T-2714

THE NODE CHAIN ALGORITHM: AN ALTERNATIVE SOLUTION METHOD FOR PURE NETWORK FLOW PROBLEMS

by

Mary D. Durham

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

T-2714

A thesis submitted to the Faculty and the Board of trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Mineral Economics).

Golden, Colorado

Date: July 25, 1983

Signed: Mary D Durham

Mary D. Durham

Approved: Robert E. Woolsey

Dr. R.E.~~D.~~ Woolsey

Thesis Advisor

Golden, Colorado

Date: Aug 15, 1983

Dr. Charles W. Berry, Head

Mineral Economics Department

ABSTRACT

In the field of network theory there exist numerous algorithms for the solution of all classes of network flow problems. This paper presents an additional algorithm, the Node Chain Algorithm, capable of solving pure minimum cost, transportation, transhipment and assignment problems that arise in many industries.

Included in this paper is a description of the algorithm, its theoretical basis and a comparison of the algorithm to another prominently used method, the Out-of-Kilter Algorithm.

The computer program used for the Node Chain Algorithm is not included in this paper but can be obtained from the Mineral Economics Department at the Colorado School of Mines. The program is stored in the Mineral Economics program library (MEL), under the name NCA.FOR.

TABLE OF CONTENTS

TABLE OF CONTENTS, Continued

LIST OF FIGURES

LIST OF FIGURES, Continued

LIST OF TABLES

LIST OF TABLES, Continued

LIST OF TABLES, Continued

ACKNOWLEDGEMENTS

I would like to acknowledge my advisor, Dr. Robert E. D. Woolsey, as well as my thesis committee members Dr. Charles E. Lienert and Dr. Ruth A. Maurer.

I want to thank my friends who put up with me for so many months while I was writing this thing. Their most important contribution was helping me to maintain my sense of humor.

Mostly I want to dedicate this thesis to my parents. I will never be able to thank them enough for all their help and support.

INTRODUCTION

Applications for network flow algorithms can be found in many aspects of modern society. Most industries face numerous distribution, transportation, communication and engineering problems which can be solved in the framework of network flow models.

Specifically, the mining and petroleum industries face distributional problems in minimizing the cost of transporting their products to their customers, refineries or mills. In many cases the formulation of network flow models for these problems reveals specific lower bounds on flows which further restrict the problem. For example, pipelines must maintain a certain minimum flow levels at all times whether for the transportation of natural gas, oil or water (in the case of water resources management). In the mining industry the railroad systems are used extensively by firms to transport the extracted ores to the firm's customers. In routing trains one also encounters a lower bound problem in that an engine will travel with at least some minimum number of cars resulting in a minimum level of ore to be transported over each route. Once again the common denominator in many distribution problems facing the extractive industries

is that the problems are lower bounded by some amount greater than zero.

Better algorithms for the solution of these problems are constantly being sought after. Currently in the field of network flow analysis there does exist an algorithm, the Out-of-Kilter Algorithm (OKA), which has been applied to many such problems. This algorithm, developed by Ford and Fulkerson in the early 1960's, can be used to solve pure capacitated network flow problems such as minimum cost, transportation and transshipment problems. Since the inception of this algorithm no other general method has been found as useful in the solution of these large distribution problems.

This paper presents an alternate method, the Node Chain Algorithm (NCA), for the solution of these problems. The NCA has one additional characteristics in that is appears to solve these lower bounded problems more efficiently than the OKA. This method was originally developed by Drs. W.M. Raike and R.E.D. Woolsey in the early 1960's. The NCA is a primal algorithm using the artificial arc method (Jensen and Barnes, 1980) which insures an initial feasible solution that is also basic.

This thesis presents the Node Chain Algorithm formally for the first time since it was first designed. Chapter 1 provides a short review of work done in the area of network

analysis. Chapter 2 provides an extensive explanation of the steps of the NCA as well as examples to clarify how the algorithm works. Chapter 3 shows, via examples that the theoretical basis for the NCA lies in the generalized upper bounding method of linear programming. In Chapter 4 the Node Chain and Out-of-Kilter algorithms are compared. Examined are their differences, similarities and the computational efficiency of each given different network structures. Chapter 5 provides a review of the paper.

CHAPTER 1

NETWORK ANALYSIS: A BACKGROUND

The original basis for network analysis lies in a branch of mathematics called graph theory that evolved with Leonhard Euler's formulation of the famous Konigsberg bridge problem in 1736 (Minieka, 1978). In the middle of the 1800's James Clerk Maxwell and Gustav Robert Kirchoff discovered some basic principles of network analysis applicable to the study of electrical circuits. Network theory became a convenient tool for looking at whole electrical systems. In the early 1900's, engineers developed network models for determining the best capacities of telephone trunk lines and switching centers so as to meet customer demands (Phillips and Garcia-Diaz, 1981).

During World War II great advances were made in the mathematical study of large scale systems. Specifically, Koopmans looked at reducing the overall shipping times as the shortage of cargo ships during the war constituted a critical bottleneck (Dantzig, 1963). In 1941, Frank L. Hitchcock, published his paper, "The Distribution of a Product from Several Sources to Numerous Localities," in which he "sketched out the partial theory of a technique

foreshadowing the simplex method." (Dantzig, 1963) In 1939, L.V. Kantorovich looked at the classical transportation problem as applied to production planning. Since the 1940's over one thousand papers have been written in the area of network analysis.

Network analysis can be divided into many different classes of problems including: pure or generalized problems, transportation, assignment, transshipment, maximum flow and minimum cost problems. In the 1950's considerable work was done in the area of transportation, maximum flow, assignment and pure minimum cost problems.

In the area of the transportation problem, Dantzig in 1951 looked at an application of the simplex method to these problems (Jensen & Barnes, 1980). Charnes and Cooper (1954) developed their "Stepping Stone Method" for the explanation of linear programming calculations. Ford and Fulkerson, in their book, Flows in Networks, present a solution method for the transportation problem that is a generalization of a procedure developed by Kuhn (1955) for the optimal assignment problem.

Solution methods for pure minimum cost problems appear in papers by Fulkerson (1961) and Orden (1956). In his 1961 paper, D.R. Fulkerson first presents what is known today as the Out-of-Kilter algorithm (or OKA). The OKA has become one of the most widely used algorithms for the solution of

pure minimum cost flow problems as well as maximal flow, transportation and transshipment problems.

The applications and problem types discussed above constitute a class of network problems called pure network flow problems. The term "pure" means that flow on an arc is conserved. Another class of problems exists, "the network-with-gains" or "generalized network flow" problem in which the flow on an arc may increase or decrease. The treatment of generalized network flow problems can be found in papers by Hultz, Klingman and Russell (1976), Charnes and Raike (1965), Balas (1966), and Glover, Hultz, Klingman and Stutz (1978), to name only a few.

It can be seen after only a cursory review of the work done in network analysis that the list of authors becomes extensive. For this reason the reader is referred to the books by Dantzig (1963), Jensen and Barnes (1980) and Ford and Fulkerson (1962) where the authors provide more complete descriptions of the work that has occurred over the years.

CHAPTER 2

THE NODE CHAIN ALGORITHM

2.1 INTRODUCTION

The Node Chain Algorithm (NCA) is a primal algorithm made up of two initialization steps and five principal steps in its main loop. The initialization steps establish a starting basic solution for the given network. The five principal steps involve for each iteration: pricing out all nodes and non-basic arcs and determining the entering basic variable (steps 3 and 4), determining the leaving basic variable and the amount of flow change (steps 5 and 6), followed by making the appropriate changes in the network (step 7).

This chapter first provides some brief definitions for the terminology used as well as describing the notation system. Next, each step in the algorithm is explained followed by three examples to clarify how the algorithm works.

2.2 DEFINITIONS, TERMINOLOGY AND NOTATIONS

Definitions and terminology

A network consists of a number of nodes or points connected by arcs. For the purposes of the Node Chain Algorithm all networks must have the following characteristics:

a) be a pure network - i.e., a network in which conservation of flow takes place on all arcs
b) a source and a sink with a return arc from sink to source
c) each arc is a directed arc
d) each arc has associated with it an upper and lower bound on flow and a cost (per unit of flow)

Arcs are said to originate at a node i and terminate at some node j.

Artificial Arcs connect supply nodes to demand nodes. All artificial arcs will have the following characteristics - an upper bound of $\bar{M}$ (where $\bar{M}$ is some very large number), a lower bound of zero and a cost per unit of flow equal to M (where M is also some very large number.

Supply Nodes are used only in initializing the network; supply nodes are those where the initial flow into the node

is greater than the initial flow out of the node. (Nodes with a positive net inflow.)

Demand Nodes are used only in initializing the network; demand nodes are those where the initial flow into the node is less than the initial outflow. (Nodes with a negative net inflow.)

Forward Arcs are those arcs which when followed from one node to another are followed in the direction of i to j, the true direction of flow on those arcs.

Backward Arcs are those arcs which when followed from one node to another are followed in the direction of j to i, the opposite direction of flow on those arcs.

A Spanning Tree is a tree that contains every node in the network for a network with "n" nodes a spanning tree will contain "n" nodes and "n-1" arcs.

Notation

The following notation system will be used throughout.

A_{ij} = an arc with flow from node i to node j

NI_n = net inflow at node n where for

$NI_{n=i} > 0$, node n is a supply node

$NI_{n=j} < 0$, node n is a demand node

AA_{ij} = artificial arc from supply node $+NI_n$ to demand node $-NI_n$

C_{ij} = cost per unit of flow on A_{ij}. For each AA_{ij}, $C_{ij} = M$.

F_{ij} = current level of flow on A_{ij}

LB_{ij} = lower bound on flow for A_{ij}. For each AA_{ij}, $LB_{ij} = 0$

P_{ij} = non-basic arc prices calculated in Step 4 (PRICE) where $P_{ij} = NP_j - NP_i - C_{ij}$.

NP_n = node price calculated in Step 3 - NEWPI where

$NP_{n=i}$ is the price of node i and

$NP_{n=j}$ is the price of node j

UB_{ij} = upper bound on flow for A_{ij}

2.3 - THE STEPS

The NCA consists of seven steps, two initialization steps with the remaining five making up the main loop of the algorithm. These steps are explained below and summarized in Section 2.1.

Step 1: Initialization

Purpose: To establish the initial flow on all arcs and create artificial arcs to connect supply nodes to demand nodes. The steps are as follows:

a) For each arc with $C_{ij} > 0$, set the initial flow (F_{ij}) equal to the arc's lower bound (LB_{ij}).

b) For each arc with $C_{ij} \leq 0$, set the initial flow equal to the arc's upper bound (UB_{ij}).

c) For each node determine the algebraic sum of inflows and outflows. Each node will be either a supply node (net inflow > 0) or a demand node (net inflow $\leq$ 0).

d) Locate a supply node i. From node i build an artificial arc (AA_{ij}) to a demand node j where the flow on AA_{ij} will be equal to:

   i) the supply at node i if that supply is less than or equal to the demand or

   ii) the demand at node at j if that demand is less than the supply.

Continue connecting the current supply node to demand nodes until all supply at that node is depleted. Move to the next supply node and continue connecting supply nodes to demand nodes in the above manner until all demands are satisfied. If all demands cannot be satisfied by the supplies the problem is infeasible.

It is should be noted that a node with a net inflow of zero is considered a demand node and will therefore have an artificial arc connected to it from some supply node. This insures that all nodes are connected in a spanning tree in the starting artificial solution. A spanning tree is maintained at all times by the algorithm. The arcs comprising the tree will be called "spanning arcs" and are equivalent in concept to basic variables in linear programming.

Step 2: PREGO

Purpose: To identify artificial arcs that parallel real arcs and eliminate those artificial arcs. The steps are as follows:

a) Locate artificial arcs that parallel real arcs.

b) Transfer as much flow as possible from the artificial to the real arc that it parallels.

c) If the flow on the artificial arc is reduced to zero the real arc replaces the artificial arc in the spanning tree and the artificial arc is removed from the network. The real arc now becomes a spanning arc to maintain the tree.

Step 3: NEWPI

Purpose: To assign a price (NP_n) to each node using the steps outlined below:

a) Select a starting node and assign it a price of zero. Follow a spanning arc from the starting node to any other node. Assign the node reached (the "terminal" node) the cost of the spanning arc followed. Do this for all nodes that can be reached from the starting node by a spanning arc.

b) For the nodes remaining without prices assigned in (a) above do the following: for nodes assigned prices above, continue to follow arcs in the tree (spanning arcs) and assign all further nodes reached a price ($NP_{n=j}$). $NP_{n=j}$ is calculated by:

Price of the node reached = price of the node at the start of the basic arc + cost of the basic arc or

$$NP_j = NP_i + C_{ij}$$

If the spanning arc followed is a backward arc then: $NP_j = NP_i - C_{ij}$. Continue this process until all nodes have been assigned a price.

Step 4: PRICE

Purpose: To calculate a price (P_{ij}) for each arc not in the tree and determine which arc will enter the tree from the set of eligible arcs determined below.

a) For each arc not in the tree calculate a price P_{ij}, where:

$$P_{ij} = NP_j - NP_i - C_{ij}$$

b) Arcs eligible for entering the tree are determined using the following guidelines:

i) if $P_{ij} > 0$ and $F_{ij} < UB_{ij}$ then A_{ij} should be considered to enter the tree where its flow will be increased.

ii) if $P_{ij} < 0$ and $F_{ij} > LB_{ij}$ then A_{ij} should be considered to enter the tree where its flow will be decreased.

iii) if $P_{ij} = 0$ then no improvement would result if A_{ij} were chosen to enter the tree.

iv) if all arcs at their upper bound and all arcs at their lower bound have $P_{ij} > 0$ or $P_{ij} < 0$, respectively, then stop.

If no flow remains on the artificial arcs the solution is optimal. If, however, flow

remains on these arcs the problem is infeasible.

c) From the eligible arcs determined in (b) above the arc with the highest absolute value for P_{ij} is chosen as the arc to enter the tree, KOMIN = A_{ij}.

Step 5: PATHFINDER

Purpose: To link node j to node i of the KOMIN arc by tracing a path through the current set of spanning arcs. This path will be referred to as "the circuit" and is determined as follows:

Starting at node j, of $KOMIN_{ij}$, trace a path through the tree of spanning arcs to node i, of $KOMIN_{ij}$.

Step 6: CHANGER

Purpose: To evaluate the arcs in the circuit determined above and select the arc that will leave the tree ($LEAVE_{ij}$) as well as the amount of flow change (ΔFLOW) to take place over the arcs in the circuit.

a) For all forward arcs calculate the maximum amount of flow change possible on these arcs,(ΔFA), where:

$$\Delta FA = \text{MIN } \{UB_{ij} - F_{ij}\}$$

b) For all backward arcs calculate the maximum amount of flow change possible on these arcs, ΔBA, where:

$$\Delta BA = \text{MIN } \{F_{ij} - LB_{ij}\}$$

c) For $KOMIN_{ij}$ calculate the maximum amount of flow change possible, ΔKA, where:

i) $\Delta KA = UB_{ij} - F_{ij}$ if $KOMIN_{ij}$ is a forward arc,

or

ii) $\Delta KA = F_{ij} - LB_{ij}$ if $KOMIN_{ij}$ is a backward arc,

d) The amount of flow change to take place over the circuit is determined by selecting the minimum of ΔFA, ΔBA and ΔKA, or

$$\Delta FLOW = \text{MIN } \{\Delta FA, \Delta BA \text{ and } \Delta KA\}$$

(ΔFLOW may be equal to zero.)

e) That arc which determines the amount of ΔFLOW is designated the arc to leave the tree, the LEAVE arc. If the KOMIN arc is also the LEAVE arc the set of spanning arcs will not change in the current iteration.

Step 7: NEWFLOWS

Purpose: To adjust the flows in the circuit upward or downward by the amount of ΔFLOW.

a) If ΔFLOW > 0 then:

   i) increase the flow on all forward arcs in the circuit by ΔFLOW and

   ii) decrease the flow on all backward arcs in the circuit by ΔFLOW.

b) If ΔFLOW = 0, the only change that will occur is the KOMIN arc will enter and the LEAVE arc will leave the basis.

c) If the KOMIN arc is the LEAVE arc, the set of spanning arcs will not change. This eliminates the need to recalculate node prices so go to Step 4, otherwise go to Step 3.

To summarize the steps just described and to provide a quick reference guide to the algorithm a summary flowchart is provided in Figure 2.1 on the following pages.

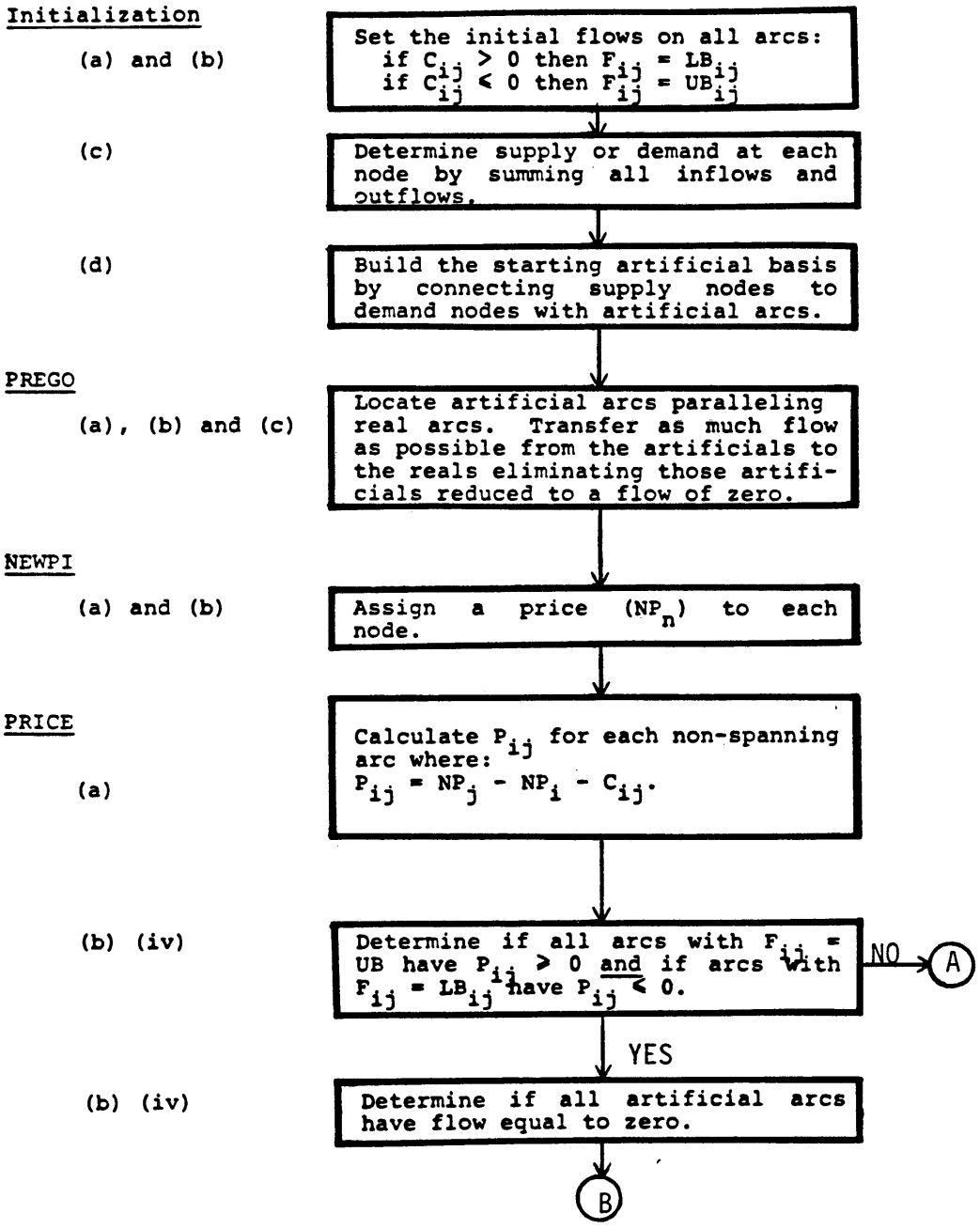

Figure 2.1: Summary flowchart of the Steps of the NCA.

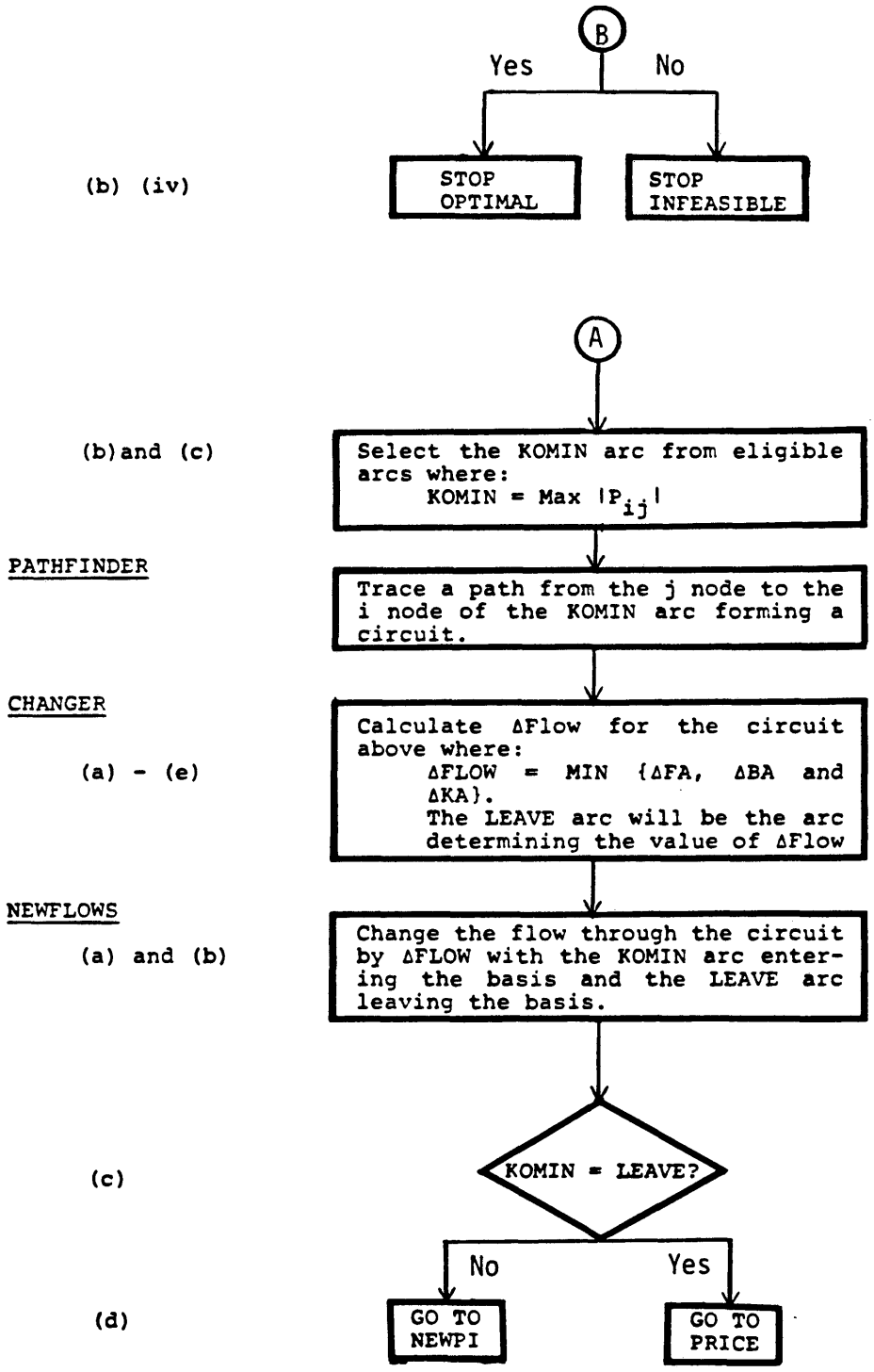

Figure 2.1, continued.

2.5 - EXAMPLES

The following examples are used to illustrate the Node Chain Algorithm.

1) A small network is solved using detailed explanations and figures at each step.
2) The solution of an infeasible network is attempted to illustrate the termination of the algorithm when no feasible solution exists. In addition this example contains an arc with a negative cost.
3) A larger network is solved using limited explanations and fewer figures.

Example 1

Consider the network shown below in Figure 2.2.

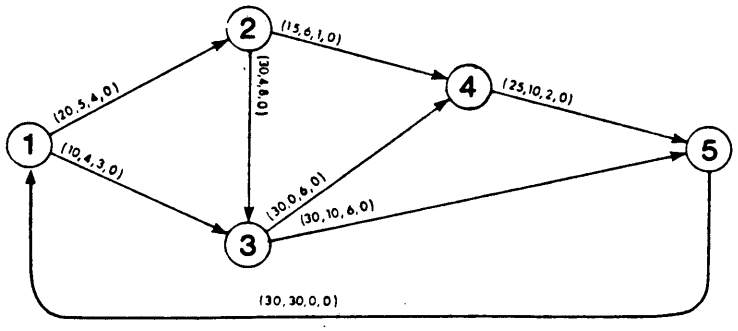

Figure 2.2: Example 1 - Initial network configuration

For all figures the parameters for each arc are listed as $(UB_{ij}, LB_{ij}, C_{ij}, F_{ij})$.

Initialization:

Initialization of the network requires establishing:

a) initial flows on all arcs (F_{ij}) and

b) calculating the net inflow at each node (NI_n).

The results of (a) and (b) above are shown in Figure 2.3.

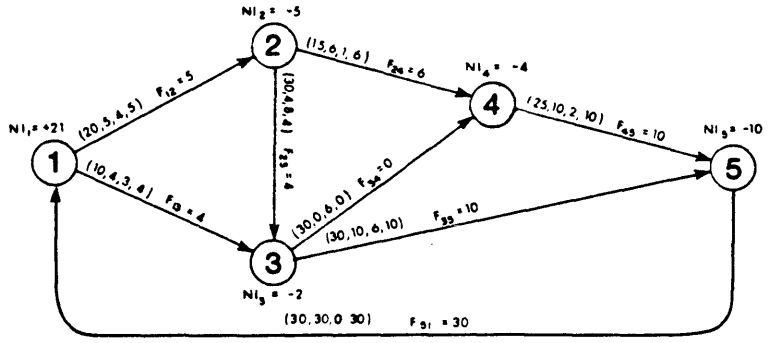

Figure 2.3: Example 1 - Initialization of arc flows with net inflows (NI) calculated at each node.

All arc costs are positive; therefore, the starting flows on all arcs are equal to their lower bounds. The net inflows at each node are calculated by summing the total inflows and then subtracting all outflows. For example, at node 2, the net inflow = 5 (inflow from A_{12}) - 6 (outflow from A_{24}) - 4 (outflow from A_{23}) = -5, indicating node 2 is a demand node.

The final step in initializing the network requires connecting the supply node(s) (node 1) to the demand node(s) (nodes 2, 3 and 4) as shown in Figure 2.4.

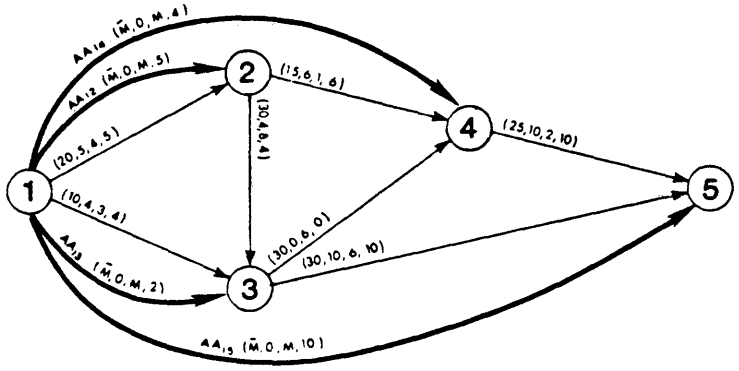

Figure 2.4: Example 1 - Artificial arcs added to balance supplies and demands.

To create the artificial arcs, node 1, which is the only supply node, is connected to all the other nodes. The artificial arcs created (AA_{12}, AA_{13}, AA_{14} and A_{15}) make up the starting set of spanning arcs.

PREGO

The network is now evaluated to determine if any of the artificial arcs parallel the real arcs. For those found, the maximum amount of flow possible is transferred to the real arc from its parallel artificial arc removing the artificial arc if possible. As shown in Figure 2.5 below AA_{12} and AA_{13} are replaced by A_{12} and A_{13} with the real arcs (A_{12} and A_{13}) now becoming part of the tree.

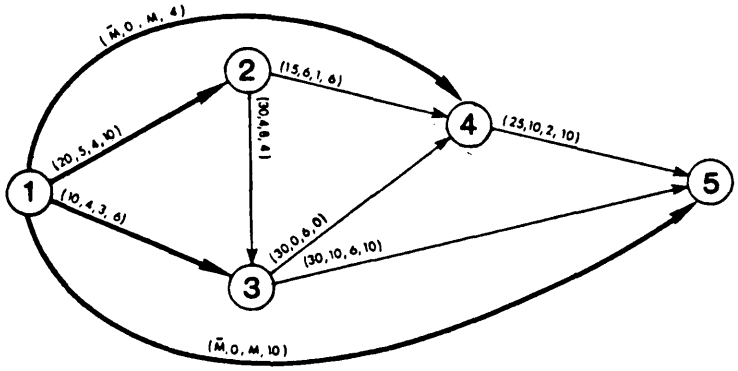

Figure 2.5: Example 1 - Results after PREGO.

NEWPI:

The first step in the main loop, NEWPI, requires calculating node prices (NP_n) for each node. Node 1 is arbitrarily selected as the starting node (see note below) and the spanning arcs from this node are followed to all other nodes with the following node prices assigned as shown in Figure 2.6.

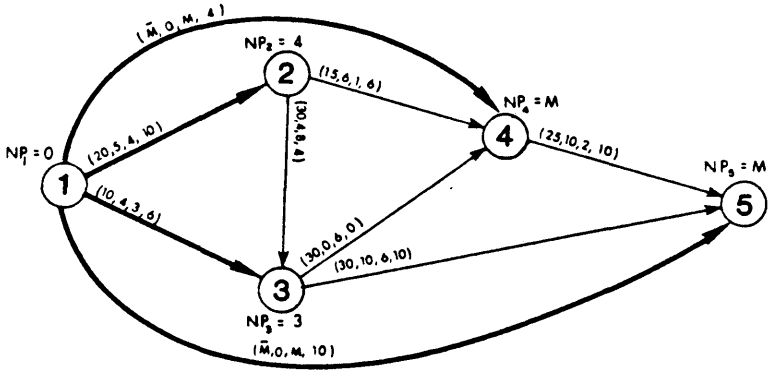

Figure 2.6: Example 1 - Iteration 1; node prices calculated.

* Note: Any node may be selected as the starting node. For simplicity and to provide consistency node 1 is selected throughout each example.

PRICE:

Prices (P_{ij}), for arcs not in the tree, are calculated using the node prices calculated above. For example $P_{24} = NP_4 - NP_2 - C_{24} = M-5$. The results are shown in Figure 2.7.

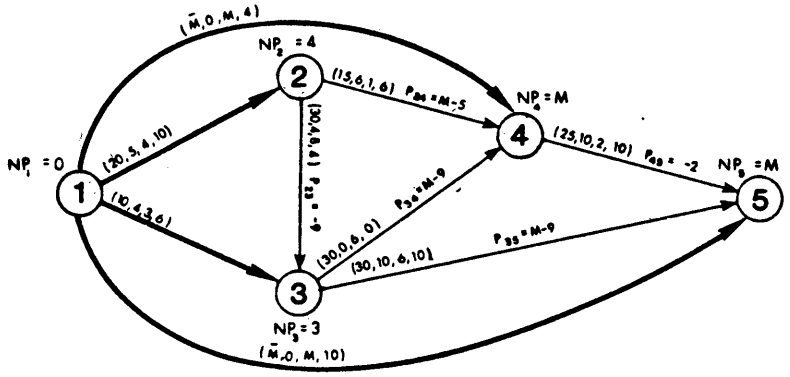

Figure 2.7: Example 1 - Iteration 1; arc price calculations.

A_{24}, with the maximum price, is selected as the KOMIN arc.

PATHFINDER :

A circuit from node 4 to node 2 of the KOMIN arc must be traced through the spanning arcs to determine which arcs are eligible to leave the arcs. For the current network this circuit will run from nodes 4 to 1 to 2. The arcs in the circuit are AA_{14} (a backward arc), A_{12} (a forward arc) and A_{24} (the KOMIN arc).

CHANGER:

Given the circuit established above, the LEAVE arc and the value of ΔFLOW must be determined. Evaluating the forward arcs, A_{12} only, ΔFA = 20 - 10 = 10. Evaluating the backward arcs, AA_{14} only, ΔBA = 4 - 0 = 4. Lastly looking at the KOMIN arc ΔKA is calculated ΔKA = 15 - 6 = 9 (where the KOMIN arc is a forward arc). Therefore, ΔFlow = MIN {10, 4, 9} = 4. Arc AA_{14} is designated the LEAVE arc as it determined the value of ΔFLOW.

NEWFLOWS

Given the results of PRICE and CHANGER, the network can now be changed with A_{14} leaving the tree and the flow change over the circuit being ΔFLOW = 4. The results are shown in Figure 2.8.

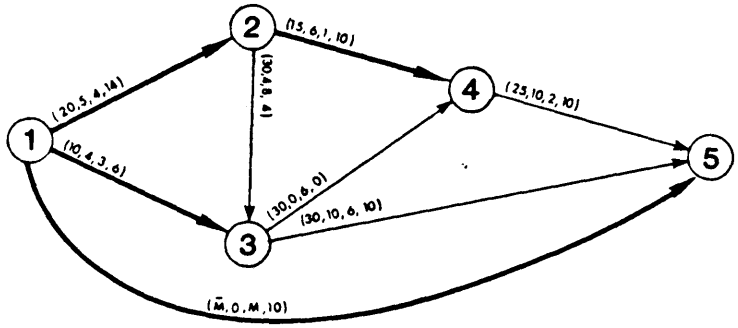

Figure 2.8: Example 1 - Results of Iteration 1.

The flow levels above were adjusted upward by ΔFLOW on all forward arcs (A_{12} and A_{24}) and downward on all backward arcs (AA_{14}). As can be seen, the flow level of AA_{14} is reduced to zero and is therefore eliminated from the network entirely.

With the tree having changed the next step requires repricing all nodes in the network, therefore go to Step 3 - NEWPI.

NEWPI:

The node prices for the second iteration are shown in Figure 2.9.

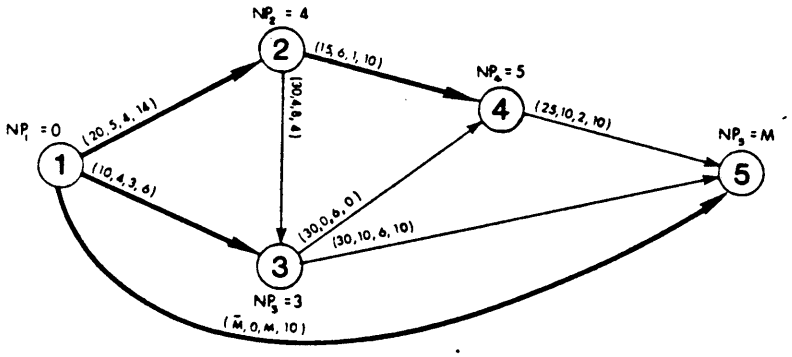

Figure 2.9: Example 1 - Iteration 2; node price calculations.

Node 1 is selected as the starting node and assigned a price of 0 ($NP_1 = 0$). From node 1 all spanning arcs leaving this node are followed to their terminal nodes. In

this case the spanning arcs are A_{12}, A_{13} and AA_{15} with $NP_2 = C_{12} = 4$, $NP_3 = C_{13} = 3$ and $NP_5 = C_{15} = M$. Node 4 still remains to have a price assigned to it. This is done by continuing from node 2, already priced out, along the last spanning arc A_{24}. $NP_4 = C_{12} + C_{24} = 5$.

PRICE

With the node prices calculated the non-spanning arcs can now be priced out remembering that $P_{ij} = NP_j - NP_i - C_{ij}$. The results are shown in Figure 2.10.

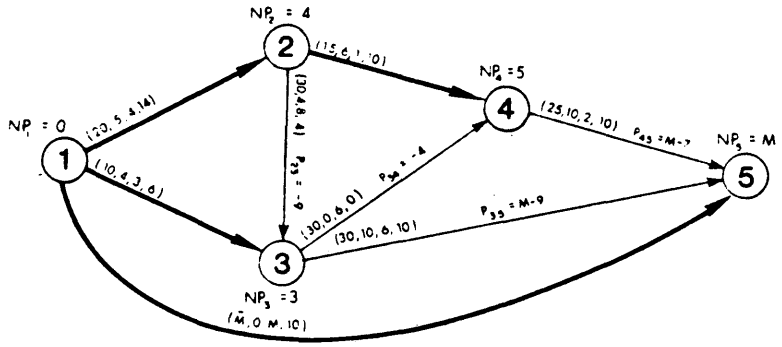

Figure 2.10: Example 1 - Iteration 2; arc price calculations.

The arc with the maximum P_{ij} value is A_{45} with $P_{45} = M - 7$. This arc is chosen as the KOMIN arc.

PATHFINDER:

A circuit is now traced from node 5 to node 4 of the KOMIN arc through the spanning arcs. The circuit will be node 5 to 1 to 2 to 4. The arcs in the circuit are AA_{15} (a backward arc), A_{12} and A_{24} (forward arcs) and finally A_{45} (the KOMIN arc).

CHANGER:

Evaluating the arcs in the circuit above the following calculations.

$\Delta FA = MIN \{6, 5\} = 5$ (from A_{24})

$\Delta BA = 10$, $\Delta KA = 15$

$\Delta FLOW = MIN \{5, 10, 15\} = 5$ (from A_{24})

The LEAVE arc is now A_{24} with $\Delta FLOW = 5$.

NEWFLOWS:

With the following information obtained in the previous steps: KOMIN = A_{45}, LEAVE = A_{24} and $\Delta FLOW = 5$, the network may now be modified. The results are shown in Figure 2.11.

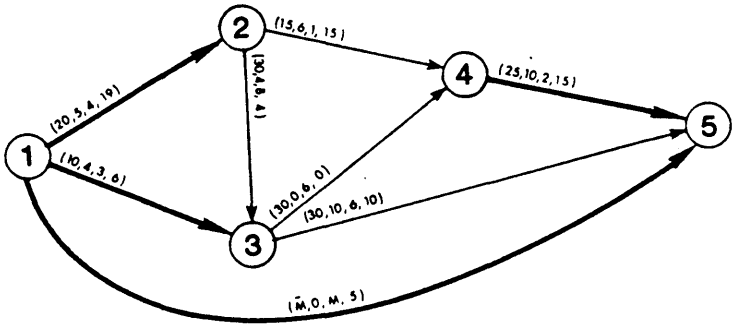

Figure 2.11: Example 1 - Results of Iteration 2.

NEWPI & PRICE:

Starting the third iteration, the nodes and non-spanning arcs must be priced out once more as shown in Figure 2.12.

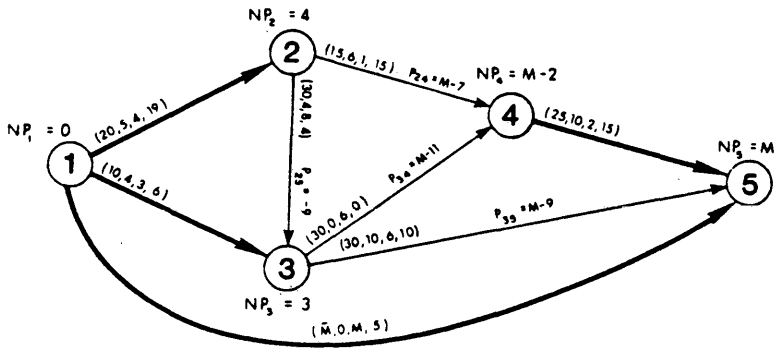

Figure 2.12: Example 1 - Iteration 3; node and arc since calculations.

The arc chosen as the KOMIN arc is A_{35} with the maximum arc price, P_{35} = M- 9.

PATHFINDER & CHANGER:

With the KOMIN arc equal to A_{35}, a circuit is now found starting from node 5 going through the spanning arcs to node 3. This circuit is: 5 to 1 to 3.

Evaluating the arcs in the circuit:

ΔFLOW = MIN {4 (for A_{13}), 5 (for AA_{15}) and 20 (for A_{35})} = 4

LEAVE = A_{13}

NEWFLOWS:

The new network can now be established and is shown in Figure 2.13.

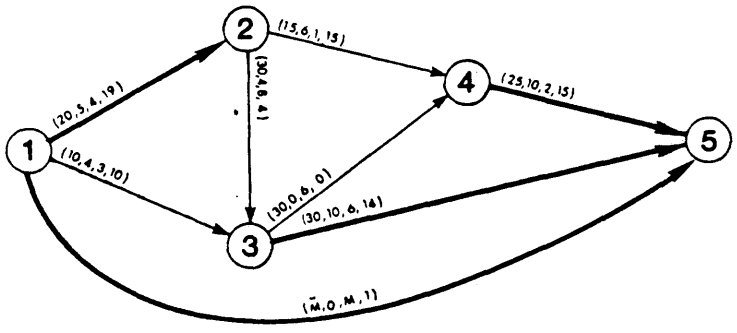

Figure 2.13: Example 1 - Results of Iteration 3.

Since the set of spanning arcs changed, all the nodes must be priced out once again, therefore return to NEWPI.

NEWPI & PRICE:

When the nodes and non-spanning arcs are priced out for the fourth time the results are shown in Figure 2.14.

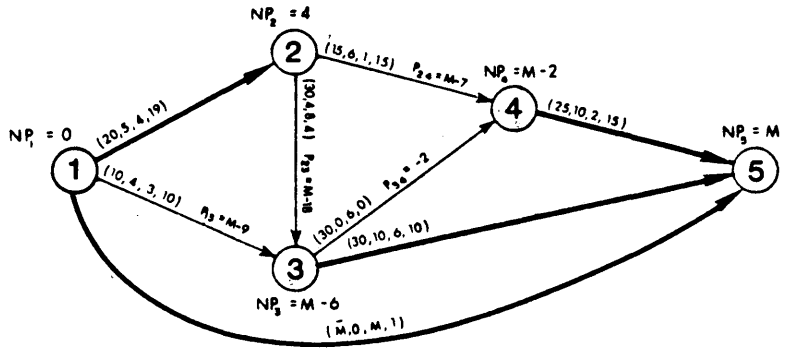

Figure 2.14: Example 1 - Iteration 4; node and arc price calculations.

In evaluating the arc prices above, P_{24} = M - 7 is the maximum price found. However A_{24} has $F_{24} = UB_{24}$ therefore this arc is not considered eligible to enter the tree. The same is true for A_{13} which is also at its upper bound. The next highest price pertains to P_{23} = M - 18, therefore KOMIN = A_{23}.

PATHFINDER & CHANGER:

In establishing a circuit from node 3 to node 2 a path is from: 3 to 5 to 1 to 2 is found. Evaluating the arcs in this circuit the following flow changes are possible:

$\Delta FA = MIN \{16, 1\} = 1$ (for A_{12})

$\Delta BA = 1$ (for AA_{15}) and $\Delta KA = 16$

$\Delta FLOW = MIN \{1, 1, 16\} = 1$ (for A_{12} or AA_{15})

The tie above is broken arbitrarily and arc AA_{15} is selected as the LEAVE arc.

NEWFLOWS:

The network is again adjusted using the above information with the results shown in Figure 2.15.

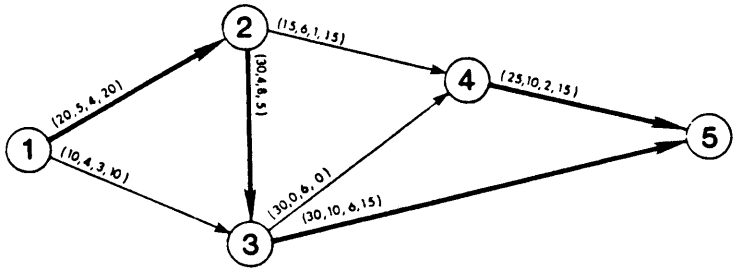

Figure 2.15: Example 1 - Results of Iteration 4.

All artificial arcs have been reduced to a flow of zero. To insure that the above figure is the optimal solution one final iteration is made.

NEWPI & PRICE:

Pricing out the nodes and non-spanning arcs results in shown in Figure 2.16.

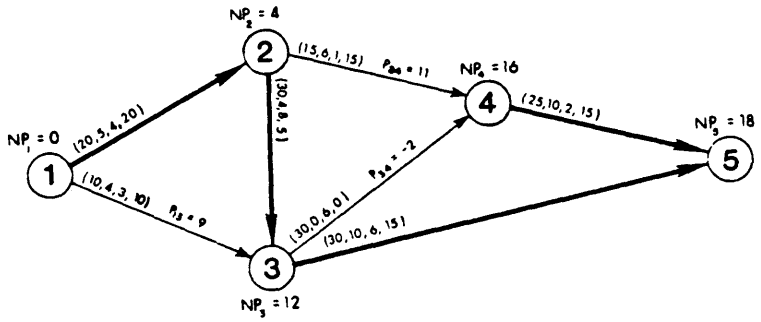

Figure 2.16: Example 1 - Iteration 5; node and arc price calculations.

Only 3 non-spanning arcs remain to be evaluated where A_{24} and A_{13}, both at their upper bounds, have a $P_{ij} > 0$ and A_{34}, at its lower bound, has $P_{34} < 0$. It can be concluded that no improvement can be made to the network and the optimal solution is shown in Figure 2.17.

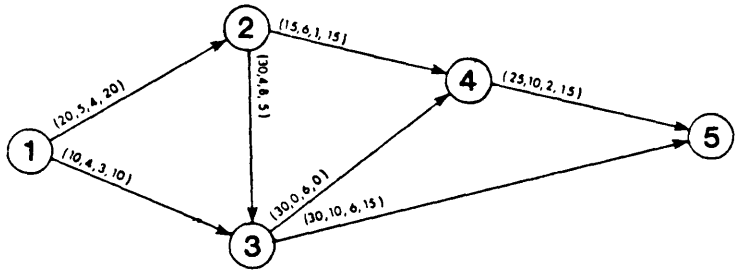

Figure 2.17: Example 1 - Final Solution.

Example 2

This example will examine the attempted solution of an infeasible network. To enhance this illustration and further illustrate the algorithm's decision rules an arc with a negative cost per unit flow is included.

A single figure is used for each iteration with each figure containing the NP_n and P_{ij} calculations for the NEWPI and PRICE steps respectively. A summary of the results of each subsequent step is also provided at each iteration. The starting network configuration is shown in Figure 2.18.

Initialization and PREGO

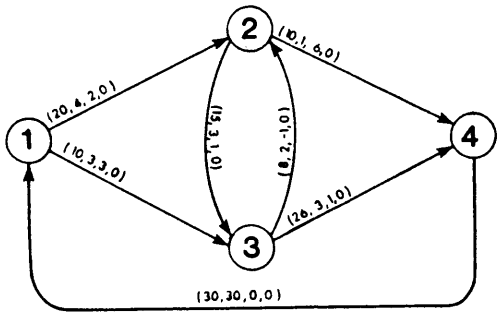

Figure 2.18: Example 2 - Initial network configuration.

The network is initialized as in the previous examples. As shown in Figure 2.18, A_{32} has a cost less than zero. The starting flow on A_{32} is therefore set at its upper bound or $F_{32} = UB_{32} = 8$. The results of the Initialization and PREGO steps are shown in the Figures 2.19 and 2.20.

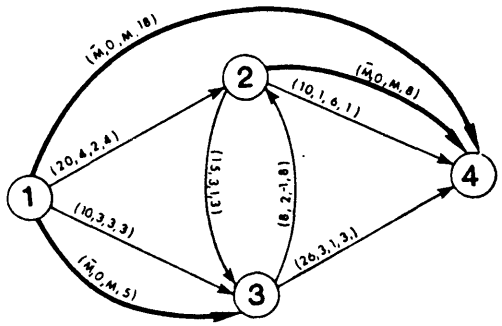

Figure 2.19: Example 2 - Artificial arcs are established.

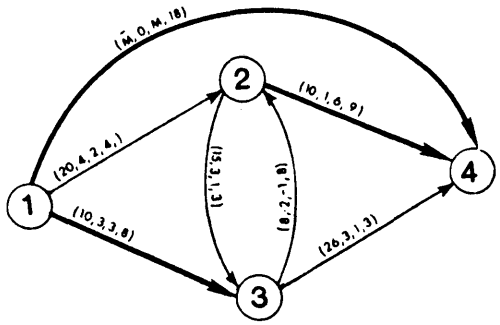

Figure 2.20: Example 2 - Results after PREGO.

Iteration 1:

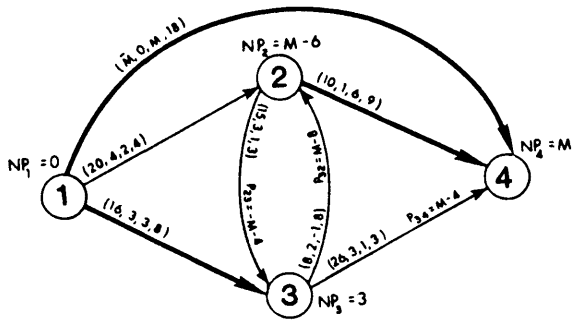

Figure 2.21: Example 2, Iteration 1.

PRICE - MAX $|P_{ij}| = P_{34} = M - 4$

KOMIN = A_{34}

PATHFINDER - The circuit = 4 to 1 to 3

CHANGER - ΔFLOW = MIN {2,18,23} = 2

LEAVE = A_{13}

NEWFLOWS - The flow over the circuit will change by 2 units and A_{34} will replace A_{13} in the tree.

Iteration 2:

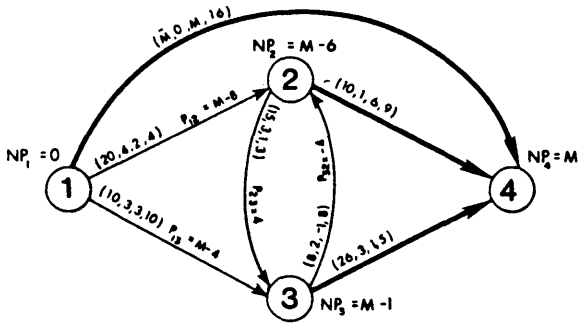

Figure 2.22: Example 2, Iteration 2.

PRICE - MAX $|P_{ij}| = P_{12} = M - 8$

KOMIN = A_{12}

(Note that P_{13} = M-4, but since F_{12} = UB_{12} this arc is not eligible to enter the tree. As in the previous example, arcs with F_{ij} = UB_{ij} will not be calculated.)

PATHFINDER - The circuit = 2 to 4 to 1

CHANGER - ΔFLOW = MIN {1, 16, 6} = 1

LEAVE A_{24}

NEWFLOWS - A_{12} will enter the tree in place of A_{24} with the flow change over the circuit being 1 unit.

Iteration 3:

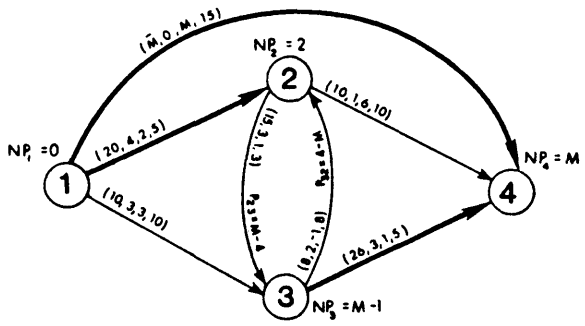

Figure 2.23: Example 2, Iteration 3.

PRICE - MAX $|P_{ij}| = P_{23} = M - 4$

$= P_{32} = 4 - M$

For A_{32}, $C_{32} = -1$; therefore the rules for the selection of eligible arcs are reversed for this arc (See Step 4 in Section 2.3) and this arc is considered ineligible to enter the basis. Therefore,

KOMIN = A_{23}

PATHFINDER - The circuit = 3 to 4 to 1 to 2

CHANGER - ΔFLOW = {15, 5, 12} = 12

LEAVE = A_{23}

NEWFLOWS - The flow over the circuit is altered by 12 units with the tree remaining the same since KOMIN = LEAVE.

Iteration 4:

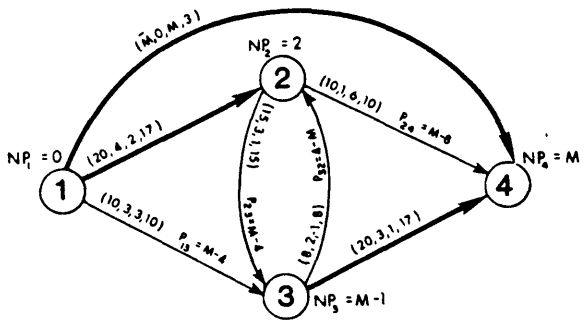

Figure 2.24: Example 2, Iteration 4.

PRICE - Given that all non-basic arcs are at their upper bounds and all $P_{ij} > 0$ (except for A_{32}) we can conclude that no arcs are eligible to enter the tree. In addition, an artificial arc (AA_{14}) remains with $F_{14} > 0$. The network is infeasible; therefore, stop.

Example 3

To further demonstrate the algorithm, a larger network is solved below. This example illustrates the following aspects of the algorithm:

a) the effects of having a flow change of zero over the circuit

b) the effects of an artificial arc with a flow of zero ($F_{ij} = 0$) and,

c) the effects of having the KOMIN arc equal the LEAVE arc.

The format used in Example 2 will be followed in this example. The starting network configuration is shown in Figure 2.25.

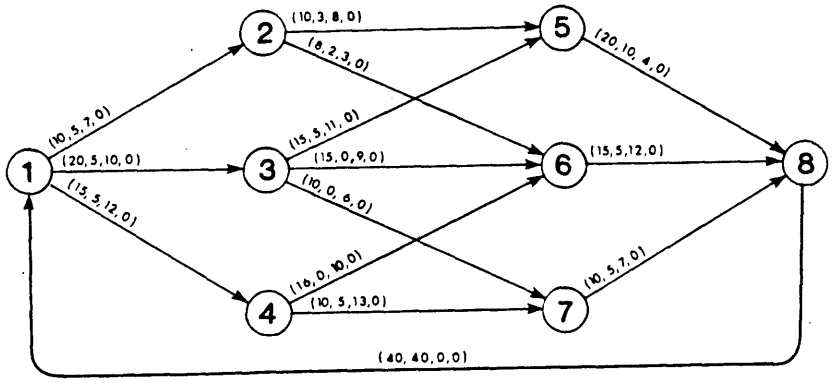

Figure 2.25: Example 3 - Initial network configuration.

Iteration 0 - Initialization

The network above is initialized by:

a) establishing the starting flows on all arcs,

b) calculating the net inflows (NI_n) at each node,

c) creating artificial arcs to connect supply nodes to demand nodes and

d) performing the PREGO step.

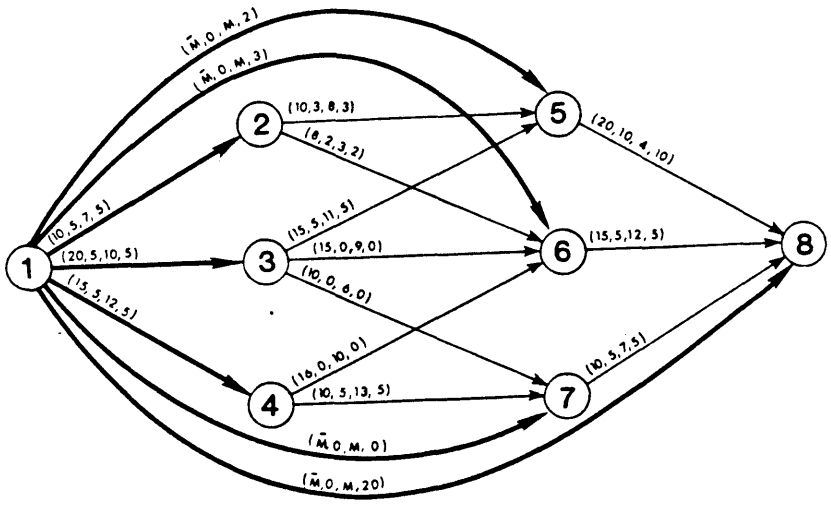

Figure 2.26: Example 3 - Results after the Initialization and PREGO steps.

Iteration 1:

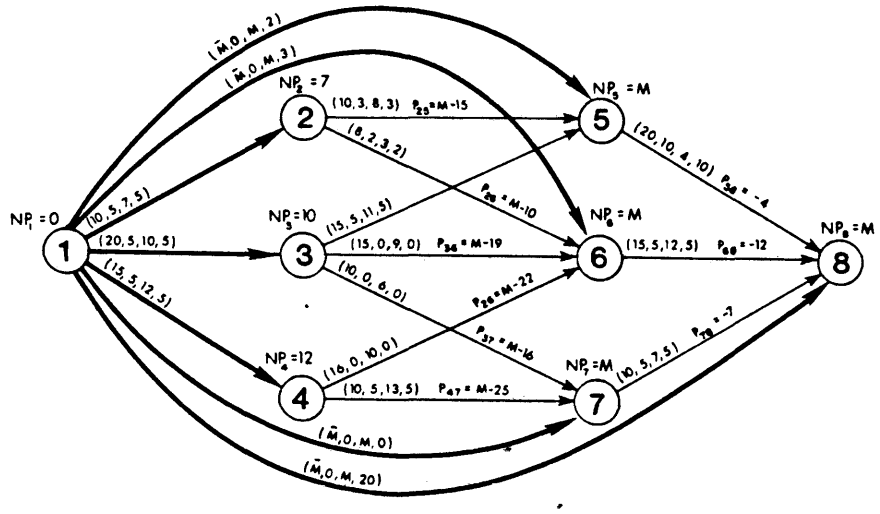

Figure 2.27: Example 3, Iteration 1.

PRICE	–	MAX $\|P_{ij}\| = P_{26} = M - 10$
		KOMIN = A_{26}
PATHFINDER	-	The circuit = 6 to 1 to 2
CHANGER	-	ΔFLOW = MIN {5, 3, 6} = 3
		LEAVE = AA_{16}
NEWFLOWS	-	Arc A_{26} will enter the tree, arc AA_{16} will leave the tree and the flow change over the circuit equals 3.

Iteration 2:

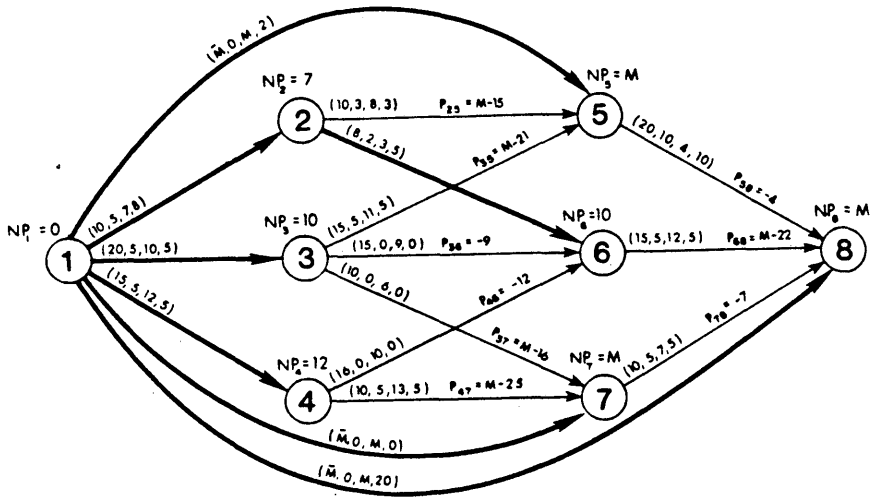

Figure 2.28: Example 3, Iteration 2.

PRICE - MAX $|P_{ij}| = P_{25} = M - 15$

KOMIN = A_{25}

PATHFINDER - The circuit = 5 to 1 to 2

CHANGER - ΔFLOW = MIN {2, 2, 7} = 2

LEAVE = AA_{15} (selected arbitrarily)

NEWFLOWS - Arc A_{25} enters and AA_{15} leaves the tree with a circuit flow change of 2.

Iteration 3:

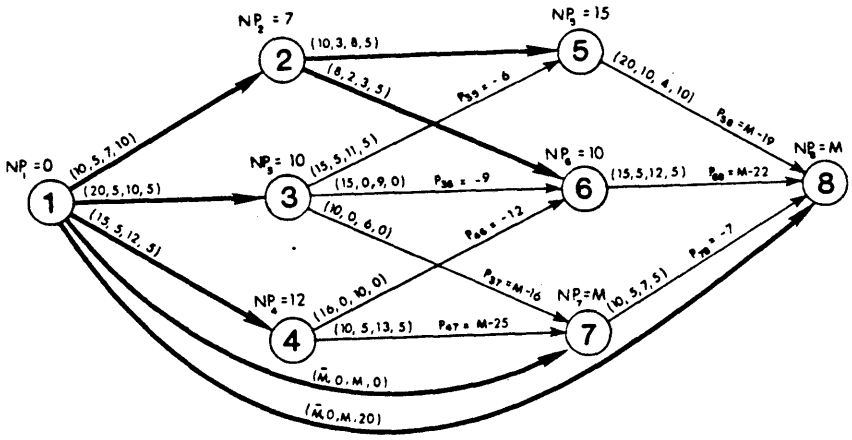

Figure 2.29: Example 3, Iteration 3.

PRICE - MAX $|P_{ij}| = P_{37} = M - 16$

KOMIN = A_{37}

PATHFINDER - The circuit = 7 to 1 to 3

CHANGER - $\Delta FLOW = MIN \{15, 0, 10\} = 0$

LEAVE = AA_{17}

NEWFLOWS - Arc A_{37} will enter and AA_{17} will leave the tree with no flow change over the circuit.

Iteration 4:

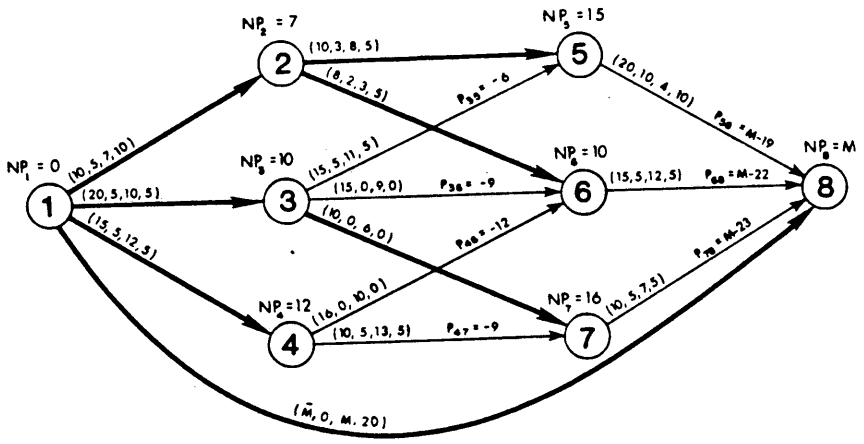

Figure 2.30: Example 3, Iteration 4.

PRICE - MAX $|P_{ij}| = P_{58} = M - 19$

KOMIN = A_{58}

PATHFINDER - The circuit = 8 to 1 to 2 to 5

CHANGER - $\Delta FLOW = MIN \{0, 20, 10\} = 0$

LEAVE = A_{12}

NEWFLOWS - As in the previous iteration, no flow change takes place; however, arc A_{58} will enter the tree in place of A_{12}.

Iteration 5:

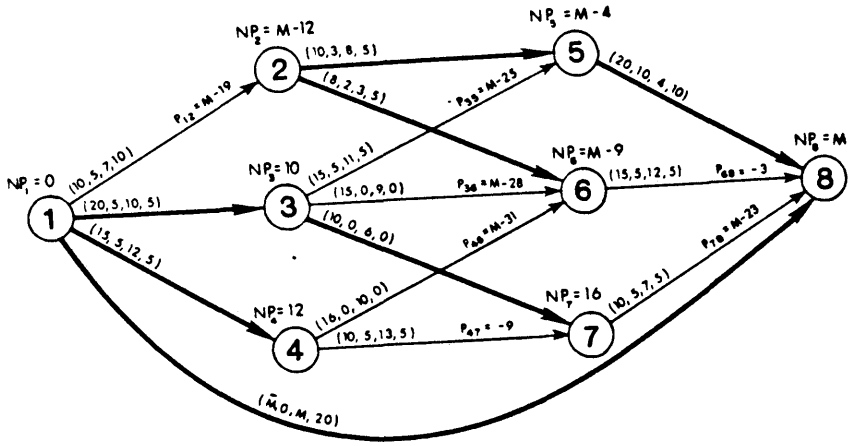

Figure 2.31: Example 3, Iteration 5.

PRICE - MAX $|P_{ij}| = P_{78} = M - 23$

KOMIN = A_{78}

(Note that $P_{12} = M - 19$, but since this arc is already at its upper bound it is not eligible to enter the tree. In the figures that follow, arc prices will not be calculated for arcs with $F_{ij} = UB_{ij}$.)

PATHFINDER - The circuit = 8 to 1 to 3 to 7

CHANGER - $\Delta FLOW = MIN \{10, 20, 5\} = 5$

LEAVE = A_{78}

NEWFLOWS - With KOMIN = LEAVE the set of spanning arcs remains unchanged. The flow over the circuit is altered by 5 units.

Iteration 6:

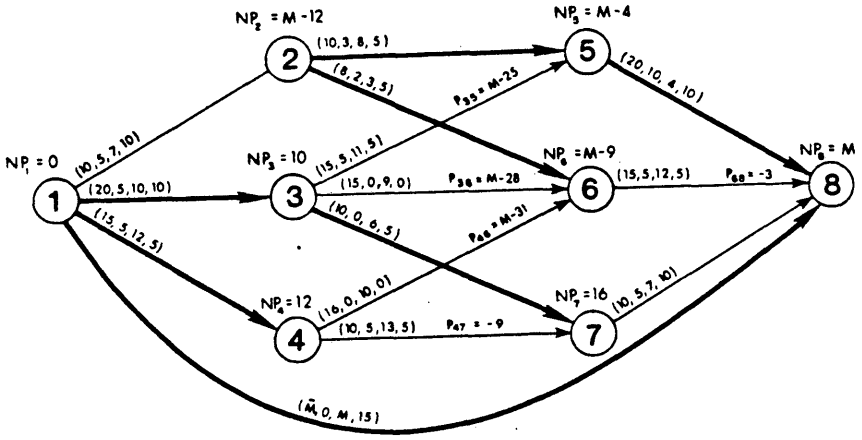

Figure 2.32: Example 3, Iteration 6.

PRICE - MAX $|P_{ij}| = P_{35} = M - 25$

KOMIN = A_{35}

PATHFINDER - The circuit = 5 to 8 to 1 to 3

CHANGER - ΔFLOW = MIN {10, 15, 10} = 10

LEAVE = A_{35} (selected arbitrarily)

NEWFLOWS - As in the previous iteration LEAVE = KOMIN therefore the set of spanning arcs remains the same with the flow change over the circuit being 10.

Iteration 7:

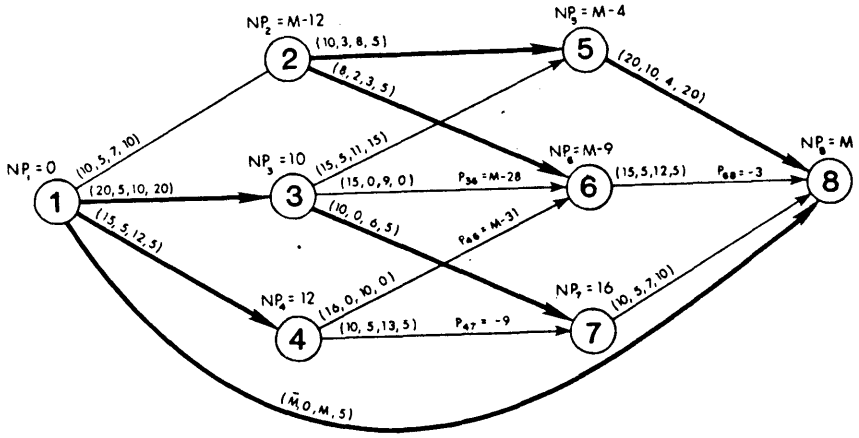

Figure 2.33: Example 3, Iteration 7.

PRICE - MAX $|P_{ij}| = P_{36} = M - 28$

KOMIN = A_{36}

PATHFINDER - The circuit = 6 to 2 to 5 to 8 to 1 to 3

CHANGER - $\Delta FLOW = MIN \{0, 3, 15\} = 0$

LEAVE = A_{58}

NEWFLOWS - Arc A_{36} will enter and A_{58} will leave the tree with no flow change taking place.

Iteration 8:

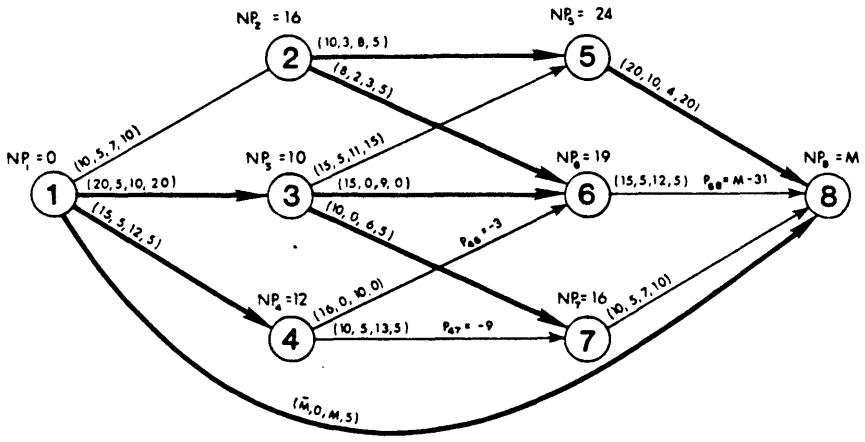

Figure 2.34: Example 3, Iteration 8.

PRICE - MAX $|P_{ij}| = P_{68} = M - 31$

KOMIN = A_{68}

PATHFINDER - The circuit = 8 to 1 to 3 to 6

CHANGER - $\Delta FLOW = MIN \{0, 5, 10\} = 0$

LEAVE = A_{12}

NEWFLOWS - The flow will remain unchanged over the circuit with A_{68} replacing A_{12} in the tree.

Iteration 9:

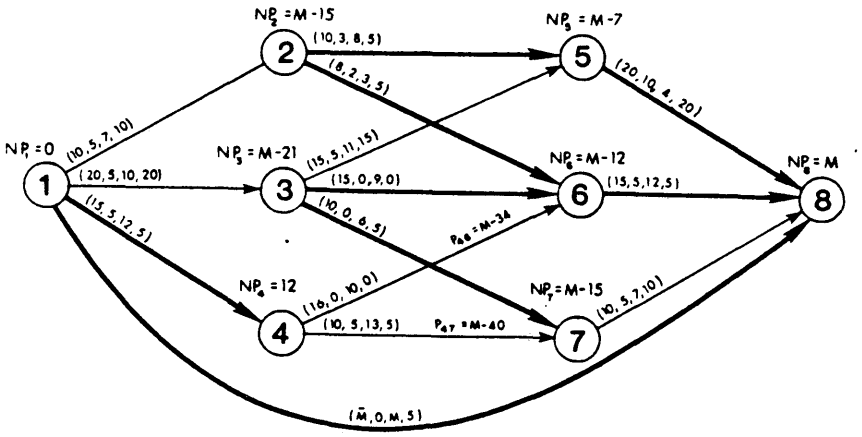

Figure 2.35: Example 3, Iteration 9.

PRICE - MAX $|P_{ij}| = P_{46} = M - 34$

KOMIN = A_{46}

PATHFINDER - The circuit = 6 to 8 to 1 to 4

CHANGER - ΔFLOW = = 5

LEAVE = AA_{18}

NEWFLOWS - Arc A_{46} will replace AA_{18} in the tree with a flow change of 5 units over the circuit.

Iteration 10:

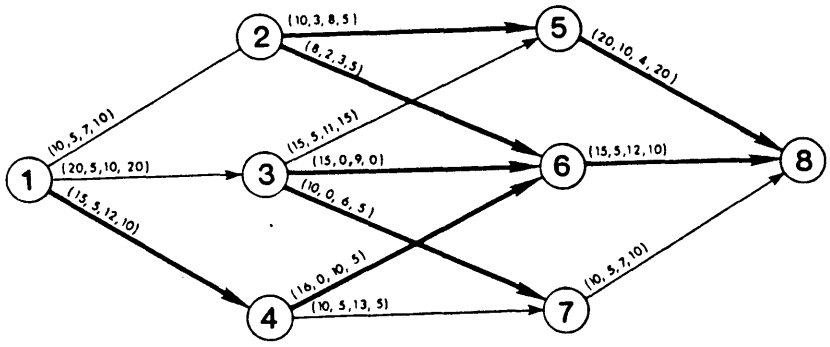

Figure 2.36: Example 3, Final Solution.

If the remaining non-spanning arcs were priced out in Figure 2.36 it would be seen that their arc prices (P_{ij}'s) would all be greater than zero. Given that each arc is also at its upper bound none would be eligible to enter the tree. In addition, since all the artificial arcs have been eliminated from the network we can conclude that the solution is optimal; therefore stop.

CHAPTER 3

THE NCA AND GENERALIZED UPPER BOUNDING

3.1 INTRODUCTION

The steps of the NCA, as described in the previous chapter, can be shown to be equivalent to the generalized upper bounding (GUB) method in linear programming. This chapter will show, by example, how these two methods compare.

To the reader unfamiliar with the GUB method, a short explanation is provided in Appendix A. For a more detailed discussion of GUB theory, the reader is referred to Dantzig-(1963).

To demonstrate the equivalency of the NCA to the GUB method, examples 1 and 2 from Chapter 2 will be used. (The solution of example 3 can be found in Appendix B.) Each network is formulated as a linear programming problem and solved using the GUB method. Following the solution of each is a comparison of the results and the methods.

3.2 EXAMPLE 1

Consider again the network shown in Figure 1. This problem can be reformulated as a linear programming problem in the following way.

MIN Z: where Z =

$$4X_{12} + 3X_{13} + 8X_{23} + X_{24} + 6X_{34} + 6X_{35} + 2X_{45}, \qquad (1)$$

subject to flow conservation requirements:

$$X_{12} + X_{13} = 30 \quad (2)$$

$$X_{12} - X_{23} - X_{24} = 0 \quad (3)$$

$$X_{13} + X_{23} - X_{34} - X_{35} = 0 \quad (4)$$

$$X_{24} + X_{34} - X_{45} = 0 \quad (5)$$

$$X_{35} + X_{45} = 30 \quad (6)$$

and subject to arc capacities:

$$5 \leq X_{12} \leq 20 \qquad 0 \leq X_{34} \leq 30$$

$$4 \leq X_{13} \leq 10 \qquad 10 \leq X_{35} \leq 30$$

$$4 \leq X_{23} \leq 30 \qquad 10 \leq X_{45} \leq 25$$

$$6 \leq X_{24} \leq 15$$

where each X_{ij} represents an arc, A_{ij}.

To facilitate the solution of the above problem, each X_{ij} is converted so that it has a lower bound of zero and

an upper bound of one. This is accomplished by using equation (7) as follows.

$$X'_{ij} = \frac{X_{ij} - LB_{ij}}{UB_{ij} - LB_{ij}} \qquad (7)$$

The problem is now reformulated by substituting each X'_{ij} into equations (1) and (3) through (6). (Constraint (2) is eliminated from the reformulation as it is a linear combination of (3) through (6) and is therefore redundant.) The resulting problem is

MIN Z: where Z =

$$60X'_{12} + 18X'_{13} + 208X'_{23} + 9X'_{24} + 180X'_{34} + 120X'_{35} + 30X'_{45} + 150,$$

Subject to:

$$\begin{aligned}
15X'_{12} - 26X'_{23} - 9X'_{24} &= 5 \\
6X'_{13} + 26X'_{23} - 30X'_{34} - 20X'_{35} &= 2 \\
9X'_{24} + 30X'_{34} - 15X'_{45} &= 4 \\
20X'_{35} + 15X'_{45} &= 10.
\end{aligned}$$

The next step is to convert the reformulated problem into a standard form. This is done by adding artificial variables to each constraint using the Big M Method as explained by Hillier and Lieberman (1974). The problem in standard form is shown in Table 3.1.

Basic Var.	Row No.	Coefficient of												Right Side
		Z	x'_{12}	x'_{13}	x'_{23}	x'_{24}	x'_{34}	x'_{35}	x'_{45}	A_1	A_2	A_3	A_4	
Z	0	-1	60	18	208	9	180	120	3	M	M	M	M	-150
A_1	1	0	15	0	-26	-9	0	0	0	1	0	0	0	5
A_2	2	0	0	6	26	0	-30	-20	0	0	1	0	0	2
A_3	3	0	0	0	0	9	30	0	-15	0	0	1	0	4
A_4	4	0	0	0	0	0	0	20	15	0	0	0	1	10

TABLE 3.1: Example 1 - Simplex Tableau for the Initial Formulation

By adding the linear combinations of rows through 4, each multiplied by a "-M," to row (0) the A_i (basic variable) coefficients are reduced to zero. The results are shown in Table 3.2.

To aid in the comparison of the NCA to the GUB method, an additional preliminary step is taken to establish the same starting basic solution*. This step consists of bringing into the basis those variables brought in in the PREGO step, variables X_{12} and X_{13} (See figures 2.3 and 2.4). The starting basic solution from which the GUB method proceeds is shown in Table 3.3

In Table 3.3 an additional row, row (5), has been included. The entries in this row reflect the results of dividing each C_{ij} in row (0), by the appropriate (UB_{ij} - LB_{ij}) value for each variable or

$$\bar{C}_{ij} = \frac{C_{ij}}{UB_{ij} - LB_{ij}} \qquad (8)$$

The $\bar{C}_{ij}$ entries in row (5) are then used to determine the entering basic variable as follows: for those X_{ij} with $C_{ij} < 0$ in row (0) select as the incoming basic variable that variable with the maximum $\bar{C}_{ij}$ value in row (5).

* Note: The spanning tree and spanning arcs are equivalent to the basis and basic variables respectively.

Basic Var.	Row No.	Coefficient of											Right Side
		Z	X'_{12}	X'_{13}	X'_{23}	X'_{24}	X'_{34}	X'_{35}	X'_{45}	A_1	A_2	A_3	A_4	
Z	0	-1	-15M+60	-6M+18	208	9	180	120	3	0	0	0	0	-21M-150
A_1	1	0	15	0	-26	-9	0	0	0	1	0	0	0	5
A_2	2	0	0	6	26	0	-30	-20	0	0	1	0	0	2
A_3	3	0	0	0	0	9	30	0	-15	0	0	1	0	4
A_4	4	0	0	0	0	0	0	20	15	0	0	0	1	10

TABLE 3.2: Example 1 - Simplex Tableau in standard form with the coefficients of the basic variables in row (0) equal to zero

Basic Var.	Row No.	Coefficient of												Right Side
		z	x'_{12}	x'_{13}	x'_{23}	x'_{24}	x'_{34}	x'_{35}	x'_{45}	A_1	A_2	A_3	A_4	
z	0	-1	0	0	234	-9M+45	-30M+270	-20M+180	30	M-4	M-3	0	0	-14M-176
x'_{12}	1	0	1	0	-26/15	-9/5	0	0	0	1/15	0	0	0	5/15
x'_{13}	2	0	0	1	26/6	0	-30/6	-20/6	0	0	1/6	0	0	2/6
A_3	3	0	0	0	0	9	30	0	-15	0	0	1	0	4
A_4	4	0	0	0	0	0	0	20	15	0	0	0	1	10
C_{ij}	5	-	0	0	9	M-5	M-9	M-9	2	-	-	-	-	-

TABLE 3.3: Example 1 - Results after bringing X_{12} and X_{13} into the basis

Iteration 1:

Only the variables X'_{24}, X'_{34} and X'_{35} are eligible candidates to enter the basis. Table 3.3 shows X'_{24} is selected as the entering basic variable with the max C_{ij} value of M - 5.

To determine the leaving basic variable the coefficients of X'_{24} in Table 3.3 must be evaluated (excluding row (5) to determine the maximum feasible value of X'_{24} such that the current basic variables do not exceed their bounds ($0 \leqslant X_{ij} \leqslant 1$).

Row	Basic Var.	Max Feasible Value of X'_{24}
0	Z	$X'_{24} \leqslant 1$
1	X_{12}	$X'_{24} \leqslant 10/9$ or $1 = 5/15 + 9/15X'_{24}$
*3	A_3	$X'_{24} \leqslant 4/9$

TABLE 3.3a: Example 1 - Evaluation of X'_{24} Coefficients to determine the leaving basic variable

The minimum value above, pertaining to A_3, is the maximum feasible value of X'_{24}. A_3 is therefore selected as the leaving basic variable. The appropriate pivot operation is made with the results shown in Table 3.4.

Basic Var.	Row No.	Coefficient of											Right Side
		Z	X'_{12}	X'_{13}	X'_{23}	X'_{24}	X'_{34}	X'_{35}	X'_{45}	A_1	A_2	A_3	A_4	
Z	0	-1	0	0	234	0	120	-20M+180	-15M+105	M-4	M-3	M-5	0	-10M-196
X'_{12}	1	0	1	0	-26/15	0	30/15	0	-1	1/15	0	1/15	0	9/15
X'_{13}	2	0	0	1	26/6	0	-30/6	-20/6	0	0	1/6	0	0	2/6
X'_{24}	3	0	0	0	0	1	30/9	0	-15/9	0	0	1/9	0	4/9
A_4	4	0	0	0	0	0	0	20	15	0	0	0	1	10
C_{ij}	5	-	0	0	9	0	4	M-9	M-7	-	-	-	-	-

TABLE 3.4: Example 1 - Results of Iteration 1

Iteration 2:

Table 3.4 shows that X'_{45} is selected as the entering basic variable from the set of eligible variables, X'_{35} and X'_{45}. Evaluating the X'_{45} coefficients we get the results shown in Table 3.4a.

Row	Basic Var.	Max Feasible Value of X'_{45}
0	Z	$X'_{45} \leqslant 1$
1	X'_{45}	$X'_{45} \leqslant 6/15$ or $1 = 9/15 + X'_{45}$
*3	X'_{24}	$X'_{45} \leqslant 5/15$ or $1 = 4/9 + 15/9X'_{45}$
4	A_4	$X'_{45} \leqslant 10/15$

Table 3.4a: Example 1 - Evaluation of X'_{45} Coefficients to determine the leaving basic variable

The leaving basic variable will be Y_{24} as X'_{24} (with a negative X'_{45} coefficient) determined the maximum feasible value of X'_{45} with $X'_{45} \leqslant 5/15$. The complement of X'_{24}, $(1 - Y_{24})$, replaces X'_{24} in row (3) and Y_{24} becomes the leaving basic variable. Table 3.5 shows the results of these operations.

Basic Var.	Row No.	Coefficient of											Right Side
		Z	X'_{12}	X'_{13}	X'_{23}	Y_{24}	X'_{34}	X'_{35}	X'_{45}	A_1	A_2	A_3	A_4	
Z	0	-1	0	0	234	9M-63	-30M+330	-20M+180	0	M-4	M-3	2	0	-5M-231
X'_{12}	1	0	1	0	-26/15	9/15	0	0	0	1/15	0	0	0	14/15
X'_{13}	2	0	0	1	26/6	0	-30/6	-20/6	0	0	1/6	0	0	2/6
X'_{45}	3	0	0	0	0	9/15	-30/15	0	1	0	0	-1/15	0	5/15
A_4	4	0	0	0	0	-9	30	20	0	0	0	1	1	5
C_{ij}	5	-	0	0	9	M-7	M-11	M-9	0	-	-	-	-	-

TABLE 3.5: Example 1 - Results of Iteration 2

Iteration 3:

Table 3.5 shows that X'_{35} will be the next entering basic variable from the set of eligible variables X'_{34} and X'_{35}. Evaluating the X'_{35} coefficients we get the results shown in Table 3.5a.

Row	Basic Var.	Max Feasible Value of X'_{35}
0	Z	$X'_{35} \leq 1$
*2	X_{13}	$X'_{35} \leq 4/20$ or $1 = 2/6 + 20/6X'_{35}$
4	A_4	$X'_{35} \leq 5/20$

Table 3.5a: Example 1 - Evaluation of the X'_{35} Coefficients

The leaving basic variable will be Y_{13} as X'_{13} (with a negative X'_{35} coefficient) determined the maximum feasible value of X'_{35}. X'_{13} is replaced by $(1 - Y_{13})$ in row (2) then the pivot step takes place on the leaving basic variable Y_{13}. Table 3.6 reflects these results.

Basic Var.	Row No.	Coefficient of												Right Side
		z	x'_{12}	y_{13}	x'_{23}	y_{24}	x'_{34}	x'_{35}	x'_{45}	A_1	A_2	A_3	A_4	
z	0	-1	0	6M-54	-26M+468	9M-63	60	0	0	M-4	6	2	0	-M-267
x'_{12}	1	0	1	0	-26/15	9/15	0	0	0	1/15	0	0	0	14/15
x'_{35}	2	0	0	6/20	-26/20	0	30/20	1	0	0	-1/20	0	0	4/20
x'_{45}	3	0	0	0	0	9/15	-30/15	0	1	0	0	-1/15	0	5/15
A_4	4	0	0	-6	26	-9	0	0	0	0	1	1	1	1
C_{ij}	5	-	0	M-9	M-18	M-7	2	0	0	-	-	-	-	-

TABLE 3.6: Example 1 - Results of Iteration 3

Iteration 4:

Row (0) in Table 3.6 reveals that only X'_{23} is eligible to be an entering basic variable. Evaluating the X'_{23} coefficients we get the results in Table 3.6a.

Row	Basic Var.	Max Feasible Value of X_{23}
0	2	$X'_{23} \leq 1$
*1	X'_{12}	$X'_{23} \leq 1/26$ or $1 = 14/15 + 26/15X'_{23}$
2	X'_{35}	$X'_{23} \leq 16/26$ or $1 = 4/20 + 26/20X'_{23}$
*4	A_4	$X'_{23} \leq 1/26$

Table 3.6a: Example 1 - Evaluation of the X_{23} Coefficients

A tie exists between X'_{12} (or $1 - Y_{12}$) and A_4 as the leaving basic variable. To minimize the number of computations needed A_4 is selected. See Table 3.7 for the results.

Basic Var.	Row No.	Coefficient of												Right Side
		Z	X'_{12}	Y_{13}	X'_{23}	Y_{24}	X'_{34}	X'_{35}	X'_{45}	A_1	A_2	A_3	A_4	
Z	0	-1	0	54	0	99	60	0	0	M-4	M-12	M-16	M-18	-285
X'_{12}	1	0	1	0	0	0	0	0	0	1/15	1/15	1/15	1/15	1
X'_{35}	2	0	0	0	0	-9/20	30/20	1	0	0	0	1/20	1/20	5/20
X'_{45}	3	0	0	0	0	9/15	-30/15	0	1	0	0	-1/15	0	5/15
X'_{23}	4	0	0	-6/26	1	-9/26	0	0	0	0	1/26	1/26	1/26	1/26
C_{ij}	-	-	0	9	0	11	2	0	0	-	-	-	-	-

TABLE 3.7: Example 1 - Results of Iteration 4 and the final tableau

Iteration 5:

When the coefficients in row (0) are evaluated it can be seen that no eligible variables remain to enter the basis. In addition, since all the artificial variables have been driven out of the basis, one can conclude the solution shown in Table 3.7 is optimal with Z = 285. The solution variables are

$X'_{12} = 1$ $X'_{34} = 0$

$X'_{13} = 1$ $X'_{35} = 5/20$

$X'_{23} = 1/26$ $X'_{45} = 5/15$

$X'_{24} = 1$

Converting the X'_{ij}'s to their appropriate X_{ij} values one will get

$X_{12} = 20$ $X_{34} = 0$

$X_{13} = 10$ $X_{35} = 15$

$X_{23} = 5$ $X_{45} = 15$

$X_{24} = 15$

3.3 - EXAMPLE 1: COMPARISON OF RESULTS

Comparing the solution values of the X_{ij}'s on the previous page to the F_{ij}'s in Figure 2.17, one sees that the same values result. A closer look at the intermediate results of both methods shows a similar correspondence. For example, the arc prices (P_{ij}'s) calculated in Step 4 of the NCA are equivalent to the values calculated in row (5) of the tables for the GUB method. A comparison of Figures 2.7, 2.10, 2.12, 2.14 and 2.16 to Tables 3.3 through 3.7 illustrates this.

To illustrate further similarities, consider again Figure 2.10 and Table 3.4 shown again on the following pages. In both cases A_{45} or X'_{45} is selected as the entering basic arc or variable. Looking at the coefficients for X'_{45} in rows (1) - (4) it can be seen that:

(a) The non-zero coefficients for X'_{45} pertain to the basic variables (arcs) X'_{12}, X'_{24} and A_4. This is the same set of arcs in the circuit for the PATHFINDER step of the NCA.

(b) The signs of the X'_{45} coefficients indicate the type of arc the basic variables correspond to; that is, a backward or forward arc by

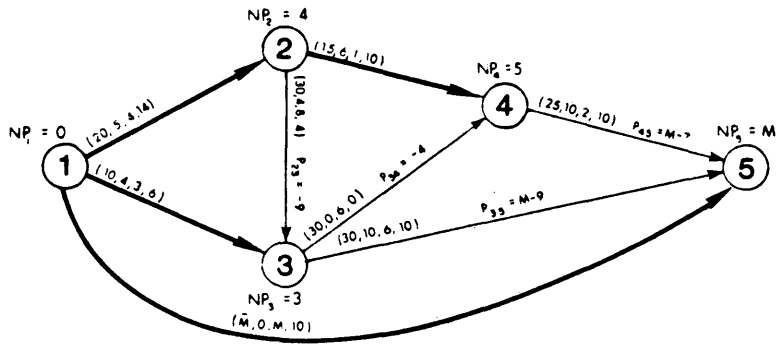

Figure 2.10: Example 1 - Iteration 2; arc price calculations.

Basic Var.	Row No.	Coefficient of											Right Side
		Z	X'_{12}	X'_{13}	X'_{23}	X'_{24}	X'_{34}	X'_{35}	X'_{45}	A_1	A_2	A_3	A_4	
Z	0	-1	0	0	234	0	120	-20M+180	-15M+105	M-4	M-3	M-5	0	-10M-196
X'_{12}	1	0	1	0	-26/15	0	30/15	0	-1	1/15	0	1/15	0	9/15
X'_{13}	2	0	0	1	26/6	0	-30/6	-20/6	0	0	1/6	0	0	2/6
X'_{24}	3	0	0	0	0	1	30/9	0	-15/9	0	0	1/9	0	4/9
A_4	4	0	0	0	0	0	0	20	15	0	0	0	1	10
C_{ij}	5	-	0	0	9	0	4	M-9	M-7	-	-	-	-	-

TABLE 3.4: Example 1 - Results of Iteration 1

(i) a coefficient less than zero indicates a forward arc over which flow will increase, and

(ii) a coefficient greater than zero indicates a backward arc over which flow will decrease.

(c) Evaluating each basic variable to determine the leaving basic variable is equivalent to Step 6, CHANGER, in the NCA. For example, in CHANGER the following values were calculated:

$\Delta FA = \{6 \text{ (for } A_{12}) \text{ or } 5 \text{ (for } A_{24})\} = 5$

$\Delta BA = 10$ $\Delta KA = 15$ $*\Delta FLOW = 5$

In the GUB method the following was calculated:

Row 1: $(X'_{12}) \rightarrow X'_{45} \leq 6/15$ or converting to X_{45} terms $X_{45} \leq 6$

*Row 3: $(X'_{24}) \rightarrow X'_{45} \leq 5/15$ or $X_{45} \leq 5$

Row 4: $(A_4) \rightarrow X'_{45} \leq 10/15$ or $X_{45} \leq 10$

Row 0: $(X'_{45}) \rightarrow X'_{45} \leq 1$ or $X_{45} \leq 15$

The calculations for rows (1) and (3) are the ΔFA values, row (4) = ΔBA and row (0) = ΔKA.

To summarize, each step of the NCA can be tied to the values in the simplex tableus

(a) The C_{ij}'s in row 0 when reduced to C_{ij} values correspond to P_{ij}'s calculated in PRICE.

(b) The non-zero coefficients for the incoming variable correspond to the arcs (variables) in the circuit in the PATHFINDER step with the signs of the coefficients indicating the arcs' direction.

(c) The steps for determining the leaving basic variable are equivalent to the CHANGER step.

It should be noted that these direct correspondences can be found only when the same starting basic solution is used. For this example we made an extra initialization step in the GUB method equivalent to the PREGO step in the NCA to establish this starting basic solution.

3.4 - EXAMPLE 2

Example 2 from Chapter 2 will be considered next (see Figure 2.17). This problem can be formulated as a linear programming problem as follows:

MIN Z: where Z =

$$2X_{12} + 3X_{13} + X_{23} + 6X_{24} - X_{32} + X_{34} \qquad (9)$$

Subject to the flow conservation requirements:

$$X_{12} + X_{13} = 30 \quad (10)$$

$$X_{12} - X_{23} - X_{24} + X_{32} = 0 \quad (11)$$

$$X_{13} + X_{23} - X_{32} + X_{34} = 0 \quad (12)$$

$$X_{24} + X_{34} = 30 \quad (13)$$

and subject to the arc capacities:

$$4 \leqslant X_{12} \leqslant 20 \qquad 1 \leqslant X_{24} \leqslant 10$$

$$3 \leqslant X_{13} \leqslant 10 \qquad 2 \leqslant X_{32} \leqslant 8$$

$$3 \leqslant X_{23} \leqslant 15 \qquad 3 \leqslant X_{34} \leqslant 26.$$

(Equation (10) is excluded from further consideration as it is redundant.)

Converting the X_{ij}'s into X'_{ij} variables with lower bounds of zero and upper bounds of one by equation (7) the problem becomes

MIN Z: where Z =

$$32X_{12} + 21X_{13} + 12X_{23} + 54X_{24} - 6X_{32} + 23X_{34},$$

Subject to

$$\begin{aligned} 16X_{12} - 12X_{23} - 9X_{24} + 6X_{32} &= -2 \\ 7X_{13} + 12X_{23} - 6X_{32} - 23X_{34} &= -1 \\ 9X_{24} + 23X_{34} &= 26. \end{aligned}$$

Adding the artificial variables to the constraints yields the problem shown in Table 3.8.

As in the previous example, to demonstrate the equivalence of the two methods the same starting basic solution must be established. Given that this problem has an arc with a negative cost (X_{32}), this variable is put at its upper bound in the starting solution. This is done by substituting $(1 - Y_{32})$ into the X_{32} column. Table 3.9 shows the results of this operation as well as that of reducing the C_{ij} values for the artificial variables to zero.

An additional preliminary step, PREGO, is needed to achieve the same starting basic solution as is shown in Figure 2.20. This requires bringing X_{13} and X_{24} into the basis which is shown in Table 3.10.

Basic Var.	Eq. No.	Coefficient of										Right Side
		Z	X'_{12}	X'_{13}	X'_{23}	X'_{24}	X'_{32}	X'_{34}	A_1	A_2	A_3	
Z	0	-1	32	21	12	54	-6	23	M	M	M	-27
A_1	1	0	16	0	-12	-9	6	0	1	0	0	-2
A_2	2	0	0	7	12	0	-6	-23	0	1	0	-1
A_3	3	0	0	0	0	9	0	23	0	0	1	26

TABLE 3.8: Example 2 - Initial formulation

Basic Var.	Eq. No.	Coefficient of										Right Side
		Z	X'_{12}	X'_{13}	X'_{23}	X'_{24}	Y_{32}	X'_{34}	A_1	A_2	A_3	
Z	0	-1	16M+32	-7M+21	-24M+12	-18M+54	-12M+6	23	0	0	0	-39M-21
A_1	1	0	-16	0	12	9	6	0	1	0	0	8
A_2	2	0	0	7	12	0	6	-23	0	1	0	5
A_3	3	0	0	0	0	9	0	23	0	0	1	26

TABLE 3.9: Example 2 - The problem in standard form with X_{32} at its upper bound

Basic Var.	Eq. No	Coefficient of										Right Side
		Z	X'_{12}	X'_{13}	X'_{23}	X'_{24}	Y_{32}	X'_{34}	A_1	A_2	A_3	
Z	0	-1	-16M+128	0	12M-84	0	6M-48	-23M+92	2M-6	M-3	0	-18M-84
X'_{24}	1	0	-16/9	0	12/9	1	6/9	0	1/9	0	0	8/9
X'_{13}	2	0	0	1	12/7	0	6/7	-23/7	0	1/7	0	5/7
A_3	3	0	16	0	-12	0	-6	23	-1	0	1	18
$\bar{c}_{ij}$	4	-	M-8	0	M-7	0	M-8	M-4	-	-	-	-

TABLE 3.10: Example 2 - Results after bringing X_{13} and X_{24} into the basis

Iteration 1:

An examination of the the row (4) entries in Table 3.10 reveals that X'_{34}, with the largest C_{ij} value, is the entering basic variable. The coefficients of X'_{34} reveals that X'_{13} will leave the basis at its upper bound and become Y_{13}. Table 3.11 contains the results of the pivot operation.

Basic	Eq.		Coefficient of									Right
Var.	No.	Z	X'_{12}	Y_{13}	X'_{23}	X'_{24}	Y_{32}	X'_{34}	A_1	A_2	A_3	Side
Z	0	-1	-16M+128	7M-28	-48	0	-24	0	2M-6	1	0	-16M-92
X'_{24}	1	0	-16/9	0	12/9	1	6/9	0	1/9	0	0	8/9
X'_{34}	2	0	0	7/23	-12/23	0	-6/23	1	0	-1/23	0	2/23
A_3	3	0	16	-7	0	0	0	0	-1	1	1	16
$\bar{C}_{ij}$	4	-	M-8	M-4	4	0	4	0	-	-	-	-

TABLE 3.11: Example 2 - Results of Iteration 1

Iteration 2:

Examining the row (0) coefficients in Table 3.11 it can be seen that X'_{12} is the entering basic variable. Evaluating the X'_{12} coefficients we see that X'_{24} will leave the basis at its upper bound and become Y_{24}. The results are shown in Table 3.12.

Basic	Eq.	Coefficient of									Right
Var.	No.	X'_{12}		Y_{13}	X'_{23}	Y_{24}	Y_{32}	X'_{34}	A_1	A_2	A_3	Side
z	0	-1	0	7M-28	-12M+48	9M-72	-6M+24	0	M+2	1	0	-15M-100
X'_{12}	1	0	1	0	-12/16	9/16	-6/16	0	-1/16	0	0	1/16
X'_{34}	2	0	0	7/23	-12/23	0	-6/23	1	0	-1/23	0	2/23
A_3	3	0	0	-7	12	-9	6	0	0	1	1	15
$\bar{c}_{ij}$	4	-	0	M-4	M-4	M-8	M-4					

TABLE 3.12: Example 2 - Results of Iteration 2

Iteration 3:

The next entering basic variable will be X'_{23}. Evaluating the coefficients for X'_{23} we see that Y_{23} will be the leaving basic variable as X'_{23} is driven to its upper bound.

The results in Table 3.13 indicate that no additional variables are candidates to enter the basis. In addition, the artificial variable, A_3, remains in the basis indicating the problem is infeasible.

Basic	Eq.	Coefficient of										Right
Var.	No.	Z	X'_{12}	Y_{13}	Y_{23}	Y_{24}	Y_{32}	X'_{34}	A_1	A_2	A_3	Side
Z	0	-1	0	7M-28	12M-48	9M-72	-6M+24	0	M+2	1	0	-3M-148
X'_{12}	1	0	1	0	12/16	9/16	-6/16	0	-1/16	0	0	13/6
X'_{34}	2	0	0	7/23	12/23	0	-6/23	1	0	-1/23	0	14/23
A_3	3	0	0	-7	-12	-9	6	0	0	1	M	3

TABLE 3.13: Example 2 - Results of Iteration 3

3.5 - EXAMPLE 2: COMPARISON OF RESULTS

As with Example 1 in 3.4, the same similarities between the two methods can be found and will not be repeated. One aspect of this example should be examined, however. The NCA in its initialization step sets flow levels on those arcs with a negative cost per unit of flow at their upper bounds. Secondly, the PREGO step brings into the basis arcs A_{24} and A_{13}. If these two principles of the NCA had not been applied when establishing a starting basic solution in the GUB method the direct similarities do not exist as clearly. The resulting answer, i.e., the problem is infeasible, however remains the same.

CHAPTER 4

A COMPARISON OF THE NODE CHAIN AND OUT OF-KILTER ALGORITHMS

4.1 INTRODUCTION

Where Chapter 3 examined the equivalent of the NCA to the GUB method in linear programming this chapter will look at the NCA as it compares to another network flow algorithm, the Out-of-Kilter Algorithm (OKA), in terms of computational efficiency. The OKA is currently one of the most widely used algorithms for the solution of a wide variety of network flow problems. As the NCA will also solve the same types of problems the OKA was considered a good basis for comparison of the NCA from a network flow standpoint.

In the framework of network flow analysis two aspects of each algorithm are examined. First, the similarities and differences between the two algorithms are discussed. Secondly, the computational efficiency of each is compared in terms of execution time and number of iterations required to solve various sets of network.

4.2 THE SIMILARITIES

A convenient similarity between the two algorithms is that they both require the same network structures for their solution processes. Both algorithms require a pure network

configuration with directed and capacitated arcs. This provides a sound basis for comparing the two methods.

Both algorithms will solve a wide variety of capacitated network problems: transportation, minimum cost, assignment, maximum flow, transshipment and shortest path. In all cases tested the results obtained from both methods were identical for total network cost (if any) and arc flows.

Both the NCA and OKA require only a knowledge of simple algebra to learn and the same basic notational conventions are used.

4.3 THE DIFFERENCES

A key difference found in the two algorithms lies in the steps of each algorithm. The OKA uses complicated decision rules and tables to determine what arcs are to be evaluated and how much flow change is to take place over specific sets of arcs. The NCA, on the other hand, follows a few simple decision rules which can be more easily taught and summarized.

An obvious difference in the two is that the NCA is a primal whereas the OKA is a dual algorithm. The NCA maintains primal feasibility at all times by the inclusion of artificial arcs (variables) and by another important step in the initialization step of the NCA all arc flows are set at

their lower bounds (upper bounds) when the arc cost is positive (zero or negative). This means that if a large proportion of arcs in the network have positive lower bounds the NCA will begin with a solution much closer to optimality than the OKA. In general, the OKA begins with all arc flows equal to zero regardless of the lower bound constraints on those arcs.

4.4 COMPUTATIONAL COMPARISONS

In order to study the computational efficiency of the two algorithms two types of data were selected for comparison. The first, the number of iterations and the second, CPU seconds required for each case run. The cases were run on the DEC-1091 at the Colorado School of Mines using a library program for the OKA and a modified version of the original NCA program written by Drs. Raike and Woolsey.

Two categories of test runs were made for each algorithm. The first consisted of sets of problems where all the arc lower bounds were equal to zero. In the second category lower bounds greater than zero were included for the same basic network configurations used in the first.

In each category three sets of problems were run with the density of the networks increasing from set 1 to 3. Each set consists of ten individual cases. Set 1 is made up

of networks with an arc to node ratio of two (number of arcs/number of nodes = 2). In set 2 the ratio is three (arcs/nodes = 3) and in set 3 the ratio is four (arcs/nodes = 4). For each set the number of nodes ranges from ten to one hundred with the number of arcs varying based on the particular set in question.

The test cases were created using a network generating program, NETGEN, provided by Dr. Fred Glover (University of Colorado, Boulder). This program allowed for the creation of random network problems given the appropriate inputs (parameters). To maintain consistency, the only parameters varied in each case were the number of nodes and arcs. The bounds on arc capacities and costs were held constant as well as total network supply. In each case a single source and sink were used with no transshipment sources or sinks. (A set of transshipment and transportation problems were run with no significant difference in results. The results from these runs are therefore excluded from the analysis that follows.)

The OKA and NCA programs require the same data input format; therefore, a single file from NETGEN was used to run both programs.

Iterations

For the NCA an iteration is defined as one pass from Step 3 (NEWPI) through Step 7 (NEWFLOWS). For the OKA the number of iterations included the sum of nonbreakthrough and breakthrough iterations as recorded in the output from the program. To compare the results of each case run, a ratio of the number of iterations taken is formed with the number for the NCA over the number for the OKA (NCA/OKA). The results for all three sets in each category are shown in figures 4.1 and 4.2. (Refer to Appendix C for tables containing detailed results of all cases.)

Figure 4.1 reflects the results for category 1 where the arc lower bounds are held at zero. Figure 4.2 shows the results for category 2 where positive lower bounds are used. It can be seen in figure 4.2 that the addition of positive lower bounds makes the Node Chain Algorithm the more efficient method, providing the OKA program begins with zero starting flows on all arcs regardless of the arc lower bounds.

An additional observation can be made by looking at tables C-1, C-3 and C-5 in Appendix C. The number of iterations required by the NCA program, when the arc lower bounds are zero, are approximately equal to the number of nodes.

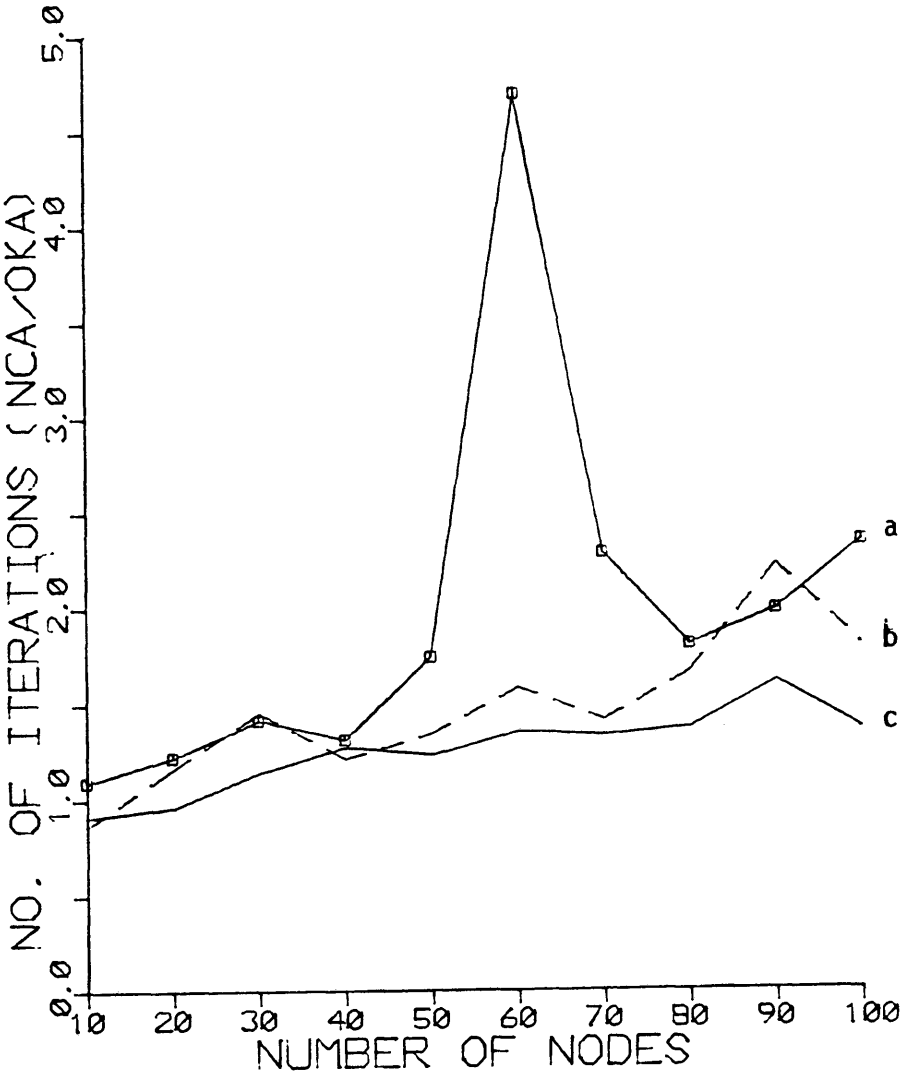

Figure 4.1: Comparison of the number of iterations when all lower bounds are zero.

(for arc to node ratios of 2 (line a), 3 (line b), and 4 (line c))

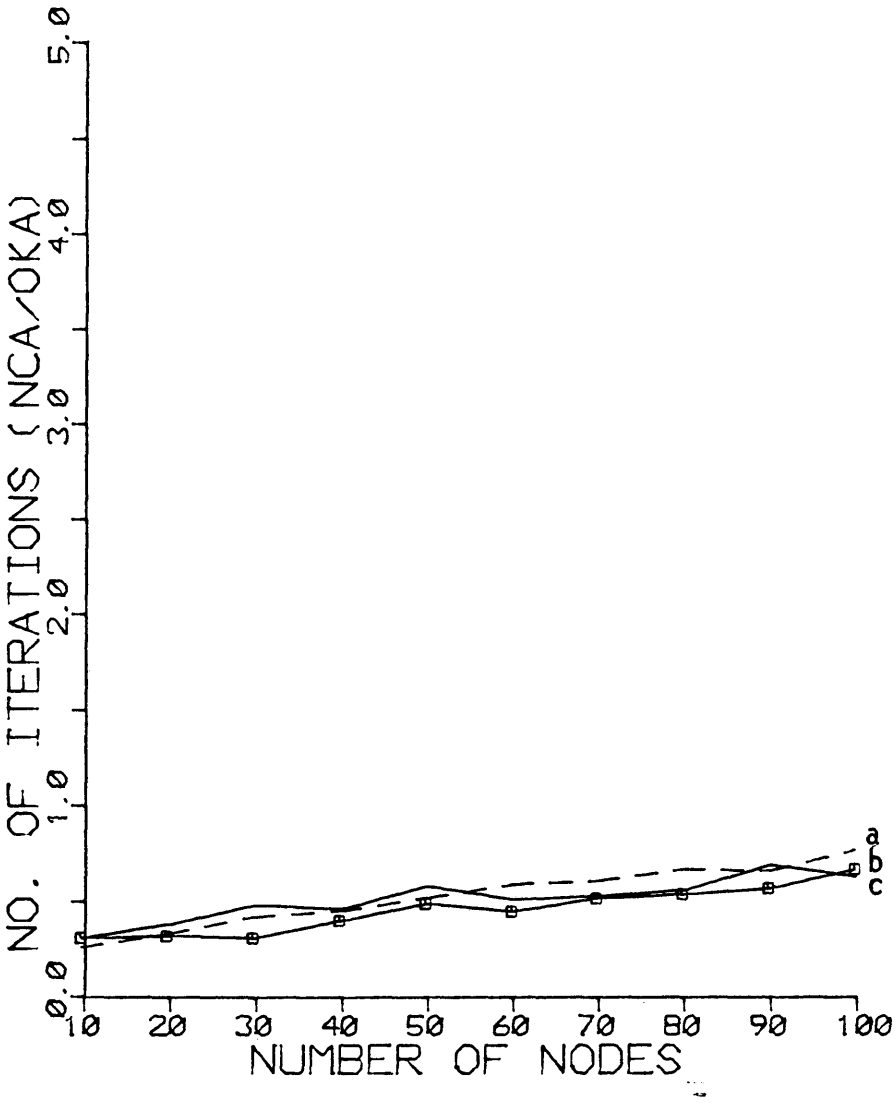

Figure 4.2: Comparison of the number of iterations when positive lower bounds exist.

(for arc to node ratios of 2 (line a), 3 (line b), and 4 (line c))

Execution Time

For each case run the CPU seconds (execution time) required for each method were recorded. A ratio of the NCA time to the OKA time was formed for each case and used as a basis of comparison. Figures 4.3 and 4.4 reflect these results for the two categories.

The results in figure 4.3 reveal that the NCA program took on the average approximately 2.3 times longer to run than the OKA program for the cases run. Looking at the positions of the three curves one can see that as the network density increases the realtive difference in execution time decreases.

The results in figure 4.4 show a marked shift downward in the curves for all three sets of case runs. The same trend found in figure 4.3 can be seen that as the network density increases the relative differences in execution times decrease. With the addition of lower bounds the NCA program took an average of 1.78 times as long to run as the OKA program, a substantial decrease.

It must be noted, in defense of the Node Chain Algorithm, that the program used is a relatively new unrefined and untested program from the standpoint of computing efficiency. The Out-of-Kilter program on the other hand has been subjected to many years of refinements in an effort to reduce execution times and costs.

4.5 SUMMARY

From an iterative standpoint the NCA appears to be a better solution method when positive lower bounds are present. The results obtained in examining execution times were not as positive in favor of the NCA but as noted previously, the NCA program is new and unrefined.

It would seem, given the necessary computational refinements program and using capacitated networks with lower bounds, the Node Chain Algorithm could be more effective solution method than Out-of-Kilter.

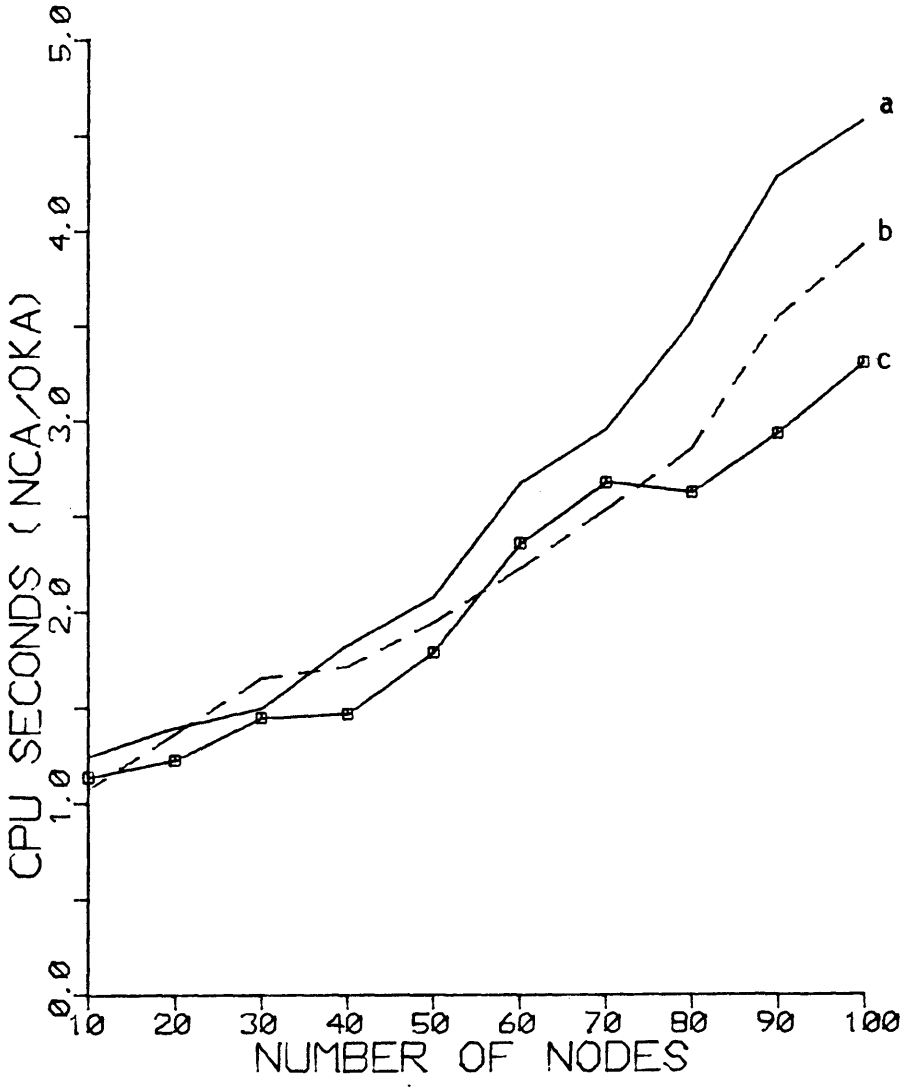

Figure 4.3: Comparison of execution times when all lower bounds are zero.

(for arc to node ratios of 2 (line a), 3 (line b), and 4 (line c))

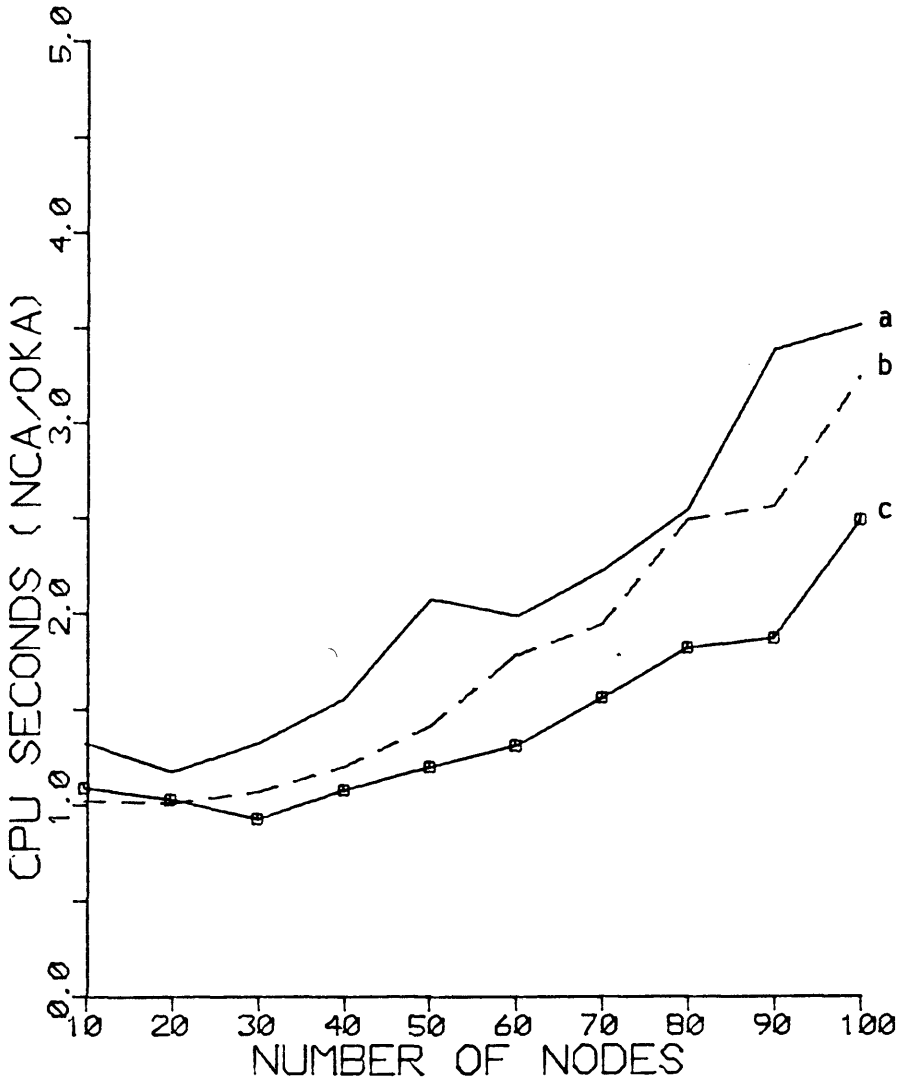

Figure 4.4: Comparison of execution times when positive lower bounds exist.

(for arc to node ratios of 2 (line a), 3 (line b), and 4 (line c))

CHAPTER 5

SUMMARY

The Node Chain Algorithm has been presented in this paper. Chapter 2 provided an explanation of the method, each step and supporting examples. From the summary provided in Figure 2.1 we can see the simplicity of the algorithm to be one of its great advantages.

In Chapter 3 the theoretical basis for the NCA algorithm was presented. It was shown that the steps of the NCA are analogous to the generalized upper bounding theory of linear programming. To show this correspondence the three examples from Chapter 2 were solved (Example 3 in Appendix B) using the GUB method. Given that the same starting basic feasible solution is established at the outset, the two methods were shown to be equivalent. As network theory has its basis in linear programming theory, this could provide a useful tool for teaching both the NCA and the GUB method in the classroom.

Chapter 4 analyzed the similarities and differences between the Out-of-Kilter Algorithm and the NCA. Both algorithms require the same network structure and will solve the same types of network problems. The NCA however is a simpler algorithm than the OKA in that it does involve the

complicated set of decision rules the OKA does.

When the efficiencies of each method are evaluated, two results are noted:

(1) As the density of the network increased, the rate of increase in execution time for the NCA was slower than for the OKA, and

(2) Based on the iterations counted per run the NCA appears to be a better solution method for problems with a high proportion of arcs with positive lower bounds.

The assertion made in (2) above is based on

(a) The computational experience gained to date.

(b) The assumption that the OKA begins with only zero starting flows on all arcs, and

(c) The belief that the average execution time for the NCA program can be markedly improved as has the OKA program over the years with the proper computational refinements.

As many minerals and petroleum related problems are ones where lower bounds prevail the NCA may prove to be useful tool to the firms of these industries. In order to improve its usefulness however, further research in the area of list processing methods should be pursued to improve the execution time of the NCA program and make it more cost effective.

SELECTED REFERENCES

Balas, Egon, 1966, "The Dual Method for the Generalized Transportation Problem," Management Science, Vol. 12, No. 7, pp. 555-568.

Charnes, A. and Cooper, W.W., 1954, "The Stepping Stone Method of Explaining Linear Programming Calculations in Transportation Problems," Management Science, No. 1, pp. 49-69.

Charnes, A. and Raike, W.M., 1965, "One-Pass Algorithms for Some Generalized Network Problems," Operations Research, Vol 14, No. 5, pp. 914-924.

Dantzig, George B., 1963, Linear Programming and Extensions, Princeton University Press, Princeton.

Ford, L.R., Jr. and Fulkerson, D.R., 1962, Flows in Networks, Princeton University Press, Princeton.

Fulkerson, D.R., 1961, "An Out-of-Kilter Method for Minimal-Cost Flow Problems," Journal for the Society of Industrial and Applied Mathematics, Vol. 9, No. 1, pp. 18-27.

Glover, F., Hultz, Jr., Klingman, D., and Stutz, J., 1978, "Generalized Networks: A Fundamental Computer Based Planning Tool," Management Science, Vol. 24, No. 12, pp. 1209-1220.

Hillier, Frederick S. and Lieberman, Gerald J., 1974, Operations Research, Second edition, Holden Day, Inc., San Francisco.

Hu, T.C., 1970, Integer Programming and Network Flows, Second printing, Addison-Wesley Publishing Co., Inc., Philippines.

Hultz, J., Klingman, D. and Russell, R., 1976, "An Advanced Dual Basic Feasible Solution for a Class of Capacitated Generalized Networks," Operations Research, Vol. 24, No. 2, pp. 301-313.

Jensen, Paul A. and Barnes, J. Wesley, 1980, Network Flow Programming, John Wiley and Sons, Inc., Canada.

Kuhn, H.W., 1955, "The Hungarian Method for the Assignment Problem," Naval Res. Logist. Quart., Vol. 2, 1955, pp. 83-97.

Lasdon, Leon S., 1970, Optimization Theory for Large Systems, Macmillan Publishing Co., Inc., New York.

Minieka, Edward, 1978, Optimization Algorithms for Networks and Graphs, Marcel Dekker, Inc., New York.

Orden, A., 1956, "The Transshipment Problem," Management Science, 2, pp. 276-285.

Phillips, Don T. and Garcia-Diaz, Alberto, 1981, Fundamentals of Network Analysis, Prentice Hall, Inc., New Jersey.

APPENDIX A

THE GENERALIZED UPPER BOUNDING TECHNIQUE

(The explanation that follows is due to material provided by Dr. C.E. Lienert, Colorado School of Mines.)

If a linear programming problem has many of its variables subject to upper bound constraints of the form

$$X_j \leqslant U_j$$

then it may be useful to use the following modification of the simplex algorithm, called the generalized upper bound (GUB) technique, in which the upper bound constraints do not appear explicitly in the tableaus.

When applying the GUB method, the entering basic variable (X_j) is chosen in the usual manner as with the simplex algorithm. The difference between the two lies in the selection of leaving basic variable.

When determining the leaving basic variable, one must calculate the "maximum feasible value of X_2" according to the following rules:

(a) For row (0), the maximum feasible value of the entering variable (X_j) is simply its upper bound.

(b) For each row in which the coefficient of X_j is positive, we form the usual ratio of b_i/a_{ij} to find out the maximum amount by which X_j can be increased without violating the nonnegativity constraint on the basic variable in row (i).

(c) For each row in which the coefficient of X_j is negative, we must determine the amount by which X_j can be increased without violating the upper bound constraint (if any) on the basic variable in this row.

To determine the leaving basic variable we select the minimum value of (a), (b) and (c) calculated above. This is the maximum feasible value the entering basic variable (X_j) may take on without violating the bounds on the current basic variables. Once the leaving variable has been determined, three cases must be considered:

Case I: The minimum value of X_j is associated with row (0) or (a) above. If so, we define a new variable Y_j as such that $X_j = U_j - Y_j$. (X_j and Y_j are called complementary decision variables.) The leaving basic variable is now Y_j. In the X_j column we substitute $(U_j - Y_j)$ in place of X_j.

This will cause a change in the signs in the X_j column and the right hand side values of the tableau only.

Case II: <u>The minimum value of X_j is associated with a row in which the coefficient of X_j is positive</u>. This is a "normal" situation in which the simplex algorithm can be applied directly and the leaving basic variable is the basic variable currently in that row.

Case III: <u>The minimum value of X_j is associated with a row in which the coefficient of X_j is negative</u>. Suppose X_i is the current basic variable in this row. We replace X_i by $U_i - Y_i$ and select Y_i as the leaving basic variable.

To illustrate the somewhat complicated decision rules discussed above consider the following example.

$$\text{MAX } Z = X_1 + 3X_2 - 2X_3,$$

Subject to:

$$\begin{aligned} X_2 - 2X_3 &\leqslant 1 \\ 2X_1 + X_2 + 2X_3 &\leqslant 8 \\ X_1 \leqslant 1,\ X_2 &\leqslant 3,\ X_3 \leqslant 2 \\ X_1,\ X_2,\ X_3 &\geqslant 0 \end{aligned}$$

The initial simplex tableau is set up in the usual manner except that the upper bound constraints are not included.

Basic Var.	Z	X_1	X_2	X_3	X_4	X_5	RHS
Z	1	-1	-3	2	0	0	0
X_4	0	0	1	-2	1	0	1
X_5	0	2	1	2	0	1	8

TABLE A-1 Original Simplex Tableau

The entering basic variable, selected in the usual manner, will be X_2. Now we must determine the maximum feasible value of X_2 according to the rules previously discussed.

Row	Max. Feasible Value of X_2	Rule
0	$X_2 \leqslant 3$	(a)
*1	$X_2 \leqslant 1$	(b)
2	$X_2 \leqslant 8$	(b)

The minimum value above corresponds to X_4 in row (1). As this is an example of Case II, we pivot on X_4 in row (1) in a standard simplex operation. The new tableau is

Basic Var.	Z	X_1	X_2	X_3	X_4	X_5	RHS
Z	1	-1	0	-4	3	0	3
X_2	0	0	1	-2	1	0	1
X_5	0	2	0	4	-1	1	7

TABLE A-2 Results of Iteration 1

The entering basic variable will now be X_3. Again we compute the maximum feasible value of the entering basic variable, now X_3.

Row	Max. Feasible Value of X_2	Rule
0	$X_3 \leq 2$	(a)
*1	$X_3 \leq 1$	(c)
2	$X_3 \leq 7/4$	(b)

The minimum value above corresponds to X_2 in row (1) (calculated by $U_2 = 1 + 2X_3$ or $3 = 1 + 2X_3$ therefore $X_3 = 1$). The coefficient of X_3 in row (1) is negative; therefore, Case III applies and Y_2 becomes the leaving basic variable.

The first step is to replace X_2 in row (1) by $(3 - Y_2)$ to get the following tableau.

Basic Var.	Z	X_1	X_2	X_3	X_4	X_5	RHS
Z	1	-1	0	-4	3	0	3
Y_2	0	0	1	2	-1	0	2
X_5	0	2	0	4	-1	1	7

TABLE A-3. Iteration 2 - $(3-Y_2)$ Substituted for X_2

The leaving basic variable is now Y_2 and the next step is to perform the usual Gauss-Jordan technique resulting in the following:

Basic Var.	Z	X_1	X_2	X_3	X_4	X_5	RHS
Z	1	-1	2	0	1	0	7
X_3	0	0	1/2	1	-1/2	0	1
X_5	0	2	-2	0	1	1	3

TABLE A-4. Results of Iteration 2.

The entering basic variable will now be X_1. Computing the maximum feasible value of X_1 we get:

Row	Max. Feasible Value of X_1	Rule
*0	$X_1 \leqslant 1$	(a)
1	--	--
2	$X_1 \leqslant 3/2$	(b)

The minimum value above corresponds to row (0) and X_1. Case I applies in this instance therefore Y_1

becomes the leaving basic variable. X_1 is replaced by $(1 - Y_1)$ in the X_1 column resulting in

Basic Var.	Z	Y_1	Y_2	X_3	X_4	X_5	RHS
Z	1	1	2	0	1	0	8
X_3	0	0	1/2	1	-1/2	0	1
X_5	0	-2	-2	0	1	1	1

TABLE A-5. Results of Iteration 3 and the Final Tableau.

No variables remain that may enter the basis; therefore, we may conclude the results in Table A-4 are optimal with the following solution values:

$$X_1 = 1,\ X_2 = 3,\ X_3 = 1 \text{ and } Z = 8$$

APPENDIX B

EXAMPLE 3 SOLUTION USING THE GUB METHOD

The network shown in Figure 2.25 (page 41) can be formulated as a linear program with each variable having it's bounds converted to a lower bound of zero and an upper bound of 1 as shown in Table B-1.

The first step is to bring the variables X_{12}, X_{13} and X_{14} in the basis as in the PREGO step. Following this the C_{ij} coefficients in row 0 for A_1 through A_4 are reduced to zero to establish the starting basic solution. These two steps are shown in Tables B-2 and B-3.

On each table, beginning with B-3 through B-11, the entering and leaving variables are enclosed in the boxes. A single vertical box indicates the entering basic variable is also the leaving basic variable as it is driven to its upper bound.

Basic Var.	Eq. No.	Coefficient of																		Right Side
		Z	x'_{12}	x'_{13}	x'_{14}	x'_{25}	x'_{26}	x'_{35}	x'_{36}	x'_{37}	x'_{46}	x'_{47}	x'_{58}	x'_{68}	x'_{78}	A_1	A_2	A_3	A_4	
Z	0	-1	35	150	120	56	18	110	135	60	60	65	40	120	35	M	M	M	M	-430
x'_{12}	1	0	5	0	0	-7	-6	0	0	0	0	0	0	0	0	0	0	0	0	0
x'_{13}	2	0	0	15	0	0	0	-10	-15	-10	0	0	0	0	0	0	0	0	0	0
x'_{14}	3	0	0	0	10	0	0	0	0	0	-6	-5	0	0	0	0	0	0	0	0
A_1	4	0	0	0	0	7	0	10	0	0	0	0	-10	0	0	1	0	0	0	2
A_2	5	0	0	0	0	0	6	0	15	0	6	0	0	-10	0	0	1	0	0	3
A_3	6	0	0	0	0	0	0	0	0	10	0	5	0	0	-5	0	0	1	0	0
A_4	7	0	0	0	0	0	0	0	0	0	0	0	10	10	5	0	0	0	1	20

TABLE B-1: Example 3 - Initial Formulation

Basic	Eq.	Coefficient of																	Right
Var.	No.	Z	X'_{12}	X'_{13}	X'_{14}	X'_{25}	X'_{26}	X'_{35}	X'_{36}	X'_{37}	X'_{46}	X'_{47}	X'_{58}	X'_{68}	X'_{78}	A_1	A_2	A_3	A_4	Side
Z	0	-1	0	0	0	105	60	210	285	160	132	125	40	120	35	M	M	M	M	-430
X'_{12}	1	0	1	0	0	-7/5	-6/5	0	0	0	0	0	0	0	0	0	0	0	0	0
X'_{13}	2	0	0	1	0	0	0	-10/15	-15/15	-10/15	0	0	0	0	0	0	0	0	0	0
X'_{14}	3	0	0	0	1	0	0	0	0	0	-6/10	-5/10	0	0	0	0	0	0	0	0
A_1	4	0	0	0	0	7	0	10	0	0	0	0	-10	0	0	1	0	0	0	2
A_2	5	0	0	0	0	0	6	0	15	0	6	0	0	-10	0	0	1	0	0	3
A_3	6	0	0	0	0	0	0	0	0	10	0	5	0	0	-5	0	0	1	0	0
A_4	7	0	0	0	0	0	0	0	0	0	0	0	10	10	5	0	0	0	1	20

TABLE B-2: Example 3 - PREGO Step (X_{12}, X_{13}, and X_{14}) are brought into the basis

Basic Var.	Eq. No.	Coefficient of																	Right Side
		z	x'_{12}	x'_{13}	x'_{14}	x'_{25}	x'_{26}	x'_{35}	x'_{36}	x'_{37}	x'_{46}	x'_{47}	x'_{58}	x'_{68}	x'_{78}	A_1	A_2	A_3	A_4	
z	0	-1	0	0	0	-7M+105	-6M+60	-10M+210	-15M+285	-10M+160	-6M+132	-5M+125	40	120	35	0	0	0	0	-25M-430
x'_{12}	1	0	1	0	0	-7/5	-6/5	0	0	0	0	0	0	0	0	0	0	0	0	0
x'_{13}	2	0	0	1	0	0	0	-10/15	-1	-10/15	0	0	0	0	0	0	0	0	0	0
x'_{14}	3	0	0	0	1	0	0	0	0	0	-6/10	-5/10	0	0	0	0	0	0	0	0
A_1	4	0	0	0	0	7	0	10	0	0	0	0	-10	0	0	1	0	0	0	2
A_2	5	0	0	0	0	0	6	0	15	0	6	0	0	-10	0	0	1	0	0	3
A_3	6	0	0	0	0	0	0	0	0	10	0	5	0	0	-5	0	0	1	0	0
A_4	7	0	0	0	0	0	0	0	0	0	0	0	10	10	5	0	0	0	1	20
$\bar{C}_{ij}$	8	-	0	0	0	M-15	M-10	M-21	M-19	M-16	M-22	M-25	4	12	7	-	-	-	-	-

TABLE B-3: Example 3 - M values for A_i's reduced to zero

Basic Var.	Eq. No.	Coefficient of																		Right Side
		Z	X'_{12}	X'_{13}	X'_{14}	X'_{25}	X'_{26}	X'_{35}	X'_{36}	X'_{37}	X'_{46}	X'_{47}	X'_{58}	X'_{68}	X'_{78}	A_1	A_2	A_3	A_4	
Z	0	-1	0	0	0	-7M+105	0	-10M+210	135	-10M+160	72	-5M+125	40	-10M+220	35	0	M-10	0	0	-22M-460
X'_{12}	1	0	1	0	0	-7/5	0	0	15/5	0	6/5	0	0	-1/5	0	0	1/5	0	0	3/5
X'_{13}	2	0	0	1	0	0	0	-10/15	-1	-10/15	0	0	0	0	0	0	0	0	0	0
X'_{14}	3	0	0	0	1	0	0	0	0	0	-6/10	-5/10	0	0	0	0	0	0	0	0
A_1	4	0	0	0	0	7	0	10	0	0	0	0	-10	0	0	1	0	0	0	2
X'_{26}	5	0	0	0	0	0	1	0	15/6	0	1	0	0	-10/6	0	0	1/6	0	0	3/6
A_3	6	0	0	0	0	0	0	0	0	10	0	5	0	0	-5	0	0	1	0	0
A_4	7	0	0	0	0	0	0	0	0	0	0	0	10	10	5	0	0	0	1	20
$\bar{C}_{ij}$	8	-	0	0	0	M-15	0	M-21	9	M-16	12	M-25	4	M-22	7	-	-	-	-	-

TABLE B-4: Example 3 - Results of Iteration 1

Basic Var.	Eq. No.	Coefficient of																		Right Side
		Z	Y_{12}	X'_{13}	X'_{14}	X'_{25}	X'_{26}	X'_{35}	X'_{36}	X'_{37}	X'_{46}	X'_{47}	X'_{58}	X'_{68}	X'_{78}	A_1	A_2	A_3	A_4	
Z	0	-1	0	0	0	0	0	60	135	-10M+160	72	-5M+125	-10M+190	-10M+220	35	M-15	M-10	0	0	-20M-490
Y_{12}	1	0	1	0	0	0	0	-10/5	-15/5	0	-6/5	0	10/5	10/5	0	-1/5	-1/5	0	0	0
X'_{13}	2	0	0	1	0	0	0	-10/15	-1	-10/15	0	0	0	0	0	0	0	0	0	0
X'_{14}	3	0	0	0	1	0	0	0	0	0	-6/10	-5/10	0	0	0	0	0	0	0	0
X'_{25}	4	0	0	0	0	1	0	10/7	0	0	0	0	-10/7	0	0	1/7	0	0	0	2/7
X'_{26}	5	0	0	0	0	0	1	0	15/6	0	1	0	0	-10/6	0	0	1/6	0	0	3/6
A_3	6	0	0	0	0	0	0	0	0	10	0	5	0	0	-5	0	0	1	0	0
A_4	7	0	0	0	0	0	0	0	0	0	0	0	10	10	5	0	0	0	1	20
$\bar{C}_{ij}$	8	-	0	0	0	0	0	6	9	M-16	12	M-25	M-19	M-22	7	-	-	-	-	-

TABLE B-5: Example 3 - Results of Iteration 2

Basic Var.	Eq. No.	Coefficient of																		Right Side
		z	Y_{12}	X'_{13}	X'_{14}	X'_{25}	X'_{26}	X'_{35}	X'_{36}	X'_{37}	X'_{46}	X'_{47}	X'_{58}	X'_{68}	X'_{78}	A_1	A_2	A_3	A_4	
z	0	-1	0	0	0	0	0	60	135	0	72	45	-10M+190	-10M+220	-5M+115	M-15	M-10	M-16	0	-20M-490
Y_{12}	1	0	1	0	0	0	0	-10/5	-15/5	0	-6/5	0	10/5	10/5	0	-1/5	-1/5	0	0	0
X'_{13}	2	0	0	1	0	0	0	-10/15	-1	0	0	5/15	0	0	-5/15	0	0	1/15	0	0
X'_{14}	3	0	0	0	1	0	0	0	0	0	-6/10	-5/10	0	0	0	0	0	0	0	0
X'_{25}	4	0	0	0	0	1	0	10/7	0	0	0	0	-10/7	0	0	1/7	0	0	0	2/7
X'_{26}	5	0	0	0	0	0	1	0	15/6	0	1	0	0	-10/6	0	0	1/6	0	0	3/6
X'_{37}	6	0	0	0	0	0	0	0	0	1	0	5/10	0	0	-5/10	0	0	1/10	0	0
A_4	7	0	0	0	0	0	0	0	0	0	0	0	10	10	5	0	0	0	1	20
$\bar{c}_{ij}$	8	-	0	0	0	0	0	6	9	0	12	9	M-19	M-22	M-23	-	-	-	-	-

TABLE B-6: Example 3 - Results of Iteration 3

Basic Var.	Eq. No.	Coefficient of																	Right Side
		z	Y_{12}	X'_{13}	X'_{14}	X'_{25}	X'_{26}	X'_{35}	X'_{36}	X'_{37}	X'_{46}	X'_{47}	X'_{58}	X'_{68}	X'_{78}	A_1	A_2	A_3	A_4	
z	0	-1	5M-95	0	0	0	0	-10M+250	-15M+420	0	-6M+186	45	0	30	-5M+115	4	9	M-16	0	-20M-490
X'_{58}	1	0	5/10	0	0	0	0	-1	-15/10	0	-6/10	0	1	1	0	-1/10	-1/10	0	0	0
X'_{13}	2	0	0	1	0	0	0	-10/15	-1	0	0	5/15	0	0	-5/15	0	0	1/15	0	0
X'_{14}	3	0	0	0	1	0	0	0	0	0	-6/10	-5/10	0	0	0	0	0	0	0	0
X'_{25}	4	0	0	0	0	1	0	0	-15/7	0	-6/7	0	0	10/7	0	0	-1/7	0	0	2/7
X'_{26}	5	0	0	0	0	0	1	0	15/6	0	1	0	0	-10/6	0	0	1/6	0	0	3/6
X'_{37}	6	0	0	0	0	0	0	0	0	1	0	5/10	0	0	-5/10	0	0	1/10	0	0
A_4	7	0	-5	0	0	0	0	10	15	0	6	0	0	0	5	1	1	0	1	20
$\bar{C}_{ij}$	8	-	M-19	0	0	0	0	M-25	M-28	0	M-31	9	0	3	M-23	-	-	-	-	-

TABLE B-7: Example 3 - Results of Iteration 4

Basic Var.	Eq. No.	Coefficient of																	Right Side
		z	y_{12}	x'_{13}	x'_{14}	x'_{25}	x'_{26}	x'_{35}	x'_{36}	x'_{37}	x'_{46}	x'_{47}	x'_{58}	x'_{68}	y_{78}	A_1	A_2	A_3	A_4	
z	0	-1	5M-95	0	0	0	0	-10M+250	-15M+420	0	-6M+186	45	0	30	5M+115	4	9	M-16	0	-15M-605
x'_{58}	1	0	5/10	0	0	0	0	-1	-15/10	0	-6/10	0	1	1	0	-1/10	-1/10	0	0	0
x'_{13}	2	0	0	1	0	0	0	-10/15	-1	0	0	5/15	0	0	5/15	0	0	1/15	0	5/15
x'_{14}	3	0	0	0	1	0	0	0	0	0	-6/10	-5/10	0	0	0	0	0	0	0	0
x'_{25}	4	0	0	0	0	1	0	0	-15/7	0	-6/7	0	0	10/7	0	0	-1/7	0	0	2/7
x'_{26}	5	0	0	0	0	0	1	0	15/6	0	1	0	0	-10/6	0	0	1/6	0	0	3/6
x'_{37}	6	0	0	0	0	0	0	0	0	1	0	5/10	0	0	5/10	0	0	1/10	0	5/10
A_4	7	0	-5	0	0	0	0	10	15	0	6	0	0	0	-5	1	1	0	1	15
$\bar{c}_{ij}$	8	-	M-19	0	0	0	0	M-25	M-28	0	M-31	-9	0	-3	M-23	-	-	-	-	-

TABLE B-8: Example 3 - Results of Iteration 5

Basic Var.	Eq. No.	Coefficient of																	Right Side
		Z	Y_{12}	Y_{13}	X'_{14}	X'_{25}	X'_{26}	Y_{35}	X'_{36}	X'_{37}	X'_{46}	X'_{47}	Y_{58}	X'_{68}	Y_{78}	A_1	A_2	A_3	A_4	
Z	0	-1	5M-95	0	0	0	0	10M-250	-15M+420	0	-6M+186	45	0	30	-5M+115	4	9	M-16	0	-5M-855
Y_{58}	1	0	-5/10	0	0	0	0	-1	15/10	0	6/10	0	1	-1	0	1/10	1/10	0	0	0
Y_{13}	2	0	0	1	0	0	0	-10/15	1	0	0	-5/15	0	0	-5/15	0	0	-1/15	0	0
X'_{14}	3	0	0	0	1	0	0	0	0	0	-6/10	-5/10	0	0	0	0	0	0	0	0
X'_{25}	4	0	0	0	0	1	0	0	-15/7	0	-6/7	0	0	10/7	0	0	-1/7	0	0	2/7
X'_{26}	5	0	0	0	0	0	1	0	15/6	0	1	0	0	-10/6	0	0	1/6	0	0	3/6
X'_{37}	6	0	0	0	0	0	0	0	0	1	0	5/10	0	0	5/10	0	0	1/10	0	5/10
A_4	7	0	-5	0	0	0	0	-10	15	0	6	0	0	0	-5	1	1	0	1	15
$\bar{C}_{ij}$	8	-	M-19	0	0	0	0	M-25	M-28	0	M-31	9	0	3	M-23	-	-	-	-	-

TABLE B-9: Example 3 - Results of Iteration 6

Basic Var.	Eq. No.	Coefficient of																	Right Side
		z	Y_{12}	Y_{13}	X'_{14}	X'_{25}	X'_{26}	Y_{35}	X'_{36}	X'_{37}	X'_{46}	X'_{47}	Y_{58}	X'_{68}	Y_{78}	A_1	A_2	A_3	A_4	
z	0	-1	45	0	0	0	0	30	0	0	18	45	10M-280	-10M+310	5M-115	M-24	M-19	M-16	0	-5M-855
X'_{36}	1	0	-5/15	0	0	0	0	-10/15	1	0	6/15	0	10/15	-10/15	0	1/15	1/15	0	0	0
Y_{13}	2	0	5/15	1	0	0	0	0	0	0	-6/15	-5/15	-10/15	-10/15	-5/15	-1/15	-1/15	-1/15	0	0
X'_{14}	3	0	0	0	1	0	0	0	0	0	-6/10	-5/10	0	0	0	0	0	0	0	0
X'_{25}	4	0	-5/7	0	0	1	0	-10/7	0	0	0	0	10/7	0	0	1/7	0	0	0	2/7
X'_{26}	5	0	5/6	0	0	0	1	10/6	0	0	0	0	10/6	0	0	-1/6	0	0	0	3/6
X'_{37}	6	0	0	0	0	0	0	0	0	1	0	5/10	0	0	5/10	0	0	1/10	0	5/10
A_4	7	0	0	0	0	0	0	0	0	0	0	0	-10	10	-5	0	0	0	1	5
$\bar{c}_{ij}$	8	-	9	0	0	0	0	3	0	0	3	9	M-28	M-31	M-23	-	-	-	-	-

TABLE B-10: Example 3 - Results of Iteration 7

Basic Var.	Eq. No.	Coefficient of																	Right Side
		Z	Y_{12}	Y_{13}	X'_{14}	X'_{25}	X'_{26}	Y_{35}	X'_{36}	X'_{37}	X'_{46}	X'_{47}	Y_{58}	X'_{68}	Y_{78}	A_1	A_2	A_3	A_4	
Z	0	-1	5M-110	15M-465	0	0	0	30	0	0	-6M+204	-5M+200	30	0	40	7	12	15	0	-5M-855
X'_{36}	1	0	0	1	0	0	0	-10/15	1	0	0	-5/15	0	0	-5/15	0	0	-1/15	0	0
X'_{68}	2	0	5/10	15/10	0	0	0	0	0	0	-6/10	-5/10	-1	1	-5/10	-1/10	-1/10	-1/10	0	0
X'_{14}	3	0	0	0	1	0	0	0	0	0	-6/10	-5/10	0	0	0	0	0	0	0	0
X'_{25}	4	0	-5/7	0	0	1	0	-10/7	0	0	0	0	10/7	0	0	1/7	0	0	0	2/7
X'_{26}	5	0	5/6	0	0	0	1	10/6	0	0	0	0	10/6	0	0	-1/6	0	0	0	3/6
X'_{37}	6	0	0	0	0	0	0	0	0	1	0	5/10	0	0	5/10	0	0	1/10	0	5/10
A_4	7	0	-5	-15	0	0	0	0	0	0	6	5	0	0	0	1	1	1	1	5
$\bar{C}_{ij}$	8	-	M-22	M-31	0	0	0	3	0	0	M-34	M-40	3	0	8	-	-	-	-	-

TABLE B-11: Example 3 - Results of Iteration 8

Basic Var.	Eq. No.	Coefficient of																		Right Side
		Z	Y_{12}	Y_{13}	X'_{14}	X'_{25}	X'_{26}	Y_{35}	X'_{36}	X'_{37}	X'_{46}	X'_{47}	Y_{58}	X'_{68}	Y_{78}	A_1	A_2	A_3	A_4	
Z	0	-1	60	45	0	0	0	30	0	0	0	30	30	0	40	M-27	M-22	M-10	M-34	-1025
X'_{36}	1	0	0	1	0	0	0	-10/15	1	0	0	-5/15	0	0	-5/15	0	0	-1/15	0	0
X'_{68}	2	0	0	0	0	0	0	0	0	0	0	0	-1	1	-5/10	0	0	0	1/10	5/10
X'_{14}	3	0	-5/10	-15/10	1	0	0	0	0	0	0	0	0	0	0	1/10	1/10	1/10	1/10	5/10
X'_{25}	4	0	-5/7	0	0	1	0	-10/7	0	0	0	0	10/7	0	0	1/7	0	0	0	2/7
X'_{26}	5	0	5/6	0	0	0	1	10/6	0	0	0	0	10/6	0	0	-1/6	0	0	0	3/6
X'_{37}	6	0	0	0	0	0	0	0	0	1	0	5/10	0	0	5/10	0	0	1/10	0	5/10
X_{46}	7	0	-5/6	-15/6	0	0	0	0	0	0	1	5/6	0	0	0	1/6	1/6	1/6	1/6	5/6

TABLE B-12: Example 3 - Results of Iteration 9 and the Final tableau

Looking at the results shown in B-12, we see the following:

$X'_{12} = 1$	or	$X_{12} = 10$
$X'_{13} = 1$	or	$X_{13} = 20$
$X'_{14} = 5/10$	or	$X_{14} = 10$
$X'_{25} = 2/7$	or	$X_{25} = 5$
$X'_{26} = 3/6$	or	$X_{26} = 5$
$X'_{35} = 1$	or	$X_{35} = 15$
$X'_{36} = 0$	or	$X_{36} = 0$
$X'_{37} = 5/10$	or	$X_{37} = 5$
$X'_{46} = 5/6$	or	$X_{46} = 5$
$X'_{47} = 0$	or	$X_{47} = 5$
$X'_{58} = 1$	or	$X_{58} = 20$
$X'_{68} = 5/10$	or	$X_{68} = 10$
$X'_{78} = 1$	or	$X_{78} = 10$

with $Z = 1025$

APPENDIX C

Computational Results of Test Cases Run Comparing the NCA to the OKA Programs

Category 1 Set 1 (Arcs/Nodes = 2)

Case	No. of Nodes	No. of Arcs	NCA No. of Iter.	NCA CPU Secs.	OKA No. of* Iter.	OKA CPU Secs.	NCA/OKA No. of Iter.	NCA/OKA CPU Secs.
1	10	20	11	.25	12	.20	.91	1.25
2	20	40	24	.35	25	.25	.96	1.40
3	30	60	32	.48	28	.32	1.14	1.50
4	40	80	42	.66	33	.36	1.27	1.83
5	50	100	57	.94	46	.45	1.23	2.08
6	60	120	73	1.34	54	.50	1.35	2.68
7	70	140	72	1.60	54	.54	1.33	2.96
8	80	160	89	2.23	65	.63	1.37	3.53
9	90	180	103	2.96	64	.69	1.61	4.29
10	100	200	103	3.48	76	.76	1.36	4.58

* Primarily nonbreakthrough iterations

TABLE C-1
Results for Set 1 with arc to node ratio = 2
and lower bounds = 0

Category 2 Set 1 (Arcs/Nodes = 2)

Case	No. of Nodes	No. of Arcs	NCA		OKA		NCA/OKA	
			No. of Iter.	CPU Secs.	No. of Iter.	CPU Secs.	No. of Iter.	CPU Secs.
1	10	20	10	.24	32	.18	.31	1.33
2	20	40	22	.33	57	.28	.38	1.18
3	30	60	36	.52	75	.39	.48	1.33
4	40	80	46	.72	99	.50	.46	1.56
5	50	100	74	1.21	127	.68	.58	2.08
6	60	120	78	1.55	153	.78	.51	1.99
7	70	140	92	2.17	171	.97	.53	2.23
8	80	160	104	2.84	185	1.11	.56	2.55
9	90	180	143	4.45	207	1.31	.69	3.39
10	100	200	141	5.33	224	1.48	.63	3.52

TABLE C-2
Results for Set 1 with the arc to node ratio = 2
and positive lower bounds

Category 1 Set 2 (Arcs/Nodes = 3)

Case	No. of Nodes	No. of Arcs	NCA No. of Iter.	NCA CPU Secs.	OKA No. of* Iter.	OKA CPU Secs.	NCA/OKA No. of Iter.	NCA/OKA CPU Secs.
1	10	30	12	.28	14	.26	.86	1.08
2	20	60	22	.41	19	.30	1.16	1.37
3	30	90	32	.58	22	.35	1.45	1.66
4	40	120	46	.81	38	.47	1.21	1.72
5	50	150	55	1.09	41	.56	1.34	1.95
6	60	180	65	1.47	41	.66	1.58	2.23
7	70	210	75	1.93	53	.76	1.41	2.54
8	80	240	83	2.43	50	.85	1.66	2.86
9	90	270	91	3.05	41	.86	2.22	3.55
10	100	300	108	4.05	60	1.03	1.80	3.93

* Primarily nonbreakthrough iterations

TABLE C-3
Results for Set 2 with the arc to node ratio = 3
and lower bounds = 0

Category 2 Set 2 (Arcs/Nodes = 3)

Case	No. of Nodes	No. of Arcs	NCA		OKA		NCA/OKA	
			No. of Iter.	CPU Secs.	No. of Iter.	CPU Secs.	No. of Iter.	CPU Secs.
1	10	30	10	.30	39	.29	.26	1.03
2	20	60	22	.43	66	.42	.33	1.02
3	30	90	44	.68	104	.63	.42	1.08
4	40	120	65	1.02	146	.84	.45	1.21
5	50	150	81	1.46	156	1.03	.52	1.42
6	60	180	109	2.29	183	1.28	.59	1.79
7	70	210	131	3.26	215	1.67	.61	1.95
8	80	240	149	4.43	221	1.77	.67	2.50
9	90	270	165	5.42	251	2.11	.66	2.57
10	100	300	192	7.54	247	2.32	.77	3.25

TABLE C-4
Results for Set 2 with the arc to node ratio = 3
and positive lower bounds

Category 1 Set 3 (Arcs/Nodes = 4)

Case	No. of Nodes	No. of Arcs	NCA		OKA		NCA/OKA	
			No. of Iter.	CPU Secs.	No. of* Iter.	CPU Secs.	No. of Iter.	CPU Secs.
1	10	40	12	.33	11	.29	1.09	1.14
2	20	80	22	.49	18	.40	1.22	1.23
3	30	120	31	.71	22	.49	1.41	1.45
4	40	160	46	1.00	35	.68	1.31	1.47
5	50	200	54	1.34	31	.79	1.74	1.79
6	60	240	61	1.72	13	.73	4.5	2.36
7	70	280	89	2.65	39	.99	2.28	2.68
8	80	320	83	2.92	46	1.11	1.80	2.63
9	90	360	93	3.62	47	1.23	1.98	2.94
10	100	400	103	4.63	44	1.40	2.34	3.31

* Primarily nonbreakthrough iterations

TABLE C-5
Results for Set 3 with the arc to node ratio = 4
and lower bounds = 0

Category 2 Set 3 (Arcs/Nodes = 4)

Case	No. of Nodes	No. of Arcs	NCA		OKA		NCA/OKA	
			No. of Iter.	CPU Secs.	No. of Iter.	CPU Secs.	No. of Iter.	CPU Secs.
1	10	40	12	.35	39	.32	.31	1.10
2	20	80	29	.53	92	.51	.32	1.04
3	30	120	43	.79	138	.84	.31	.94
4	40	160	71	1.24	177	1.14	.40	1.09
5	50	200	94	1.84	191	1.50	.49	1.21
6	60	240	103	2.42	230	1.83	.45	1.32
7	70	280	127	3.48	244	2.21	.52	1.57
8	80	320	156	4.90	287	2.67	.54	1.83
9	90	360	172	6.12	302	3.25	.57	1.88
10	100	400	224	9.02	336	3.60	.67	2.50

TABLE C-6
Results for Set 3 with the arc to node ratio = 4
and positive lower bounds