED.//////)
      GO TO 1
      END
```

11.6 Network topology tables

AEP14

I	IBSLOC(I)
1	1
2	3
3	7
4	9
5	14
6	18
7	22
8	25
9	26
10	30
11	32
12	34
13	36
14	39
15	41

I	LSB	KS(LSB)	JA(LSB)
1	1	1	2
	2	2	5
2	3	-1	1
	4	3	3
	5	4	4
	6	5	5
3	7	-3	2
	8	6	4
4	9	-4	2
	10	-6	3
	11	7	5
	12	16	7
	13	17	9
5	14	-2	1
	15	-5	2
	16	-7	4
	17	18	6
6	18	8	11
	19	9	12
	20	10	13
	21	-18	5
7	22	-16	4
	23	19	8
	24	20	9
8	25	-19	7
9	26	11	10
	27	12	14
	28	-17	4
	29	-20	7
10	30	-11	9
	31	13	11
11	32	-8	6
	33	-13	10
12	34	-9	6
	35	14	13
13	36	-10	6
	37	-14	12
	38	15	14
14	39	-12	9
	40	-15	13

12 Machine dependencies

The FORTRAN input/output device assignment numbers used by PSDAP are:

1 teletype output

3 paper tape reader

5 paper tape punch

Up to 5 alphameric characters can be stored per FORTRAN word, thus FORMAT 950 contains an A5 specification.

13 Program modifications

Because of the interdependence of PSDAP and CMLFP, these are discussed in the latter document.

14 Checkout status

The branch series admittances of AEP14 checked satisfactorily against those provided with the test data. Hand calculation of the bus self-impedances revealed no errors. The small test system in reference 3 (see CMLFP) provided another check on the correct assembly of power system immittances.

15 References

As for CMLFP.

Appendix 3 PROGRAM DESCRIPTION: NFESP, NETWORK FLOW ECONOMIC SCHEDULING PROGRAM

The main program of NFESP has but six statements: four subroutine CALLs, STOP, and END. Each of the major subroutines PSDNFM, ESMANF, TOPSDR, and DIGRAM is a functional module, and corresponds to an overlay segment. The program's modular design permits efficient core utilization. Only the programs and data needed at each step in the calculations are resident in the machine core storage, the remainder being kept out on backing store until required.

Because the program is heavily interspersed with comment cards, its logic can be followed without the need for a supplementary narrative. The call to MULTAR, the network flow routine, is located in subroutine ESMANF. Subroutines GMODEL and CONGEN build and interpret generator models for British sets. The corresponding routines for American generating units are not included.

D.W. Norris, a co-student, wrote subroutines COSPLT and PLTOUI. They are line printer graphics utility routines, producing plots of data and results.

```
C     NFESP
C     MAIN PROGRAM. ECONOMIC SCHEDULING OF ELECTRIC POWER
C     SYSTEMS USING FULKERSONS OUT-OF-KILTER NETWORK FLOW
C     ALGORITHM.
C     MULTIPLE PARALLEL ARCS USED TO GIVE PIECEWISE LINEAR
C     APPROXIMATION TO QUADRATIC COST CURVES.
C          PROGRAM       - D.G. STONE
C          LANGUAGE      - FORTRAN IV
C          SYSTEM        - IBM7094, IBSYS, IMPERIAL COLLEGE
C          DATE WRITTEN  - 15 MARCH 1968.
C     OVERLAY, LINKING, AND SCRATCH DATA TAPES USED TO
C     SAVE CORE
C          INPUT AND ASSEMBLY OF POWER SYSTEM DATA TO BUILD
C          THE NETWORK FLOW MODEL
      CALL PSDNFM
C          USES NETWORK FLOW ROUTINE TO FIND OPTIMAL ECONOMIC
C          SCHEDULE OF GENERATION, AND
C          CORRESPONDING LINE FLOWS
      CALL ESMANF
C          TABULATES POWER SYSTEM DATA AND RESULTS
      CALL TOPSDR
C          INPUTS PLOTTING DATA, PRINTS POWER SYSTEM DATA AND
C          RESULTS AS A SEMIGEOGRAPHIC DIAGRAM
      CALL DIGRAM
      STOP
      END


      SUBROUTINE TIMER(I,R,IC,RA)
C     RETURNS TIME AND INTERVAL VIA ARGUMENTS TO THE MAIN
C     PROGRAM FOR OUTPUT THERE.
      LOGICAL FORK
      FORK=.FALSE.
      CALL INT(I)
      IF(M.EQ.1234) GO TO 1
      IB=I
      IA=I
      M=1234
    1 J=I-IA
      IA=I
    4 I=J/3600
      R=FLOAT(J-3600*I)/60.0
      IF(FORK) RETURN
      FORK=.TRUE.
      RA=R
      IC=I
      J=IA-IB
      GO TO 4
      END
```

```
      SUBROUTINE PSDNFM
C     INPUT AND ASSEMBLY OF POWER SYSTEM DATA TO BUILD THE
C     NETWORK FLOW MODEL FOR ECONOMIC SCHEDULING STUDY
      COMMON /BKLNLS/ TLENGT,LTYPE,DEVOLT,NRCCTS,NRCONS
      COMMON /BKF   / FLLOSS
      COMMON /BKBMOD/ AVINCO,STTBUS,ENDBUS,NSPB,LIMPWR,IB,A
      COMMON /BKCLB / CSCALE,LINK,BUSNR
      COMMON /BKNETW/ L(1200),U(1200),C(1200),S(1200),T(1200)
      COMMON /BKGMDL/ GMIN,SOURCE,NSPG,GMAX,ECR,
     1SCINCO,COSTIN
      COMMON /BKDMDL/ DEMAND,SINK
      DIMENSION TITLE(10),SNAME(2),TNAME(2)
      DIMENSION AT1(200),AT2(200),ASN(200),AEN(200),ACT(200),
     1LLC(200),ABN(200)
      INTEGER XFMRS,GENS,DEMS,BUSSES,GENBUS,SEREAC,XFMSER,
     1STTBUS,ENDBUS,FLLOSS,DEVOLT,XFMRNR,HV,SOURCE,SINK,TL,
     2TU,BUSNR,SETNR,GMAX,GMIN,ECR,TDEMAN,DEMAND,ARCS,U,C,S,T
      INTEGER AEN,ABN
C     ALPHABETIC TITLE,SNAME,TNAME,STANAM,AT1,AT2,ASN
C                    TIMING THE COMPUTATION
      CALL TIMER(MINT,SECT,MINI,SECI)
C                    INPUT SOME OPERATIONAL DATA
      READ(5,907) CSCALE,AVINCO,NSPB,NSPG
  907 FORMAT(F10.0,F10.4/2I5)
C                    PRESERVE WHOLE NUMBERS IN OUTPUT
      CSCALE=CSCALE+.001
C                    INPUT POWER SYSTEM PARAMETERS
      READ(5,902) TITLE
  902 FORMAT(10A6)
      READ(5,901) LINES,XFMRS,GENS,DEMS,BUSSES,GENBUS,SEREAC
  901 FORMAT(7I5)
C                    STORE POWER SYSTEM INFO ON SCRATCH TAPE
      WRITE(3) TITLE
      WRITE(3) LINES,XFMRS,GENS,DEMS,BUSSES,GENBUS,SEREAC
      WRITE(3) CSCALE,NSPB,NSPG,AVINCO
C                    INCLUDE SERIES REACTORS WITH TRANSFORMERS
      XFMSER=XFMRS+SEREAC
C                    INPUT THE LINE DATA AND CONSTRUCT THE
C                    NETWORK FLOW EQUIVALENT MODEL
      DO 500 IB=1,LINES
      READ(5,903) LINENR,STTBUS,ENDBUS,TLENGT,LIMPWR,FLLOSS,
     1SNAME,TNAME,NRCCTS,DEVOLT,NRCONS,LTYPE
  903 FORMAT(I3,2I4,19X,F5.1,I5,I6,1X,2A5,1X,2A5,I2,I3,2I1)
C                    COMPUTES FULL-LOAD LINE LOSS
      CALL LINLOS
C                    CONSTRUCTS NETWORK FLOW MODEL OF LINE
      CALL BMODEL
C                    STORE LINE DATA ON SCRATCH TAPE
```

```
      WRITE(3) LINENR,TLENGT,LIMPWR,SNAME,TNAME,DEVOLT,A
  500 CONTINUE
C                     CHECK FOR NO TRANSFORMERS OR REACTORS
      IF(XFMSER.EQ.0) GO TO 39
C                     INPUT TRANSFORMER AND REACTOR DATA AND
C                     CONSTRUCT THE NETWORK FLOW MODEL
      DO 501 IX=1,XFMSER
      READ(5,904) XFMRNR,STTBUS,ENDBUS,LIMPWR,FLLOSS,SNAME,
     1TNAME,NRCCTS,HV,LV
  904 FORMAT(I3,2I4,24X,I5,I6,1X,2A5,1X,2A5,I2,2I3)
      IB=LINES+IX
      CALL BMODEL
C                     STORE TRANSFORMER AND SERIES REACTOR
C                     DATA ON SCRATCH TAPE
      WRITE(3) XFMRNR,LIMPWR,SNAME,TNAME,HV,LV,A
  501 CONTINUE
C                     CREATION OF SOURCE AND SINK NODES
   39 SOURCE=BUSSES+1
      SINK=BUSSES+2
C                     INITIALIZATION OF GENERATOR TOTALS
      TL=0
      TU=0
C                     LINK NOW HAS CORRECT VALUE FOR GENERATORS
C                     LINK USUALLY COMPUTED IN MODELLING
C                     SUBROUTINES
      LINK=LINK+1
C                     INPUT THE GENERATOR DATA AND CONSTRUCT
C                     THE NETWORK FLOW EQUIVALENT MODEL
      DO 502 IG=1,GENS
      READ(5,905) TNAME,BUSNR,STANAM,SETNR,GMAX,GMIN,COSTIN,
     1ECR,SCINCO,MAXSPA
  905 FORMAT(2A5,I4,A2,I3,2I4,F5.2,5X,I3,F5.2,I3)
      TU=TU+GMAX
      TL=TL+GMIN
C                     CONSTRUCTS NETWORK FLOW MODEL OF GENERATOR
      CALL GMODEL
C                     STORE GENERATOR DATA ON SCRATCH TAPE
      WRITE(3) TNAME,STANAM,SETNR,COSTIN,SCINCO,GMAX
C                     STORE GENERATOR DATA FOR SORTING
      AT1(IG)=TNAME(1)
      AT2(IG)=TNAME(2)
      ASN(IG)=STANAM
      AEN(IG)=SETNR
      ACT(IG)=COSTIN
      ABN(IG)=BUSNR
  502 CONTINUE
C                     INPUT THE DEMAND DATA
      TDEMAN=0
      DO 503 ID=1,DEMS
      READ(5,906) BUSNR,SNAME,DEMAND
  906 FORMAT(I3,1X,2A5,I5)
C                     CONSTRUCTS NETWORK FLOW MODEL OF DEMAND
      CALL DMODEL
      TDEMAN=TDEMAN+DEMAND
C                     STORE DEMAND BUS NAMES ON SCRATCH TAPE
      WRITE(3) SNAME
```

```
  503 CONTINUE
C                       WRITE DATA SUMMARY INFO ON SCRATCH TAPE
      WRITE(3) TL,TU,TOEMAN
C                       CREATE THE RETURN ARC
      LINK=LINK+1
      L(LINK)=0
      U(LINK)=55555
      S(LINK)=SINK
      T(LINK)=SOURCE
      C(LINK)=0
C                       SET NETWORK FLOW MODEL SIZE
      ARCS=LINK
      NODES=BUSSES+2
C                       ALARM AND HALT FOR TOO LARGE A NETWORK
      IF(ARCS.GT.1200.OR.NODES.GT.400) GO TO 41
      GO TO 43
   41 WRITE(6,908) ARCS,NODES
  908 FORMAT(19H NETWORK TOO LARGE.,I4,6H ARCS,I4,7H NODES.,
     16H HALT.)
      STOP
C                       SORT GENERATOR DATA IN TEMPORARY LOCATIONS
C                       INTO AN ORDER-OF-MERIT TABLE
   43 CALL SORTEM(GENS,ACT,LLC)
C                       WRITE ORDER-OF-MERIT TABLE ON SCRATCH
C                       TAPE FOR FINAL OUTPUT
C                       J LOCATES GENERATORS IN UNSORTED ARRAYS
C                       COST ARRAY HAS BEEN SORTED INTO MERIT ORDER
      DO 517 IG=1,GENS
      J=LLC(IG)
C                       ARRAYS WRITTEN ON SCRATCH TAPE
C                       IN MERIT ORDER
  517 WRITE(3) AT1(J),AT2(J),ASN(J),AEN(J),ACT(IG),J,ABN(J)
C                       WRITE NETWORK FLOW MODEL DATA
C                       ON LINKING TAPE
      WRITE(4) ARCS,NODES,(L(K),U(K),C(K),S(K),T(K),K=1,ARCS)
      REWIND 4
      RETURN
      END
```

```
      SUBROUTINE BMODEL
C     CONSTRUCTS MULTIPLE ARC, PIECEWISE LINEAR NETWORK FLOW
C     MODEL OF POWER SYSTEM BRANCH (LINES, TRANSFORMERS, AND
C     SERIES REACTORS)
      COMMON /BKBMOD/ AVINCO,STTBUS,ENDBUS,NSPB,LIMPWR,IB,A
      COMMON /BKCLB / CSCALE,LINK,BUSNR
      COMMON /BKNETW/ L(1200),U(1200),C(1200),S(1200),T(1200)
      COMMON /BKF   / FLLOSS
      INTEGER SEG,FLLOSS,STTBUS,ENDBUS,S,T,U,C
C                     DETERMINE SEGMENT SIZE
      SEG=LIMPWR/NSPB
C                     COMPUTE COST CURVE COEFFICIENT A
      FLCOST=AVINCO*FLOAT(FLLOSS)/1000.
      A=FLCOST/(FLOAT(LIMPWR)**2)
      OLDCST=0.
C         EACH DO ITERATION MODELS ONE COST CURVE SEGMENT
C         BY CONSTRUCTING TWO OPPOSING ARCS
      DO 700 IS=1,NSPB
C                     COMPUT ARC INDEX FROM BRANCH
C                     AND SEGMENT INDICIES
      LINK=2*(NSPB*(IB-1)+IS)-1
      L(LINK)=0
      L(LINK+1)=0
      S(LINK)=STTBUS
      T(LINK)=ENDBUS
      S(LINK+1)=ENDBUS
      T(LINK+1)=STTBUS
C                     TEST FOR LAST SEGMENT
      IF(IS.EQ.NSPB) GO TO 1
C                     ALL SEGMENTS BUT THE LAST ONE
      U(LINK)=SEG
      U(LINK+1)=SEG
      PF=SEG*IS
      DELTA=SEG
      GO TO 2
C                     LAST SEGMENT
    1 U(LINK)=LIMPWR-(NSPB-1)*SEG
      U(LINK+1)=U(LINK)
      PF=LIMPWR
      DELTA=U(LINK)
C                     QUADRATIC COST CURVE FOR ALL SEGMENTS
    2 COSTPF=A*PF**2
      SINCST=(COSTPF-OLDCST)/DELTA
      C(LINK)=SINCST*CSCALE
      C(LINK+1)=C(LINK)
      OLDCST=COSTPF
  700 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GMODEL
C     CONSTRUCTS NETWORK FLOW MODEL OF GENERATORS WITH
C     LINEAR COST CURVES, AND TWO-SLOPE COST CURVES
      COMMON /BKNETW/ L(1200),U(1200),C(1200),S(1200),T(1200)
      COMMON /BKCLB / CSCALE,LINK,BUSNR
      COMMON /BKGMDL/ GMIN,SOURCE,NSPG,GMAX,ECR,
     1SCINCO,COSTIN
      INTEGER U,C,S,T,GMIN,SOURCE,BUSNR,GMAX,ECR
      LINK=LINK+1
C                     MODELLING SINGLE COST SETS, AND LOW COST
C                     REGION OF TWO COST SETS
      L(LINK)=GMIN
      C(LINK)=COSTIN*CSCALE
      S(LINK)=SOURCE
      T(LINK)=BUSNR
C                     CHECK FOR TWO-COST SETS
      IF(ECR.GT.0) GO TO 2
      U(LINK)=GMAX
      RETURN
    2 U(LINK)=ECR
C                     SECOND ARC MODELS HIGH-COST REGION
      LINK=LINK+1
      L(LINK)=0
      U(LINK)=GMAX-ECR
      C(LINK)=SCINCO*CSCALE
      S(LINK)=SOURCE
      T(LINK)=BUSNR
      RETURN
      END


      SUBROUTINE LINLOS
C     COMPUTES FULL-LOAD TRANSMISSION LINE LOSS FROM LINE
C     INDICATIVE DATA FOR NON-MIXED,STANDARD BRITISH LINES.
C     NORMAL WEATHER RATING, 3PHASE LOSS
      COMMON /BKLNLS/ TLENGT,LTYPE,DEVOLT,NRCCTS,NRCONS
      COMMON /BKF   / FLLOSS
      INTEGER DEVOLT,FLLOSS
C                     TEST FOR NON-STANDARD LINE
      IF(LTYPE.EQ.0) RETURN
      IF(LTYPE.EQ.1) LTYPE=175
C         DETERMINE CONDUCTOR SIZE
      IF(LTYPE.EQ.4) GO TO 4
C                     .175 SQIN CONDUCTORS
      FLLPMI=174.
      GO TO 43
C                     .4  SQIN CONDUCTORS
    4 FLLPMI=160.
C     SAME LOSS EXPRESSION  GOOD FOR BUNDLED CONDUCTORS AND
C     MULTIPLE CIRCUITS
   43 FLLOSS=FLLPMI*TLENGT*FLOAT(NRCCTS*NRCONS)
      RETURN
      END
```

```
      SUBROUTINE DMODEL
C     CONSTRUCTS NETWORK FLOW MODEL OF A DEMAND
      COMMON /BKNETW/ L(1200),U(1200),C(1200),S(1200),T(1200)
      COMMON /BKCLB / CSCALE,LINK,BUSNR
      COMMON /BKDMDL/ DEMAND,SINK
      INTEGER U,C,S,T,SINK,DEMAND,BUSNR
      LINK=LINK+1
      C(LINK)=0
      IF(DEMAND.LT.0) GO TO 1
C                     EXPORT
      L(LINK)=DEMAND
      U(LINK)=DEMAND
      S(LINK)=BUSNR
      T(LINK)=SINK
      RETURN
C                     IMPORT
    1 L(LINK)=-DEMAND
      U(LINK)=-DEMAND
      S(LINK)=SINK
      T(LINK)=BUSNR
      RETURN
      END


      SUBROUTINE SORTEM(N,KEY,LOCATE)
C     MINIMUM-IN-PASS SORT BY KEY, SMALLEST IN KEY(1), AND
C     LARGEST IN KEY(N).  ORIGINAL POSITION RECORDED IN LOCATE.
      DIMENSION KEY(N),LOCATE(N)
C                     INITIALIZE THE LOCATION ARRAY
      DO 516 I=1,N
  516 LOCATE(I)=I
C                     NUMBER OF SORTING PASSES
      NM1=N-1
C                     SORT
      DO 515 J=1,NM1
      JP1=J+1
      DO 515 I=JP1,N
      IF(KEY(I).GE.KEY(J)) GO TO 515
      KCELL=KEY(J)
      KEY(J)=KEY(I)
      KEY(I)=KCELL
      LCELL=LOCATE(J)
      LOCATE(J)=LOCATE(I)
      LOCATE(I)=LCELL
  515 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ESMANF
C     CALLS NETWORK FLOW SUBROUTINES TO FIND THE
C     OPTIMAL GENERATION SCHEDULE AND LINE POWER FLOWS
C     MULTIPLE ARCS NETWORK FLOW MODEL
      COMMON /BKKIL/ L(1200),U(1200),C(1200),X(1200),
     1CBAR(1200),S(1200),T(1200),PI(400),KLABEL(400),
     2ELABEL(400),KS(2400),NODLOC(401),DIF,FECIRC,ARCS,NODES,
     3STAIND,STADES,K,NEXT,OFL,TFL,INFINY,TBT,TNBT,LDFN,STAGES
      INTEGER U,C,S,T,ARCS,X,PI,FECIRC
      INTEGER TBT,TNBT
C                    READ DATA FROM LINKING TAPE
      READ(4) ARCS,NODES,(L(K),U(K),C(K),S(K),T(K),K=1,ARCS)
      REWIND 4
C                    INITIALIZATION OF THE NETWORK FLOW MODEL
      DO 504 LINK=1,ARCS
  504 X(LINK)=0
      CALL NODARC
      DO 505 I=1,NODES
  505 PI(I)=0
      CALL CPIPI
      CALL TIMER(MINT,SECT,MINI,SECI)
      WRITE(6,924) MINI,SECI
  924 FORMAT(43H TIME TO INPUT AND ASSEMBLE PROBLEM DATA IS,
     1I3,9H MINUTES,F7.2, 9H SECONDS.)

C          SUBROUTINE MULTAR ATTEMPTS TO FIND AN OPTIMAL,
C          FEASIBLE FLOW FOR THE ASSEMBLED NETWORK
      CALL MULTAR
      CALL TIMER(MINT,SECT,MINI,SECI)
C                    TEST FEASIBLE CIRCULATION INDICATOR
      IF(FECIRC.EQ.1) GO TO 61
      WRITE(6,916)
  916 FORMAT(39H INFEASIBILITY AROSE DURING SCHEDULING.
     128H DIAGNOSTIC OUTPUT AND HALT.)
      GO TO 62
   61 CONTINUE
      WRITE(6,921) MINI,SECI
  921 FORMAT(38H TIME TO FIND THE ECONOMIC SCHEDULE IS,I3,
     1 9H MINUTES,F7.2, 9H SECONDS.)
C                    ADD FLOWS TO NETWORK DATA ON LINKING TAPE
   62 WRITE(4) ARCS,NODES,(L(K),U(K),C(K),S(K),T(K),X(K),
     1K=1,ARCS)
      REWIND 4
C                    WRITE ALGORITHM PERFORMANCE FIGURES
C                    ON SCRATCH TAPE
      WRITE(3) MINI,SECI,TBT,TNBT
      REWIND 3
      RETURN
      END
```

```
      SUBROUTINE TOPSOR
C     TABULAR OUTPUT OF POWER SYSTEM DATA AND RESULTS
      COMMON /BKLX/ L(1200),U(1200),C(1200),S(1200),T(1200),
     1X(1200)
      COMMON /BKTBG/ CSCALE,LINK,BUSNR
      COMMON /BKCNFL/ STTBUS,ENDBUS,IB,NSPB,BRICST,PF(300),
     1IC(300),A,BRCOST
      COMMON /BKGCON/ GMAX,GMIN,ECR,G(200),ICG(200),
     1OPDISP(200),GL(200),GU(200),IG,NSPG
     2,FCOST,SCINCO,COSTIN
      COMMON /BKSUMY/ TITLE,LINES,GENS,XFMRS,GENBUS,SEREAC,
     1BUSSES,DEMS,TU,TDEMAN,TL,RLCOST,XRCOST,GNCOST,
     2NODES,AVINCO,ARCS
      DIMENSION TITLE(10),SNAME(2),TNAME(2),HEAD(30),UP(300),
     1IMPEXP(200),OPINDR(11),OPINCS(200),GENPWR(200)
      DIMENSION ALPHAB(6)
      INTEGER ARCS,UP,DEVOLT,PF,STTBUS,ENDBUS,XFMRS,GENBUS,
     1SEREAC,XFMSER,XFMRNR,HV,SETNR,GMAX,G,ECR,DEMS,DEMFLO,
     2SINK,BUSNR,GMIN,GENS,BUSSES,U,C,S,T,X,TDEMAN,TU,TL
      INTEGER TBT,TNBT
C     ALPHABETIC TITLE,ALPHAB,HEAD,SNAME,TNAME,STANAM,ALFD,
C     ASTRSK,BLANK,OPINDR
      DATA (ALPHAB(I),I=1,6)/1HL,1HO,1HR,1HE,1HH,1HU/
      DATA ASTRSK,BLANK/4H *   ,4H    /
C                   READ NETWORK FLOW QUANTITIES FROM
C                   LINKING TAPE
      READ(4) ARCS,NODES,(L(K),U(K),C(K),S(K),T(K),
     1X(K),K=1,ARCS)
      REWIND 4
C                   READ POWER SYSTEM PARAMETERS FROM
C                   SCRATCH TAPE
      READ(3) TITLE
      READ(3) LINES,XFMRS,GENS,DEMS,BUSSES,GENBUS,SEREAC
      READ(3) CSCALE,NSPB,NSPG,AVINCO
C                   OUTPUT THE LINE DATA AND POWER FLOWS
      READ(5,902) HEAD
  902 FORMAT(10A6)
      WRITE(6,916) LINES
  916 FORMAT(1H1,25HLINE DATA AND POWER FLOWS/I4,6H LINES)
      WRITE(6,902) HEAD
      RLCOST=0.
      DO 506 IB=1,LINES
C                   READING LINE DATA FROM SCRATCH TAPE
      READ(3) LINENR,TLENTH,UP(IB),SNAME,TNAME,DEVOLT,A
C                   CONSTRUCT LINE POWER FLOW FROM ARC FLOWS
C                   IN THE NETWORK FLOW MODEL
      CALL CONFLO
C                   SUM THE LINE COSTS
```

```
      RLCOST=RLCOST+BRCOST
      LENGTH=TLENTH
C                     ARRAYS UP,PF,IC STORE DATA FOR PLOTTING
C                     OUTPUT
      WRITE(6,908) LINENR,STTBUS,ENDBUS,DEVOLT,LENGTH,
     1BRICST,UP(IB),PF(IB),SNAME,TNAME
  908 FORMAT(5I4,F7.4,2I5,2X,2A5,1X,2A5)
  506 CONTINUE
C                     OUTPUT TRANSFORMER AND SERIES REACTOR
C                     DATA AND POWER FLOWS
      READ(5,902) HEAD
      XFMSER=XFMRS+SEREAC
      XRCOST=0.
C                     TEST FOR NO TRANSFORMERS OR SERIES REACTORS
      IF(XFMSER.EQ.0) GO TO 73
      WRITE(6,917) XFMRS,SEREAC
  917 FORMAT(///1X,30HTRANSFORMER AND SERIES REACTOR/
     11X,20HDATA AND POWER FLOWS/I4,13H TRANSFORMERS,I4,
     216H SERIES REACTORS)
      WRITE(6,902) HEAD
      DO 507 IX=1,XFMSER
      IB=LINES+IX
C                     READING TRANSFORMER AND SERIES REACTOR
C                     DATA FROM SCRATCH TAPE
      READ(3) XFMRNR,UP(IB),SNAME,TNAME,HV,LV,A
C                     CONSTRUCT TRANSFORMER AND SERIES REACTOR
C                     POWER FLOWS FROM ARC FLOWS IN THE
C                     NETWORK FLOW MODEL
      CALL CONFLO
C                     SUM TRANSFORMER AND SERIES REACTOR COSTS
      XRCOST=XRCOST+BRCOST
C                     OUTPUT
      WRITE(6,908) XFMRNR,STTBUS,ENDBUS,HV,LV,BRICST,UP(IB),
     1PF(IB),SNAME,TNAME
  507 CONTINUE
   73 GNCOST=0.
C                     WRITE POWER SYSTEM PARAMETERS ON
C                     LINKING TAPE
      WRITE(4) LINES,XFMRS,GENS,DEMS,BUSSES,GENBUS,SEREAC
C                     COMPUTE BRANCH INDEXING PARAMETER
      NRBRAN=LINES+XFMRS+SEREAC
C                     WRITE BRANCH DATA ON LINKING TAPE
      WRITE(4) (IC(ILXS),UP(ILXS),PF(ILXS),ILXS=1,NRBRAN)
C                     OUTPUT THE GENERATOR DATA AND POWER
C                     SCHEDULE BY BUS NUMBERS
      READ(5,902) HEAD
      WRITE(6,918) GENS
  918 FORMAT(1H1,   31HGENERATOR DATA AND POWER OUTPUT/
     11X,I4,11H GENERATORS)
      WRITE(6,902) HEAD
      LINK=LINK+1
      ITGP=0
      DO 508 IG=1,GENS
C                     READING GENERATOR DATA FROM SCRATCH TAPE
      READ(3) TNAME,STANAM,SETNR,COSTIN,SCINCO,GMAX
C                     CONSTRUCT GENERATOR POWER FLOW AND FIND
```

```
C                       ITS DISPOSITION FROM THE NETWORK FLOW MODEL
      CALL CONGEN
C                       FLOAT AND STORE FOR INCREMENTAL COST PLOT
      GENPWR(IG)=G(IG)
C                       SUMMING THE GENERATED POWERS
      ITGP=ITGP+G(IG)
C                       SUMMING THE GENERATOR COSTS
      GNCOST=GNCOST+FCOST
C                       ONE, OR TWO-COST GENERATORS
      IF(ECR.EQ.GMAX) GO TO 44
C                       OUTPUT FOR TWO-COST GENERATORS
      WRITE(6,911) BUSNR,STANAM,SETNR,G(IG),GMIN,ECR,GMAX,
     1COSTIN,SCINCO,TNAME
  911 FORMAT(I4,A2,I3,4I4,2F5.2,8X,2A5)
      GO TO 508
C                       OUTPUT FOR SINGLE-COST GENERATORS
   44 WRITE(6,930) BUSNR,STANAM,SETNR,G(IG),GMIN,GMAX,COSTIN,
     1TNAME
  930 FORMAT(I4,A2,I3,2I4,4X,I4,F5.2,13X,2A5)
  508 CONTINUE
C                       PRINT TOTAL GENERATED POWER
      WRITE(6,932) ITGP
  932 FORMAT(/7X,I6,25H MW TOTAL GENERATED POWER)
C                       WRITE GENERATOR DATA ON LINKING TAPE
      WRITE(4) (ICG(IG),GU(IG),G(IG),GL(IG),IG=1,GENS)
      IALARM=0
C                       OUTPUT THE DEMAND DATA
      READ(5,902) HEAD
      WRITE(6,919) DEMS
  919 FORMAT(1H1,    21HDEMAND DATA AND CHECK/
     11X,I4, 8H DEMANDS)
      WRITE(6,902) HEAD
      SINK=BUSSES+2
      DO 509 ID=1,DEMS
      LINK=LINK+1
C                       READ DEMAND BUS NAMES FROM SCRATCH TAPE
      READ(3) SNAME
C                       TEST FOR THIS DEMAND SATISFIED
      IF(X(LINK).EQ.U(LINK)) GO TO 42
      IALARM=1
      ALFD=ASTRSK
      GO TO 43
   42 ALFD=BLANK
C                       TEST FOR IMPORT OR EXPORT
   43 IF(S(LINK).EQ.SINK) GO TO 40
C                       EXPORTS ARE POSITIVE DEMANDS
      BUSNR=S(LINK)
      DEMFLO=X(LINK)
      GO TO 41
C                       IMPORTS ARE NEGATIVE DEMANDS
   40 BUSNR=T(LINK)
      DEMFLO=-X(LINK)
   41 WRITE(6,912) BUSNR,DEMFLO,ALFD,SNAME
  912 FORMAT(I4,I5,A4,2A5)
      IMPEXP(ID)=DEMFLO
  509 CONTINUE
```

```
C                     TEST FOR A FAILED DEMAND, PRINT ALARM
      IF(IALARM.EQ.1) WRITE(6,920)
  920 FORMAT(9X,28H * THIS DEMAND NOT SATISFIED)
C                     READ SUMMARY DATA FROM SCRATCH TAPE
      READ(3) TL,TU,TDEMAN
C                     PRINT TOTAL NET DEMAND
      WRITE(6,921) TDEMAN
  921 FORMAT(/I9,20H MW TOTAL NET DEMAND)
C                     WRITE DEMAND DATA ON LINKING TAPE
      WRITE(4) (IMPEXP(ID),ID=1,DEMS)
      REWIND 4
C                     OUTPUT THE MERIT-ORDER TABLE INDICATING
C                     GENERATOR DISPOSITIONS AND INCREMENTAL
C                     COSTS AT PRESENT GENERATION
      READ(5,902) HEAD
      WRITE(6,922) GENS
  922 FORMAT(1H1,    17HMERIT-ORDER TABLE/
     11X,I4,11H GENERATORS)
      WRITE(6,902) HEAD
C                     SET SPACES TO BLANK IN DISPOSITIONS CHART
      DO 521 IA=1,5
  521 OPINDR(2*IA)=BLANK
      DO 518 IG=1,GENS
C                     READ MERIT-ORDER TABLE FROM SCRATCH TAPE
      READ(3) TNAME,STANAM,SETNR,COSTIN,J,BUSNR
C                     J REFERENCES UNSORTED TABLE OF GENERATORS
C                     TEST GENERATOR OPERATING DISPOSITION
      DO 519 IA=1,6
      OPINDR(2*IA-1)=BLANK
      IF(OPDISP(J ).EQ.ALPHAB(IA)) OPINDR(2*IA-1)=ALPHAB(IA)
  519 CONTINUE
      OPINCS(J)=FLOAT(ICG(J))/CSCALE
      WRITE(6,914) BUSNR,STANAM,SETNR,G(J),COSTIN,OPINCS(J),
     1OPINDR,TNAME
  914 FORMAT(I4,A2,I3,I4,2F5.2,2X,11A1,11X,2A5)
  518 CONTINUE
      READ(5,931) SCALMW
  931 FORMAT(F5.0)
C                     PRINT THE GENERATOR DISPOSITION-
C                     INCREMENTAL COST-POWER OUTPUT DIAGRAM
      CALL COSPLT(GENPWR,OPINCS,OPDISP,SCALMW,GENS)
C                     PRINT DATA AND RESULT SUMMARY PAGE.
      CALL SUMMRY
      RETURN
      END
```

```
      SUBROUTINE CONFLO
C     CONSTRUCTS BRANCH POWER FLOWS FROM THE ARC FLOWS IN
C     THE NETWORK FLOW MODEL OF THE BRANCH
      COMMON /BKLX   / L(1200),U(1200),C(1200),S(1200),
     1T(1200),X(1200)
      COMMON /BKTBG/ CSCALE,LINK,BUSNR
      COMMON /BKCNFL/ STTBUS,ENDBUS,IB,NSPB,BRICST,PF(300),
     1IC(300),A,BRCOST
      INTEGER SEG,U,C,S,T,X,STTBUS,ENDBUS,PF
C                     PRECOMPUTE SEGMENT SIZE
      LINK=2*NSPB*(IB-1)+1
      LSAVE=LINK
      ISAVE=1
      STTBUS=S(LINK)
      ENDBUS=T(LINK)
      SEG=U(LINK)
      DO 800 IS=1,NSPB
C                     COMPUTE ARC INDEX FROM BRANCH INDEX
      LINK=2*(NSPB*(IB-1)+IS)-1
C                     DETECT POWER FLOW LEVEL
      IF(X(LINK).GT.0.OR.X(LINK+1).GT.0) GO TO 1
      GO TO 800
    1 LSAVE=LINK
      ISAVE=IS
  800 CONTINUE
C                     COMPUTE THE PIECEWISE LINEAR
C                     INCREMENTAL COST
      IC(IB)=C(LSAVE)
      BRICST=FLOAT(IC(IB))/CSCALE
C                     DETERMINE POWER FLOW DIRECTION
      IF(X(LSAVE+1).GT.0) GO TO 2
C                     POWER FLOW IS POSITIVE (FORWARD DIRECTION)
      PF(IB)=X(LSAVE)+(ISAVE-1)*SEG
      GO TO 3
C                     POWER FLOW IS NEGATIVE (REVERSE DIRECTION)
    2 PF(IB)=-(X(LSAVE+1)+(ISAVE-1)*SEG)
C                     COMPUTE THE POWER FLOW COST,
C                     QUADRATIC COST CURVE
    3 BRCOST=A*FLOAT(PF(IB))**2
      RETURN
      END
```

```
      SUBROUTINE CONGEN
C     CONSTRUCTS GENERATOR POWER AND FINDS ITS DISPOSITION
C     FROM THE NETWORK FLOW MODEL
C     ONE AND TWO-COST GENERATORS
      COMMON /BKTBG/ CSCALE,LINK,BUSNR
      COMMON /BKLX  / L(1200),U(1200),C(1200),S(1200),
     1T(1200),X(1200)
      COMMON /BKGCON/ GMAX,GMIN,ECR,G(200),ICG(200),
     1OPDISP(200),GL(200),GU(200),IG,NSPG
     2,FCOST,SCINCO,COSTIN
      DIMENSION ALPHAB(6)
      INTEGER U,C,S,T,X,BUSNR,GMIN,GMAX,ECR,G,GU,GL
C                     ALPHABETIC ALPHAB,OPDISP
      DATA (ALPHAB(I),I=1,6)/1HL,1HD,1HR,1HE,1HH,1HU/
      LINK=LINK+1
      GMIN=L(LINK)
      BUSNR=T(LINK)
      GU(IG)=GMAX
      GL(IG)=GMIN
C                     TEST FOR ONE OR TWO-COST GENERATOR
      IF(U(LINK).EQ.GMAX) GO TO 1
C                     TWO-COST GENERATOR
      ECR=U(LINK)
      G(IG)=X(LINK)+X(LINK+1)
C                     COST OF GENERATED POWER IN L/HR
      FCOST=FLOAT(C(LINK)*X(LINK)+C(LINK+1)*X(LINK+1))/CSCALE
C                     DETERMINE WHICH COST IS OPERATIVE
      IF(G(IG).GT.ECR) GO TO 3
C                     GENERATOR OPERATING IN FIRST COST RANGE
      ICG(IG)=C(LINK)
C                     DETERMINE GENERATOR DISPOSITION
      OPDISP(IG)=ALPHAB(2)
      IF(G(IG).EQ.GMIN) OPDISP(IG)=ALPHAB(1)
      IF(G(IG).EQ.ECR) OPDISP(IG)=ALPHAB(4)
      LINK=LINK+1
      RETURN
C                     GENERATOR OPERATING IN SECOND COST RANGE
    3 LINK=LINK+1
      ICG(IG)=C(LINK)
C                     DETERMINE GENERATOR OPERATING DISPOSITION
      OPDISP(IG)=ALPHAB(5)
      IF(G(IG).EQ.GMAX) OPDISP(IG)=ALPHAB(6)
      RETURN
C                     ONE-COST GENERATOR
    1 G(IG)=X(LINK)
      ICG(IG)=C(LINK)
C                     DETERMINE GENERATOR DISPOSITION
      OPDISP(IG)=ALPHAB(3)
      IF(G(IG).EQ.GMIN) OPDISP(IG)=ALPHAB(1)
      IF(G(IG).EQ.GMAX) OPDISP(IG)=ALPHAB(6)
      ECR=GMAX
C                     COST OF GENERATED POWER IN L/HR
      FCOST=FLOAT(ICG(IG)*G(IG))/CSCALE
      RETURN
      END
```

```
      SUBROUTINE COSPLT(GEN,COST,ALF,SCALMW,N)
C GENERATOR INCREMENTAL COST PLOTTING SUBROUTINE
      DIMENSION AL(121),S(13),GEN(N),COST(N),ALF(N)
      DATA BLANK/6H      /,BLH/6HI     /,ALH/6H+     /
C O/P FIRST LINE ON NEW PAGE
      WRITE(6,1)
    1 FORMAT(7H1$/MWHR )
C MW SCALE TAKEN AS SCALMW PER INCH FOR 12 INCHES
C COST SCALE FIXED AT 1.0 $/MWHR PER 10 LINES
      XMAX=12.0*SCALMW
      DO11 I=1,54
C CLEAR TO BLANK LINE
      DO 3 J=1,121
    3 AL(J)= BLANK
      IFLAG=0
      L=55-I
C DECIDE I OR + START FOR LINE
      AL(1)=BLH
      IF(L.NE.(L/10)*10) GO TO 4
      IFLAG=1
      AL(1)=ALH
    4 CONTINUE
      TOP= FLOAT(L)/10.0 +0.05
      BOT= TOP - 0.10
      DO 2 J=1,N
C TEST EACH GEN INSIDE COST QUANTUM
      IF(COST(J).GT.TOP.OR.COST(J).LE.BOT )GO TO 2
C QUANTISE MW INTO CELL ALONG LINE
      K=IFIX((GEN(J)/XMAX)*120. +0.5 ) +1
      IF(K.LT.1.OR.K.GT.121) GOTO 2
C WRITE ALPHANUMERIC REQ. INTO CELL IN LINE
      AL(K)= ALF(J)
    2 CONTINUE
      IF(IFLAG.GT.0) GO TO 6
C O/P LINE BEGINING  'I'  ETC.
      WRITE(6,5) AL
    5 FORMAT(5X,121A1)
      GO TO 11
    6 W= FLOAT(L)/10.
C O/P LINE BEGINING   3.0 +  ETC.
      WRITE(6,7) W,AL
    7 FORMAT(F4.1,1X,121A1)
   11 CONTINUE
C O/P X AXIS LINE INC. M W
      WRITE(6,8)
    8 FORMAT(5H 0.0 ,12(10H+---------- ),5H+ MW  )
      DO 9 I=1,13
    9 S(I)=ABS(FLOAT(I-1)*SCALMW)
C O/P XAXIS SCALING  MARKS
      WRITE(6,10)  S
   10 FORMAT ( 2X,12(F5.0,5X),F5.0)
      RETURN
      END
```

```
      SUBROUTINE SUMMRY
C     PRINTS DATA AND RESULTS SUMMARY PAGE
      COMMON /BKTBG/ CSCALE,LINK,BUSNR
      COMMON /BKCNFL/ STTBUS,ENDBUS,IB,NSPB,BRICST,PF(300),
     1IC(300),A,BRCOST
      COMMON /BKGCON/ GMAX,GMIN,ECR,G(200),ICG(200),
     1OPDISP(200),GL(200),GU(200),IG,NSPG
     2,FCOST,SCINCO,COSTIN
      COMMON /BKSUMY/ TITLE,LINES,GENS,XFMRS,GENBUS,SEREAC,
     1BUSSES,DEMS,TU,TDEMAN,TL,RLCOST,XRCOST,GNCOST,
     2NODES,AVINCO,ARCS
      DIMENSION TITLE(10)
      INTEGER GENS,XFMRS,GENBUS,SEREAC,BUSSES,DEMS,TU,
     1TDEMAN,TL,ARCS,TBT,TNBT
C     ALPHABETIC ALPHAB,OPDISP,TITLE
      READ(3) MINF,SECF,TBT,TNBT
      REWIND 3
      NRBRAN=LINES+XFMRS+SEREAC
      WRITE(6,923) TITLE,LINES,GENS,XFMRS,GENBUS,SEREAC,
     1BUSSES,NRBRAN,DEMS
  923 FORMAT(1H1,   35HECONOMIC SCHEDULING BY NETWORK FLOW//
     11X,24HDATA AND RESULTS SUMMARY//1X,10A6//
     25X,5HLINES,10X,I4,2X,10HGENERATORS,6X,I4/
     35X,12HTRANSFORMERS,3X,I4,2X,16HGENERATOR BUSSES,I4/
     45X,15HSERIES REACTORS,I4,2X, 6HBUSSES,10X,I4/
     55X,14HTOTAL BRANCHES,1X,I4,2X, 7HDEMANDS,9X,I4)
C                    POWER SUMMARY
      WRITE(6,924) TU,TDEMAN,TL
  924 FORMAT(//1X,13HPOWER SUMMARY//
     15X,24HTOTAL MAXIMUM GENERATION,I6,3H MW/
     25X,20HTOTAL DEMANDED POWER,4X,I6,3H MW/
     35X,24HTOTAL MINIMUM GENERATION,I6,3H MW)
C                    COMPUTE SYSTEM COSTS
      TOCOST=RLCOST+XRCOST+GNCOST
      TOTDEM=TDEMAN
      AVCGEN=GNCOST/TOTDEM
      AVCSUP=TOCOST/TOTDEM
C                    COST SUMMARY
      WRITE(6,926) RLCOST,XRCOST,GNCOST,TOCOST,AVCGEN,AVCSUP
  926 FORMAT(//1X,12HCOST SUMMARY//
     15X, 9HLINE COST,26X,F11.3,5H L/HR/
     25X,35HTRANSFORMER AND SERIES REACTOR COST,F11.3,5H L/HR/
     35X,15HGENERATION COST,20X,F11.3,5H L/HR/
     45X,10HTOTAL COST,25X,F11.3,5H L/HR/
     55X,31HAVERAGE COST OF GENERATED POWER,4X,F11.5,7H L/MWHR/
     65X,30HAVERAGE COST OF SUPPLIED POWER ,5X,F11.5,7H L/MWHR)
      WRITE(6,928) CSCALE,AVINCO
  928 FORMAT(5X,10HCSCALE    ,E10.3/
```

```
     15X,10HAVINCO     ,E10.3)
C                        PROGRAM TIMING
      CALL TIMER(MINT,SECT,MINI,SECI)
      WRITE(6,927) MINF,SECF,MINT,SECT
  927 FORMAT(//1X,14HPROGRAM TIMING/
     128X,3HMIN,3X,3HSEC/
     25X,22HNETWORK FLOW ALGORITHM,I4,F7.2/
     35X,18HTOTAL RUNNING TIME,4X,I4,F7.2)
C                        NETWORK FLOW MODEL SUMMARY
      WRITE(6,933) ARCS,NODES,TBT,TNBT,NSPB,NSPG
  933 FORMAT(//1X,26HNETWORK FLOW MODEL SUMMARY//
     15X, 4HARCS,I5,5X, 5HNODES,I5/
     25X,13HBREAKTHROUGHS,I5,5X,16HNONBREAKTHROUGHS,I5/
     35X,19HSEGMENTS PER BRANCH,I3,5X,
     422HSEGMENTS PER GENERATOR,I3)
      RETURN
      END
```

```
      SUBROUTINE DIGRAM
C     INPUTS PLOTTING DATA, ASSEMBLES AND PLOTS POWER SYSTEM
C     DATA AND RESULTS FOR SEMI-GEOGRAPHICAL DIAGRAM
      COMMON /BKPARS/ LINES,XFMRS,GENS,DEMS,BUSSES,SEREAC
      COMMON /BKPLT/ IHB(300),IVB(300),ISB(300),IHG(200),
     1IVG(200),ISG(200),IHD(200),IVD(200),ISD(200),NRSHTS
      DIMENSION IAA(100),IBB(100),ICC(100),IDD(100),IHOR(100),
     1IVER(100),ICOD(100)
      DIMENSION IC(300),UP(300),PF(300),ICG(200),GU(200),
     1G(200),GL(200),IMPEXP(200)
      INTEGER XFMRS,GENS,DEMS,SEREAC,UP,PF,GU,GL,G
C                     IF THERE IS NO DIAGRAM, RETURN
      READ(5,930) IDIGSW
  930 FORMAT(I5)
      IF(IDIGSW.EQ.0) RETURN
C                     SCALES COST DATA FOR THE DIAGRAM
      READ(5,930) IPLTSC
C                     READ DATA AND RESULTS FROM THE LINKING TAPE
      READ(4) LINES,XFMRS,GENS,DEMS,BUSSES,GENBUS,SEREAC
      NRBRAN=LINES+XFMRS+SEREAC
      READ(4) (IC(ILXS),UP(ILXS),PF(ILXS),ILXS=1,NRBRAN)
      READ(4) (ICG(IG),GU(IG),G(IG),GL(IG),IG=1,GENS)
      READ(4) (IMPEXP(ID),ID=1,DEMS)
C                     INPUT THE PLOTTING DATA
      CALL PLDAIN
      DO 301 ISH=1,NRSHTS
      N=0
C                     SELECTS LINE, TRANSFORMER, AND SERIES
C                     REACTOR DATA FOR SHEET IS
      DO 302 ILXS=1,NRBRAN
      IF(ISB(ILXS).NE.ISH) GO TO 302
      N=N+1
      IBB(N)=IC(ILXS)/IPLTSC
      ICC(N)=UP(ILXS)
      IDD(N)=PF(ILXS)
      IHOR(N)=IHB(ILXS)
      IVER(N)=IVB(ILXS)
      ICOD(N)=3
  302 CONTINUE
C                     SELECTS GENERATOR DATA FOR SHEET ISH
      DO 304 IG=1,GENS
      IF(ISG(IG).NE.ISH) GO TO 304
      N=N+1
      IAA(N)=ICG(IG)/IPLTSC
      IBB(N)=GU(IG)
      ICC(N)=G(IG)
      IDD(N)=GL(IG)
      IHOR(N)=IHG(IG)
```

```
      IVER(N)=IVG(IG)
      ICOD(N)=4
  304 CONTINUE
C                     SELECTS DEMAND DATA FOR SHEET ISH
      DO 305 ID=1,DEMS
      IF(ISD(ID).NE.ISH) GO TO 305
      N=N+1
      IDD(N)=IMPEXP(ID)
      IHOR(N)=IHD(ID)
      IVER(N)=IVD(ID)
      ICOD(N)=1
  305 CONTINUE
C                     PLOTS DATA ON SHEET ISH
      CALL PLTOUI(IAA,IBB,ICC,IDD,IHOR,IVER,ICOD,N)
  301 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PLDAIN
C     INPUT PLOTTING COORDINATES FOR EACH SYSTEM COMPONENT
C     PLOTTING DATA IN SAME ORDER AS POWER SYSTEM DATA
      COMMON /BKPARS/ LINES,XFMRS,GENS,DEMS,BUSSES,SEREAC
      COMMON /BKPLT/ IHB(300),IVB(300),ISB(300),IHG(200),
     1IVG(200),ISG(200),IHD(200),IVD(200),ISD(200),NRSHTS
      INTEGER XFMRS,GENS,DEMS,BUSSES,SEREAC
C                     INPUT THE PLOTTING DATA
      READ(5,950) NRSHTS
  950 FORMAT(I5)
      NRBRAN=LINES+XFMRS+SEREAC
C                     LINES, TRANSFORMERS AND SERIES REACTORS
      READ(5,951) (IHB(ILXS),IVB(ILXS),ISB(ILXS),ILXS=1,NRBRAN)
C                     GENERATORS
      READ(5,951) (IHG(IG),IVG(IG),ISG(IG),IG=1,GENS)
C                     DEMANDS
      READ(5,951) (IHD(ID),IVD(ID),ISD(ID),ID=1,DEMS)
  951 FORMAT(6X,2I3,3X,I3)
      RETURN
      END
```

```
      SUBROUTINE PLTOUI(A,B,C,D,IHOR,IVERT,ICODE,N)
C     NORRIS ROUTINE FOR AUTOMATIC SPATIAL PRINTING OF DATA
C     TO GO WITH A LINE DIAGRAM OF THE PHYSICAL PLANT.
      DIMENSION A(N),B(N),C(N),D(N),IHOR(N),IVERT(N),ICODE(N)
      DIMENSION LINA(28),KHOR(26),IOP(26)
      INTEGER A,B,C,D
      DATA LINA(1) /6H(1H*, /,LINA(28)/6H  )    /
      DATA IFEVEX /6H  5X, /,IFIVE /6H  I5, /
      WRITE(6,2)
    2 FORMAT(1H1)
C     REPEAT EACH SET OF 4 LINES 14 TIMES FOR A PAGE
      DO 100 KLINE=1,14
      KEMPTY=0
C     GIVEN ONE SET OF 4 LINES, INSPECT COORD DATA FOR
C     OCCUPANCY OF EACH CELL OF 26 ACROSS THE PAGE.
      DO 21 K=1,26
      KHOR(K)=0
      DO 20 I=1,N
      IF(IHOR(I).EQ.K.AND.IVERT(I).EQ.KLINE) GO TO 10
      GO TO 20
C     STORE THE INDEX OF THE DATA VECTOR WHICH OCCUPIES CELL K
   10 KHOR(K)=I
      KEMPTY=1
   20 CONTINUE
   21 CONTINUE
C     IF A COMPLETE SET OF 4 LINES IS BLANK THEN WRITE AS
C     BLANK SET AND SAVE WORK
      IF(KEMPTY.EQ.0) GO TO 30
      DO 99 LINE=1,4
C     REPEAT EACH LINE OF 4 WITHIN A SET
      LINEG=5-LINE
      LL=0
      DO 50 K=1,26
C     FILL EACH CELL OF 26 ACROSS THE PAGE WITH RELEVANT
C     FORMAT DETAIL
C     FORM BASIC BLANK FORMAT(1H*,5X,5X,...,5X)
      LINA(K+1)=IFEVEX
      I=KHOR(K)
C     REDISCOVER IF CELL IS FULL OR EMPTY
      IF(I.EQ.0) GO TO 50
      IF(ICODE(I).LT.LINEG) GO TO 50
C     CELL TAKEN AS EMPTY DEPENDING ON ICODE AND OVERRIDING
C     COORD DATA.
C     FULL CELL REQUIRES I5, STORED IN FORMAT(1H*,5X,I5,...,5X)
      LINA(K+1)=IFIVE
      LL=LL+1
      GO TO (110,120,130,140),LINE
C     EACH LINE IN SET OF 4 HAS RELEVANT DATA TO BE OUTPUT.
```

```
C       FOR ICODE(I)=4, 4 LINES OF DATA OUTPUT IN A BLOCK.
C       FOR ICODE(I)=1, ONLY LAST LINE IN BLOCK IS OUTPUT.
C       REFERENCE POINT REGARDED AS LOWER RIGHT HAND OF BLOCK.
  110   IOP(LL)=A(I)
        GO TO 50
  120   IOP(LL)=B(I)
        GO TO 50
  130   IOP(LL)=C(I)
        GO TO 50
  140   IOP(LL)=D(I)
   50   CONTINUE
C       IF A LINE IS STILL EMPTY, DUE TO ICODE EFFECT, WRITE
C       A BLANK LINE.
        IF(LL.EQ.0) GO TO 61
C       WRITE A COMPLETE LINE OF CELLED DATA
        WRITE(6,LINA) (IOP(J),J=1,LL)
        GO TO 99
   61   WRITE(6,62)
   62   FORMAT(1H*)
C       1H* REQUIRED IN ALL FORMATS FOR EDGE CAPABILITY.
   99   CONTINUE
        GO TO 100
   30   WRITE(6,31)
   31   FORMAT(////)
  100   CONTINUE
C       RECYCLE TO FILL PAGE WITH 4*14 LINES
        RETURN
        END
```

Appendix 4 TEST POWER SYSTEMS

A4.1 GHKS5 Glimn, Habermann, Kirchmayer, and Stagg 5 bus system.

The smallest of the test systems, GHKS5 is a selection of five relatively isolated busses from the third American test system, KS28. GHKS5 was selected for several reasons. The small number of busses and generators makes it easy to do exploratory studies using CMLFP. However its geographical extent of about 300 miles suits it for economic scheduling studies that include transmission losses. GHKS5 has appeared as the test example in recent papers on economic scheduling. (15,16) Originally it was given in a paper on loss formulas, and takes its name and bus numbering from that work.(23)

A peculiarity of the line data is that the shunt susceptances are all listed as zero. This is done to conform to the test system of reference 15, so as to allow comparison of results. The actual values of line shunt susceptance may be extracted from the line data of test system KS28. Line thermal limits and number of parallel circuits per branch are assumed values.

Table III of reference 15 is the source of the generator data on costs and limits. Omitting the constant, no-load cost of generation, the quadratic functions approximating the generator cost characteristics are:

$f_1 = 0.005g_1^2 + 2.45g_1$ TWIN BRANCH

$f_2 = 0.005g_2^2 + 3.89g_2$ FORT WAYNE

$f_3 = 0.005g_3^2 + 3.51g_3$ PORTSMOUTH

f is in units of dollars per hour, and g in megawatts. From data in Table III the no-load generation cost for each generator can be calculated: 106.0, 40.6, and 44.4 dollars per hour, respectively; totalling 191.0 dollars per hour. Synchronous capacitors are installed at busses 4 and 5, so all busses are voltage-controlled.

The 266 MW demand pattern is that of Table II, in the paper by Dopazo, et al.(15) The reactive demands and the bus voltage magnitudes are included with the listings of optimal and check solutions in Chapter 3. They are from Tables II and V.

The base case system diagram of Figure 1, reference 23 is the source of the 186 MW demand data, as well as the reactive demands and nominal voltage profile. Check solutions for this second case are not given in the thesis.

A4.2 SQ19 Squires' 19 bus system

In his paper on economic scheduling,(53) Squires gives reduced data for the entire high-voltage system of a midwestern utility company. Squires' bus numbering is used without change, but geographical names are lacking.

The branch series admittances, included in the system

diagram of Figure 5, have been converted to series impedances in per cent on a 100 MVA base. An attempt has been made to re-distribute the line shunt susceptances, previously lumped at each bus. Six shunt susceptances left over are listed with the data for completeness of the complex model. Line lengths and thermal limits have been estimated, but can only be approximate values.

The author presents four linear incremental cost of generation curves, one per generator bus. Integrating, we have the equivalent cost of generation functions:

$f_{16} = 0.00167g_{16}^2 + 1.517g_{16}$

$f_{17} = 0.002833g_{17}^2 + 1.483g_{17}$

$f_{18} = 0.00062g_{18}^2 + 1.263g_{18}$

$f_{19} = 0.005g_{19}^2 + 1.20g_{19}$

Generation limits are not available from his paper, so values wide enough not to influence the economic dispatch were assumed.

Figure 6 of the paper lays out his 100% load pattern on the system diagram. Reactive demands are included, but no bus voltage magnitudes or line power flows. Consequently, a nominal voltage profile was assumed, with no attempt or check for optimality. Generator bus voltage magnitudes were all four set at 1.0 p.u. This data may be found with the optimal generation schedule in Chapter 3. The 10 real power demands total 310.7 MW. For the network flow model, each was rounded to the nearest mega-

watt, to total 311 MW.

Squires' 70% load point was chosen as the second test problem. The previous demands were scaled down in proportion, totalling 217 MW; and the same nominal voltage profile was applied.

A4.3 KS28 Kirchmayer and Stagg AEP 28 bus system

This early version of the American Electric Power system was taken from a paper on transmission losses.(34) The system's 132 kV transmission network extends over approximately 400 miles. The authors have introduced some physical simplifications to reduce the model size, yet have retained the essential characteristics of the system as it was in 1950. KS28 was the largest test system that could be conveniently checked in the CMLFP. It turned out, in addition, to be the sternest test of the scheduling method, and thus the most interesting system of the test series. Busses are identified by their original geographic names, and the generator bus numbering is also kept.

The line parameters were taken from Figure 2, the system impedance diagram; and converted to per cent on a 100 MVA base. Line lengths were estimated from a system map. The number of parallel circuits per line was deduced from the line parameters. Thermal limits were

approximated from other data. The three-winding transformer at Windsor is represented by an equivalent tee circuit with centre point bus number 3, the Windsor generating bus.

Generator operating costs were derived from Table 5.2 of Kirchmayer's book on economic system operation.(32)

$f_1 = 0.0082g_1^2 + 1.28g_1$ TWIN BRANCH

$f_2 = 0.0044g_2^2 + 0.795g_2$ TIDD

$f_3 = 0.0019g_3^2 + 1.809g_3$ WINDSOR

$f_4 = 0.00429g_4^2 + 0.657g_4$ PHILO

$f_5 = 0.00222g_5^2 + 0.889g_5$ SPORN

$f_6 = 0.012g_6^2 + 0.300g_6$ CABIN CREEK

$f_7 = 0.0208g_7^2 + 0.635g_7$ LOGAN

$f_8 = 0.0127g_8^2 + 0.572g_8$ GLEN LYN

Real power generation limits were not available for these plants, so arbitrary values were assigned, leaving enough generation range to schedule the 1139 MW and 1730 MW demand patterns. The quadratic cost curves and the power limits were originally derived from the primary data of the individual sets, more than one of which was present at each location. The Lorain and Burger plants of the Ohio Edison System are not part of this load dispatching exercise and thus have their generations fixed. They generate 18 MW and 111 MW respectively for the 1139 MW loading.

As with real power, the generator reactive power limits were not known. The assumption was made that the

generators could deliver reactive megavars at the high-voltage bus to the value of one-half the real power capacity. For example, at Twin Branch 200 MVArs can be delivered to bus 1.

Nine synchronous capacitors of unknown limits are connected to the system for voltage control. They are located at busses: 9, 10, 13, 14, 15, 17, 19, 22, and 25. Representative reactive generation capacities are 24, 40, and 50 MVArs. Transformers were operating on nominal taps.

The first set of demands, totalling 1139 MW, are for a typical summer Sunday daytime average load. They are taken from Figure 4 of the paper by Kirchmayer and Stagg,(34) and are referred to there as the D-1 loading. The values given there were rounded to the nearest megawatt so that the rounded sum, 1139 MW, came closest to the unrounded sum, 1138.5 MW. The optimal solution listing in Chapter 3 includes the D-1 real power loading with reactive demands as well. These demands are a result of netting local demand, transfers over interconnections, and the local generation not being scheduled. The voltage profile given in Figure 4 was checked for optimality and no improvement could be made in it. Consequently it was used in the scheduling studies, and may be found with the other bus results and data in the optimal solution of Chapter 3.

The second demand pattern, a much heavier total load of 1730 MW, is representative of a summer weekday afternoon. The geographical distribution of this loading, named B-1 in the paper, is slightly different from that of the D-1 load pattern. Table VII contains this information, and was used, along with judicious guessing of the division boundaries, to develop the actual B-1 loads from the D-1 loads.

A voltage profile similar to the one given in Table V, B-1 loading, was found for this heavier system loading. Some adjustment was required to satisfy reactive generation constraints and also to avoid a low voltage condition at some load busses. No attempt to optimize the voltage-controlled bus settings was made. An optimal real power schedule was found, but is omitted from the thesis.

The third listing of demand data incorporates the 41 MW system losses for the D-1 demands into those demands in proportion to their size.

A4.4 MB23 CEGB 275/132 kV 23 bus system

The configuration of this test system is radial in character, with a generating station located at the end of each of the four arms. Four transformers are situated at the central point, connecting the high and the low voltage parts of the system. The greatest geographical spread

of the system is 160 miles from extremity to extremity. Shen and Laughton (50) used this test system to confirm their method of economic scheduling.

The 25 transmission lines in the system are all CEGB standard, being one of two types:
275 kV, 2 conductor bundle, 0.4 sq. in. conductors
132 kV, 1 conductor per phase, 0.175 sq. in. conductors
Line 10 is a double circuit line reduced to a single circuit equivalent.

The 180 MVA transformer is a standard type, and its three phase full load loss was tabulated on a CEGB data sheet. The three remaining transformers, rated at 155 MVA, each have an estimated loss of 773 kW. A single 90 MVA series reactor separates bus 1 from bus 2.

24 single-cost generators supply power to the system. The first six, all in one B station, are evenly divided between busses 1 and 2. The next four sets are connected to bus 11, and have incremental costs quite close to sets 5 and 6 of bus 2. All six generators at bus 14 are fed from nuclear reactors, and so are capable of only minor load changes. The output of each generator is therefore fixed at its maximum, 83 MW. The remaining generators are in reality all located at one geographical point. They are large, economical to run, and are distributed between busses 20 and 23.

Estimates of the reactive generation limits at each voltage-controlled bus were made.

BUS NO.	LAGGING MAXIMUM REACTIVE MVAR	LEADING MINIMUM REACTIVE MVAR
1	117	-66
2	117	-66
11	148	-84
14	234	-192
20	409	-352
23	513	-578

The 2643 MW demand figures are the result of netting local demands, and transfers to other parts of the 132 kV and 275 kV systems. Negative quantities represent net imports. A power factor of 0.95 was assumed in estimating the reactive demands from real power demand data abstracted from a CEGB megawatt load flow study.

The permissable voltage level variation at 132 kV and 275 kV is $\pm$10%. Operating MB23, good practice would be to attempt to hold all bus voltage magnitudes between 1.00 p.u. and 1.05 p.u. This was achieved by setting all generator busses at 1.05 p.u. and all transformer taps at a ratio of 1.03. Slight voltage and reactive generation constraint violations may be noted. The reactive part of the transfers were not included in the original data, and somewhat inappropriate values may have been assumed. As a

result, there was some difficulty in establishing a feasible reactive power schedule and voltage profile.

The 2300 MW demand level is scaled directly from the 2643 MW pattern. The optimal solution to the latter loading is the one given in Chapter 3.

A4.5 BCA56 Automatic system control experiment test area 56 bus system

Approximately one-third of the Bristol Grid Control Area in the South-Western Region was the object of an extensive experiment in automatic system control.(18) Figure 5 of that paper shows the geographical spread of this test area: 180 miles. A version of this system was assembled for economic scheduling studies.

Of the 43 generators, 13 were older, two-cost sets.

The voltage profile and its corresponding reactive power schedule, as well as the reactive demands were developed in the same way as those of MB23.

A4.6 GR142 Grinstead Grid Control Area 142 bus system

Since, in CEGB practice the smallest area scheduled is a grid control area, it seemed appropriate to select such a system for experiment with the network flow method. The Grinstead Grid Control Area in the South-Eastern Region was made available for study in the following way. Two standard CEGB programs, NASAP, and NELS were run on

the same basic data set, yielding output listings. The required data, pertaining to a March weekday 9 a.m. loading, was drawn from these results.

Detailed data on each branch was not obtainable, and so the tests used the branch reactances and thermal limits listed by NASAP. Parallel circuits were lumped together. For simplicity, a few supergrid lines from the national grid were included in the final data set. NELS had ordered up 60 generators to supply the 9 a.m. load. 14 of these were two-cost machines.

Demand data was only assembled after a meticulous correlation of NASAP load, transfer, and generator data. Data for a complete complex variable system model could not be extracted from the file by these programs.

A4.7 SG149 Supergrid 149 bus system

To assemble the data by hand from printed records was an arduous task. It was undertaken with the goal of demonstrating the network flow method on an example of national scale. The test system transmission network comprised the Supergrid of England and Wales as it was in 1965 at the beginning of the conversion from 275 kV to 400 kV. Lower voltage transmission lines were omitted from the study, with no attempt being made to include their affect on the solution. For this reason generating stations remote from supergrid points were not included.

177 individual generating sets at 28 different localities were delivering power to the system at 59 generating busses. In effect then, this study was to schedule most of the important generators in the national system. The rather light total demand in the source data set was therefore scaled up to fall within the generation range. True generator costs were impossible to get, so semi-fictional ones based on set size and geographical position were developed.

Complete branch data was assembled from system diagrams and listings. Station diagrams were built up from tabulated connection data. The system switching was modified to accomodate the heavier demand level. Although sufficient data was available to complete the complex model, this time-consuming burden was avoided.

A4.8 Test power systems data

The following nine pages list data for the power systems: GHKS5, SQ19, KS28, and MB23. Unfortunately the data for the other three systems BCA56, GR142, and SG149 is too bulky to be included. As far as NFESP is concerned, the data sets of this appendix are complete for each system. However, only the branch data for the complex model appears here.

GHKSS GLIMN• HABERMANN• KIRCHMAYER• STAGG 5 BUS SYSTEM

LINS	XFMS	GENS	DEMS	BUSS	GBUS	SERE	SHRE	MVAB
5	0	3	4	5	5	0	0	100

LINE DATA

LIN NO•	STT BUS NO•	END BUS NO•	R PC	X PC	B PC	LINE LENG MI	THERM LIMIT MVA	3PHAS FLLOS KW	START BUS NAME	END BUS NAME	CDESCT TVOLOY S KVNP
1	1	2	3•0	10•3	0•0	75•	300	33780	TWIN BRANC	FT WAYNE	313210
2	2	5	8•0	26•2	0•0	47•	100	9860	FT WAYNE	MARION	113210
3	2	4	10•5	34•7	0•0	52•	100	13800	FT WAYNE	MUNCIE	113210
4	4	5	3•3	11•8	0•0	35•	100	3710	MUNCIE	MARION	113210
5	3	4	10•6	40•3	0•0	190•	140	31600	PORTSMOUTH	MUNCIE	213210

GENERATOR DATA

BUS NAME	BUS NO•	SSET A TNO•	GMX MW	GMN MW	INCR L/MR COST	CMR MW	ECR MW	2ND L/MR COST
TWIN BRANC	1	1	200	50	2•45	212		5•
FT WAYNE	2	1	100	10	3•89	106		5•
PORTSMOUTH	3	1	100	10	3•51	106		5•

DEMAND DATA

266MW TOTAL NET DEMAND

BUS NO•	BUS NAME	REAL DMND MW
1	TWIN BRANC	86
2	FT WAYNE	80
4	MUNCIE	70
5	MARION	30

DEMAND DATA

186MW TOTAL NET DEMAND

BUS NO•	BUS NAME	REAL DMND MW
1	TWIN BRANC	86
2	FT WAYNE	47
4	MUNCIE	36
5	MARION	17

SQ19 SQUIRES 19 BUS SYSTEM

LINS	XFMS	GENS	DEMS	BUSS	GBUS	SERE	SHRE	MVAB
24	0	4	10	19	4	0	6	100

LINE DATA

LIN NO.	STT BUS NO.	END BUS NO.	R PC	X PC	B PC	LINE LENG MI	THERM LIMIT MVA	3PHAS FLLOS KW	START BUS NAME	END BUS NAME	CDESCT TVOLOY S KVNP
1	2	1	1.50	4.20	6.4	42.	250	11200			0
2	2	3	1.80	5.20	.0	52.	250	13400			0
3	18	2	2.36	20.80	1.8	208.	275	21700			0
4	18	1	2.56	22.75	1.8	227.	275	24500			0
5	18	10	3.02	25.90	1.8	259.	275	31700			0
6	1	10	.67	3.21	.6	32.	200	3100			0
7	10	4	.30	.60	.0	6.	100	400			0
8	4	5	.36	1.47	6.76	14.	300	3490			0
9	6	5	1.32	3.68	.86	36.	250	9900			0
10	16	6	1.15	3.20	.86	32.	250	8500			0
11	6	13	1.0	3.0	.78	30.	250	6000			0
12	10	11	.26	.50	.0	5.	100	330			0
13	12	11	.19	.45	.0	4.	100	300			0
14	19	12	.30	.60	.0	6.	100	400			0
15	19	16	1.50	4.29	1.3	42.	250	11400			0
16	19	3	.80	2.30	.0	23.	250	6100			0
17	16	7	2.08	6.25	1.4	62.	250	16600			0
18	7	8	2.70	7.60	1.9	76.	250	20000			0
19	8	9	1.30	6.60	1.5	66.	200	6300			0
20	9	15	1.00	5.00	1.5	50.	200	4800			0
21	17	8	3.30	17.10	4.6	171.	200	16300			0
22	17	16	4.20	21.40	3.2	214.	200	20400			0
23	3	14	2.86	8.42	3.1	84.	250	22100			0
24	17	14	4.20	14.20	5.6	142.	250	34200			0

GENERATOR DATA

BUS NAME	BUS NO.	SSET TNO. A	GMX MW	GMN MW	INCR COST L/MR	CMR MW	ECR MW	2ND COST L/MR
	16	1	55	51	.517	60		1.67
	17	1	45	101	.483	50		2.833
	18	1	300	701	.263	330		.62
	19	1	60	151	.20	65		5.00

DEMAND DATA
311MW TOTAL NET DEMAND

BUS NO.	BUS NAME	REAL DMND MW
1		13
2		43
4		26
5		53
7		10
8		16
9		14
10		37
11		39
12		60

DEMAND DATA
217MW TOTAL NET DEMAND

BUS NO.	BUS NAME	REAL DMND MW
1		9
2		30
4		18
5		37
7		7
8		11
9		10
10		26
11		27
12		42

SHUNT REACTANCE DATA

BN	BSHUNT
4	7.37
10	17.01
11	11.55
12	11.65
15	6.6
16	7.3

KS28 KIRCHMAYER AND STAGG AEP 28 BUS SYSTEM

LINS	XFMS	GENS	DEMS	BUSS	GBUS	SERE	SHRE	MVAB
32	3	10	24	28	19	0	0	100

LINE DATA

LIN NO.	STT BUS NO.	END BUS NO.	R PC	X PC	B PC	LINE LENG MI	THERM LIMIT MVA	3PHAS FLLOS KW	START BUS NAME	END BUS NAME	CDESCT TVOLOY S KVNP.
1	1	9	3.0	10.3	20.	70.	270	23000	TWIN BRANC	FT WAYNE	313210
2	9	10	8.0	26.2	5.6	70.	90	6700	FT WAYNE	MARION	113210
3	9	22	10.5	34.7	7.2	80.	90	9400	FT WAYNE	MUNCIE	113210
4	22	10	3.3	11.8	2.4	35.	90	2300	MUNCIE	MARION	113210
5	19	9	4.2	14.0	12.8	56.	180	14000	LIMA	FT WAYNE	213210
6	13	19	3.3	10.4	8.4	54.	180	12000	FOSTORIA	LIMA	213210
7	14	13	3.0	10.0	8.4	45.	180	11000	HOWARD	FOSTORIA	213210
8	4	14	3.9	17.5	15.2	80.	180	14000	PHILO	HOWARD	213210
9	5	4	5.4	17.9	15.2	75.	180	19000	SPORN	PHILO	213210
10	5	16	3.9	13.3	11.2	60.	180	13000	SPORN	SOUTH POIN	213210
11	15	22	10.6	40.3	34.0	175.	180	35000	PORTSMOUTH	MUNCIE	213210
12	16	15	2.3	7.7	6.4	35.	180	7800	SOUTH POIN	PORTSMOUTH	213210
13	17	16	3.3	10.8	10.0	52.	180	12000	TURNER	SOUTH POIN	213210
14	17	6	1.2	3.8	6.8	20.	270	9300	TURNER	CABIN CREK	313210
15	17	7	3.3	9.2	7.6	42.	180	13000	TURNER	LOGAN	213210
16	7	18	3.3	11.2	9.2	45.	180	11000	LOGAN	SWITCHBACK	213210
17	18	8	1.9	6.4	5.6	33.	180	6500	SWITCHBACK	GLEN LYN	213210
18	4	25	5.9	16.7	20.8	75.	270	46000	PHILO	CANTON	313210
19	2	25	2.7	12.1	10.4	50.	180	8900	TIDD HV	CANTON	213210
20	3	25	4.7	12.8	10.4	60.	180	16000	WINDSOR HV	CANTON	213210
21	26	27	7.3	12.4	0.	15.	50	2400	TIDD LV	WINDSOR LV	2 6610
22	21	13	13.4	40.1	8.0	90.	90	11000	SANDUSKY	FOSTORIA	113210
23	11	21	4.2	12.8	2.4	30.	90	3700	LORAIN	SANDUSKY	113210
24	20	11	3.7	10.6	8.4	42.	180	13000	ASHLAND	LORAIN	213210
25	20	14	3.25	9.7	0.	25.	90	4000	ASHLAND	HOWARD	113210
26	23	20	3.3	9.3	4.0	22.	90	4600	MASSILLON	ASHLAND	113210
27	12	20	9.6	47.5	10.4	110.	90	7800	BURGER	ASHLAND	113210
28	12	23	8.4	34.0	7.2	79.	90	6850	BURGER	MASSILLON	113210

29	12	24	8.7	35.2	7.2	82.	90	7700	BURGER	ALLIANCE	113210
30	25	23	1.9	5.4	1.2	12.	90	2500	CANTON	MASSILLON	113210
31	24	25	2.9	8.2	2.0	19.	90	2600	ALLIANCE	CANTON	113210
32	5	17	3.3	11.0	9.6	50.	180	11500	SPORN	TURNER	213210

TRANSFORMER AND REACTOR DATA

XFM NO.	STT BUS NO.	END BUS NO.	R PC	X PC	TAP RGE PC	THERM LIMIT MVA	3PHAS FLLOS KW	START BUS NAME	END BUS NAME	C T S	HV KV	LV KV
33	28	3	.0	3.2	10	90	600	WINDSOR HV	WINDSOR CP	1	132	0
34	3	27	.0	2.9	0	45	300	WINDSOR CP	WINDSOR LV	1	0	66
35	2	26	.0	7.3	10	60	400	TIDD HV	TIDD LV	1	132	66

GENERATOR DATA

BUS NAME	BUS NO.	SSET TNO. A	GMX MW	GMN MW	INCR COST L/MR	CMR MW	ECR MW	2ND COST L/MR
TWIN BRANC	1	1	400	50	1.28	420		8.20
TIDD	2	1	240	60	.795	255		4.40
WINDSOR	3	1	160	40	1.809	150		1.90
PHILO	4	1	455	100	.657	480		4.29
SPORN	5	1	310	70	.889	330		2.22
CABIN CREK	6	1	220	116	.300	235		12.0
LOGAN	7	1	85	45	.635	90		20.8
GLEN LYN	8	1	136	80	.572	145		12.7
LORAIN	11	1	18	18	.70	60		15.
BURGER	12	1	111	111	2.0	200		2.0

DEMAND DATA
1139MW TOTAL NET DEMAND

BUS NO.	BUS NAME	REAL DMND MW
1	TWIN BRANC	86
2	TIDD HV	51
4	PHILO	81
5	SPORN	5
6	CABIN CREK	117
7	LOGAN	84
8	GLEN LYN	84
9	FT WAYNE	47
10	MARION	18
11	LORAIN	28
13	FOSTORIA	15
14	HOWARD	25
15	PORTSMOUTH	28
16	SOUTH POIN	54
17	TURNER	29
18	SWITCHBACK	55
19	LIMA	32
20	ASHLAND	23
21	SANDUSKY	17
22	MUNCIE	38
23	MASSILLON	17
24	ALLIANCE	30
25	CANTON	127
28	WINDSOR HV	48

DEMAND DATA
1730MW TOTAL NET DEMAND

BUS NO.	BUS NAME	REAL DMND MW
1	TWIN BRANC	157
2	TIDD HV	72
4	PHILO	115
5	SPORN	7
6	CABIN CREK	170
7	LOGAN	122
8	GLEN LYN	122
9	FT WAYNE	86
10	MARION	33
11	LORAIN	46
13	FOSTORIA	21
14	HOWARD	36
15	PORTSMOUTH	40
16	SOUTH POIN	77
17	TURNER	41
18	SWITCHBACK	80
19	LIMA	45
20	ASHLAND	38
21	SANDUSKY	28
22	MUNCIE	70
23	MASSILLON	28
24	ALLIANCE	50
25	CANTON	179
28	WINDSOR HV	67

DEMAND DATA
1180MW TOTAL NET DEMAND

BUS NO.	BUS NAME	REAL DMND MW
1	TWIN BRANC	89
2	TIDD HV	53
4	PHILO	83
5	SPORN	5
6	CABIN CREK	121
7	LOGAN	87
8	GLEN LYN	87
9	FT WAYNE	49
10	MARION	19
11	LORAIN	29
13	FOSTORIA	16
14	HOWARD	26
15	PORTSMOUTH	29
16	SOUTH POIN	56
17	TURNER	30
18	SWITCHBACK	57
19	LIMA	33
20	ASHLAND	24
21	SANDUSKY	18
22	MUNCIE	39
23	MASSILLON	18
24	ALLIANCE	31
25	CANTON	131
28	WINDSOR	50

MB23 CEGB 275/132KV 23 BUS SYSTEM

LINS	XFMS	GENS	DEMS	BUSS	GBUS	SERE	SHRE	MVAB
25	4	24	23	23	6	1	0	100

LINE DATA

LIN NO.	STT BUS NO.	END BUS NO.	R PC	X PC	B PC	LINE LENG MI	THERM LIMIT MVA	3PHAS FLLOS KW	START BUS NAME	END BUS NAME	CDESCT TVOLOY S KVNP
1	1	3	2.27	5.99	1.26	15.8	100				113211
2	1	4	2.72	7.15	1.51	18.9	100				113211
3	2	5	3.54	9.33	1.97	24.6	100				113211
4	8	5	2.83	7.45	1.58	19.7	100				113211
5	2	7	6.15	16.20	3.42	42.7	100				113211
6	3	6	5.76	15.16	3.20	40.0	100				113211
7	4	9	2.66	7.00	1.48	18.5	100				113211
8	9	7	2.01	5.29	1.11	13.9	100				113211
9	8	6	3.92	10.03	2.18	27.2	100				113211
10	11	10	2.07	5.45	4.66	29.1	200				213211
11	8	10	5.25	13.80	2.91	36.4	100				113211
12	9	10	5.25	13.80	2.91	36.4	100				113211
13	13	14	.39	3.56	24.30	53.9	620				127524
14	14	12	.39	3.56	24.30	53.9	620				127524
15	15	12	.35	3.12	21.24	47.2	620				127524
16	18	15	.32	2.92	19.84	44.8	620				127524
17	23	13	.79	7.14	48.71	108.7	620				127524
18	16	17	.10	.80	5.43	12.5	620				127524
19	17	18	.19	1.70	11.56	24.0	620				127524
20	19	18	.13	1.19	8.11	18.2	620				127524
21	20	19	.41	4.23	25.43	56.5	620				127524
22	22	18	.22	1.95	13.28	29.5	620				127524
23	20	21	.18	1.58	10.80	25.0	620				127524
24	21	22	.12	1.06	7.20	16.0	620				127524
25	23	16	.20	1.64	11.09	25.2	620				127524

TRANSFORMER AND REACTOR DATA

XFM NO.	STT BUS NO.	END BUS NO.	R PC	X PC	TAPRGE PC	THERM LIMIT MVA	3PHAS FLLOS KW	START BUS NAME	END BUS NAME	C T S	HV KV	LV KV
26	12	8	0.	10.64	15	155	773			1	275	132
27	13	8	0.	10.42	15	155	773			1	275	132
28	12	9	0.	8.36	15	180	864			1	275	132
29	13	9	0.	10.64	15	155	773			1	275	132
30	1	2	0.	20.00	0	90	200			1	132	132

GENERATOR DATA

BUS NAME	BUS NO.	SSET A	TNO.	GMX MW	GMN MW	INCR COST L/MR	CMR MW	ECR MW	2ND COST L/MR
	1	B	1	61	15	3.22	65		
	1	B	2	61	15	3.22	65		
	1	B	3	61	15	3.22	65		
	2	B	4	61	15	3.22	65		
	2	B	5	61	30	2.20	65		
	2	B	6	61	30	2.20	65		
	11		1	57	43	2.16	60		
	11		2	58	43	2.19	60		
	11		3	58	43	2.17	60		
	11		4	58	43	2.14	60		
	14		1	83	83	0.85	87		
	14		2	83	83	0.85	87		
	14		3	83	83	0.85	87		
	14		4	83	83	0.85	87		
	14		5	83	83	0.85	87		
	14		6	83	83	0.85	87		
	20	B	5	112	22	1.71	120		
	20	C	10	334	135	1.42	350		
	20	C	12	357	143	1.21	375		
	23	B	6	112	22	1.67	120		
	23	B	7	112	22	1.71	120		
	23	B	8	112	22	1.67	120		
	23	C	9	334	135	1.35	350		
	23	C	11	358	143	1.15	375		

DEMAND DATA
2643MW TOTAL NET DEMAND

BUS NO.	BUS NAME	REAL DMND MW
1		64
2		101
3		0
4		47
5		51
6		41
7		48
8		1
9		150
10		177
11		130
12		6
13		-4
14		480
15		201
16		132
17		344
18		104
19		376
20		-100
21		375
22		-210
23		129

DEMAND DATA
2300MW TOTAL NET DEMAND

BUS NO.	BUS NAME	REAL DMND MW
1		56
2		88
3		0
4		41
5		45
6		36
7		42
8		1
9		133
10		157
11		115
12		6
13		-4
14		425
15		177
16		117
17		304
18		92
19		333
20		-100
21		332
22		-210
23		114