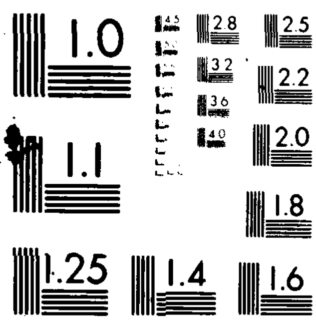
AD-A110 492 SOUTHERN METHODIST UNIV DALLAS TX DEPT OF OPERATIONS--ETC F/G 15/5
AN ANALYSIS OF THE LOGAIR DISTRIBUTION SYSTEM USING OPTIMIZATIO--ETC(U)
NOV 81 J L KENNINGTON AFOSR-77-3151

UNCLASSIFIED AFOSR-TR-82-0029 NL

END
DATE
FILMED
3 82
DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

AFOSR-TR- 82-0029

LEVEL II

(12)

AD A110492

FINAL SCIENTIFIC REPORT

Prepared For

DEPARTMENT OF THE AIR FORCE

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

BOLLING AIR FORCE BASE, DC 20332

GRANT AFOSR 77-3151

DTIC ELECTE FEB 4 1982 S H

AN ANALYSIS OF THE LOGAIR DISTRIBUTION SYSTEM USING OPTIMIZATION PRINCIPLES

Jeff L. Kennington
Department of Operations Research
Southern Methodist University
Dallas, Texas 75275

November 1981

DTIC FILE COPY

82 03 02 126

Approved for public release;
distribution unlimited.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR- 82-0029	2. GOVT ACCESSION NO. AD-A110 492	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AN ANALYSIS OF THE LOGAIR DISTRIBUTION SYSTEM USING OPTIMIZATION PRINCIPLES		5. TYPE OF REPORT & PERIOD COVERED FINAL, 1 OCT 76-30 SEP 81
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Jeff L. Kennington		8. CONTRACT OR GRANT NUMBER(s) AFOSR-77-3151
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Operations Research Southern Methodist University Dallas TX 75275		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F; 2304/A6
11. CONTROLLING OFFICE NAME AND ADDRESS Mathematical & Information Sciences Directorate Air Force Office of Scientific Research Bolling AFB DC 20332		12. REPORT DATE NOV 81
		13. NUMBER OF PAGES 40
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

82 03 02 126

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report presents a summary of the research sponsored under the grant. The major objective of ~~the~~ research was to build mathematical programming models and specialized software to assist Air Force personnel at Wright-Patterson AFB in the design of the LOGAIR Distribution System. A description of the problem, the mathematical models developed, and the software developed is presented. The software has been documented and installed at Wright-Patterson and is currently being used by Air Force Logistics Command personnel. (CONTINUED)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ITEM #20, CONT.:

A secondary objective of the principal investigator was to address the issue of how one can automatically convert general linear programs into network models. The underlying mathematical results which can be used to develop either exact transformation algorithms or heuristic transformation algorithms are presented in Section III. This section concludes with a heuristic algorithm which, it is believed, holds the best hope for routinely converting linear programs into network programs or network programs with extra constraints.

I. INTRODUCTION

This report presents a summary of the research sponsored by AFOSR under Grant Number 77-3151. This research was performed during the years 1976 through 1981 by the principal investigator and a group of graduate students.

The major objective of the research was to build mathematical programming models and specialized software to assist Air Force personnel at Wright-Patterson AFB in the design of the LOGAIR Distribution System. A description of the problem, the mathematical models developed, and the software developed is presented in Section II. The software has been documented and installed at Wright-Patterson and is currently being used by Air Force Logistics Command personnel.

A secondary objective of the principal investigator was to address the issue of how one can automatically convert general linear programs into network models. The underlying mathematical results which can be used to develop either exact transformation algorithms or heuristic transformation algorithms are presented in Section III. This section concludes with a heuristic algorithm which, I believe, holds the best hope for routinely converting linear programs into network programs or network programs with extra constraints.

The list of technical reports which have been either fully or partially supported by this grant are as follows:

1. "Primal Simplex Network Codes: State-Of-The-Art Implementation Technology," (Ali, Helgason, Kennington, Lall), Networks 8(4) (1978), 315-346.

2. "Nonextreme Point Solution Strategies For Linear Programs," (Cooper and Kennington), Naval Research Logistics Quarterly, 26(3) (1979), 447-461.

3. "Computational Comparison Among Three Multicommodity Network Flow Algorithms," (Ali, Helgason, Kennington, Lall), Operations Research 28(4) (1980), 995-1000.

4. "A Polynomially Bounded Algorithm for a Singly Constrained Quadratic Program," (Helgason, Kennington, Lall) Mathematical Programming 18(3) (1980), 338-343.

5. "The Convex Cost Network Flow Problem: A Survey of Algorithms," (Ali, Helgason, Kennington).

6. "Spike Swapping in Basis Reinversion," (Helgason, Kennington), Naval Research Logistics Quarterly, 27(4) (1980), 697-701.

7. "Splitting the Bump in an Elimination Factorization," (Helgason, Kennington), (to appear in Naval Research Logistics Quarterly).

8. Algorithms for Network Programming, (Helgason, Kennington) John Wiley, (1980).

9. "MNETGN Program Documentation," (Ali, Kennington).

10. "A Primal Partitioning Code for Solving Multicommodity Network Flow Problems (Version 1)," (Kennington).

11. "Using Multicommodity Network Models for the Air Force Logistics Command," (with A. Ali and R. Helgason).

12. "A Lagrangean Relaxation Approach to the Generalized Fixed Charge Multicommodity Minimal Cost Network Flow Problem," (unpublished dissertation by R. Helgason).

13. "Two Node-Routing Problems," (unpublished dissertation by I. Ali).

14. "The M-Travelling Salesmen Problem," (with I. Ali).

15. "The M-Travelling Salesmen Problem: A Duality Based Branch-And-Bound Algorithm," (with I. Ali).

16. "An Application of Mathematical Programming to Assess Productivity in the Houston Independent School District," (with A. Bessent, W. Bessent, and B. Reagan).

17. "Computational Comparison of General Purpose Versus Specialized Software for Multicommodity Network Programs", (with B. Patty and B. Shetty).

II. THE LOGAIR DISTRIBUTION SYSTEM

This section reports on the successful application of mathematical programming in a decision support system for the Air Force Logistics Command. A complementary pair of multicommodity network flow models is used to aid Air Force personnel in designing the air cargo network and shipment plan (LOGAIR) utilized by the Air Force to support sixty bases in the continental U. S. A. State-of-the-art software was developed to solve these models and this software is currently being used at Wright-Patterson AFB in an integrated man-machine system to aid Air Force personnel in making annual design changes in the route structure.

2.1 Problem Description

The U. S. Air Force has major repair facilities which are responsible for the maintenance of serviceable spares for all aircraft, missiles, and ground radar systems. When a subsystem fails, it is removed and replaced by an operating subsystem. The failed system is shipped to the repair facility, repaired, and returned to either the base of origin or to inventory. During 1980, for example, over 2700 tons of serviceable spares were shipped from Wright-Patterson AFB in Ohio to Tinker AFB in Oklahoma and for the entire system of 60 bases, well over 100,000 tons of cargo was moved. Due to the magnitude of these shipping requirements, the Air Force maintains a separate air cargo system for shipment of these serviceable spares. Each year Air Force Logistics Command (AFLC) personnel develop a daily air cargo shipment plan to be used for the entire fiscal year. This section reports on a complementary pair of multicommodity network flow models used to aid Air Force personnel in designing the air cargo network and shipment plan.

2.2 Survey of Literature

Due to numerous applications, *routing and scheduling* problems have been extensively studied in the operations research literature. Invariably, simplifying assumptions are made to specialize a problem for a given situation (e.g. [19, 20, 41]). Many characteristics of the Air Force cargo shipment plan design problem are also present in the areas of bus, train, and ship routing.

In school bus routing problems one is concerned with routing in a single period and with only a single destination. Problems in this class are almost always approached with a heuristic method based on a modification of the nearest unvisited city procedure developed for the traveling salesman problem (e.g. [6, 48, 56]).

Silman, Barzily, and Passy [50] present heuristic procedures for developing schedules for city buses. They propose a two phase approach for devising bus routes and schedules. Phase 1 obtains a set of potential routes while the second phase gives the frequency of travel. Their general approach is adaptable to the Air Force problem but their specific heuristic is specialized for only routing buses. Billheimer and Gray [12] address the general fixed-charge multicommodity network flow problem in the context of Mass Transportation Network Design. However, they assume that all arcs have infinite capacity which greatly simplifies the solution procedure.

Ferguson and Dantzig [18] present a model for assigning aircraft to routes. However, they assume the routes given and ignore all fixed charges. Bellmore, Bennington, and Lubore [9] present a model for assigning tankers to shipping routes to maximize a utility function. They view the tankers as the commodities and assume a possible loading after the tankers have been assigned to routes. Again the routes are assumed given and there are no fixed charges incurred for using a shipping lane. A similar study on the movement of train cars over a rail system was conducted by White and Wrathall [58]. Unlike the Air Force problem, the network topology and schedules are input for their system. Geoffrion and Graves [21] solved a large warehouse location distribution problem for Hunt-Wesson Foods, Inc. and; Marsten and Muller [45] solved some special models for the Flying Tiger Line, but these models are not applicable for the Air Force problem.

Demmy and Brant [15], in an early paper, were the first to model the Air Force problem. Their model was a large linear program with GUB constraints. Agin and Cullen [1] present a model for the general vehicle routing and scheduling problem, and Richardson [49] presents a routing model for commercial airline schedule planning. Unfortunately, these models when applied to the Air Force problem produce mixed integer programs for which there is little hope of finding an efficient solution procedure.

2.3 The General Decision Support System

The AFLC defines a *cargo-route* as a sequence of bases and an aircraft type such that the first base and the last base in the sequence are the same. This guarantees that both the aircraft and crew are returned to the home base. Two major types of aircraft, the Lockheed L100 and L188, are currently being used in the air cargo system. The characteristics of these aircraft are given in Table 1. The sequence of bases {Tinker, Hill, Travis, Robbins, Tinker} along with the Lockheed L100 is a cargo-route in which the fixed costs (cost for flying the route with an empty L100), the variable cost (fuel consumption cost as a function of cargo weight for the L100), and cargo capacity are known. A set of cargo-routes for the 60 base system is called an *air cargo plan*.

TABLE 1. AIRCRAFT CHARACTERISTICS (1980 DATA)

Aircraft Characteristics	Aircraft	
	L100	L188
Transportation Cost ($/mile)	3.5260	2.4128
Fuel Consumption Cost ($/mile)	0.9936	1.0304
Empty Weight (lbs)	74,746	56,013
Full Weight (lbs)	120,746	86,538
Usable Cargo Capacity (lbs)	43,160	28,640
Total Cargo Capacity (Cargo plus pallets) (lbs)	46,000	30,525
Variable Component of Cost ($/mile) $\frac{F \cdot B}{E \cdot D}$	0.87703×10^{-5}	1.2691×10^{-5}
Fixed Component of Cost ($/mile) $A - \frac{F \cdot B}{D}$	3.1475	2.0493

Based on forecasted cargo shipment data, each year AFLC personnel develop an air cargo plan to be flown on a daily basis for the next fiscal year. The routes in the plan are flown by civilian carriers and are not subject to change during the fiscal year. The objective of AFLC personnel is to select the least cost set of cargo-routes which satisfy the point-to-point demands for cargo movement among 60 Air Force Bases.

Following the work of Agin and Cullen [1] a global optimization model of this planning problem can be developed. This global model is a mixed integer program with over 3 million continuous variables and over 60 thousand binary variables. In contrast to the above approach, we have chosen to develop an integrated man-machine system which may be used in the development of an air cargo plan. The three inputs of this system are as follows:

(i) aircraft characteristics (Table 1),

(ii) a 60 by 60 cargo forecast matrix, and

(iii) a 60 by 60 distance matrix which gives the flight distance between all pairs of bases.

Using only the cargo forecast matrix and the distance matrix as inputs, a nominal set of cargo-routes is produced. The nominal set is selected in such a way that the system pound-miles is a minimum. These routes use hypothetical aircraft and may violate constraints on the length of time a crew travels before it returns to the home base.

Using the aircraft available and taking into consideration other system constraints, Air Force personnel modify the nominal cargo-routes to form a set of potential routes. An integer programming problem is then solved to develop an air cargo plan from the set of potential routes. The analytical tools, input data, and the man-machine interaction is illustrated in Figure 1.

2.4 The Flow Generator and Route Selector Models

We now present the mathematical notation used to define a cargo-route. A *network* $G = [N, A]$ consists of a node set N and a set of ordered pairs of nodes $A = \{e_1, \ldots, e_t\}$. A *circuit* is defined to be a finite sequence of the form $\{s_1, (s_1,s_2), s_2, (s_2,s_3), \ldots, s_m, (s_m,s_1), s_1\}$ where $s_i \in N$ and each pair $(s_i,s_j) \in A$. A circuit along with an aircraft type specifies a cargo-route.

Let A denote the node-arc incidence matrix for a network and let C denote the set of arcs in some circuit in the network. Let $\underline{y}$ be any vector such that $A\underline{y} = \underline{0}$. Such a vector has been referred to as a *flow* by Berge and Ghouila-Houri [11]. Let

$$z_j = \begin{cases} 1, \text{ if the jth arc is a member of C,} \\ 0, \text{ otherwise.} \end{cases}$$

Then the vector $\underline{z}$ is a flow and will be referred to as a *vector-circuit* corresponding to C.

For the Air Force problem, we use a linear program to obtain a vector $\underline{y}$ satisfying $A\underline{y} = \underline{0}$ and $\underline{y} \geq \underline{0}$. We then apply a simple labeling algorithm to decompose $\underline{y}$ into a set of vector-circuits and nonnegative multipliers such that $\underline{y} = \sum_{i=j}^{p} \alpha_i \underline{z}^i$. The problem of finding a basis of cycles in a graph has been extensively studied in the literature (see [2, 10, 46 47, 56]). The $\underline{z}^i$ may be viewed as a set of nominal cargo-routes each with aircraft having capacity of at least α_i.

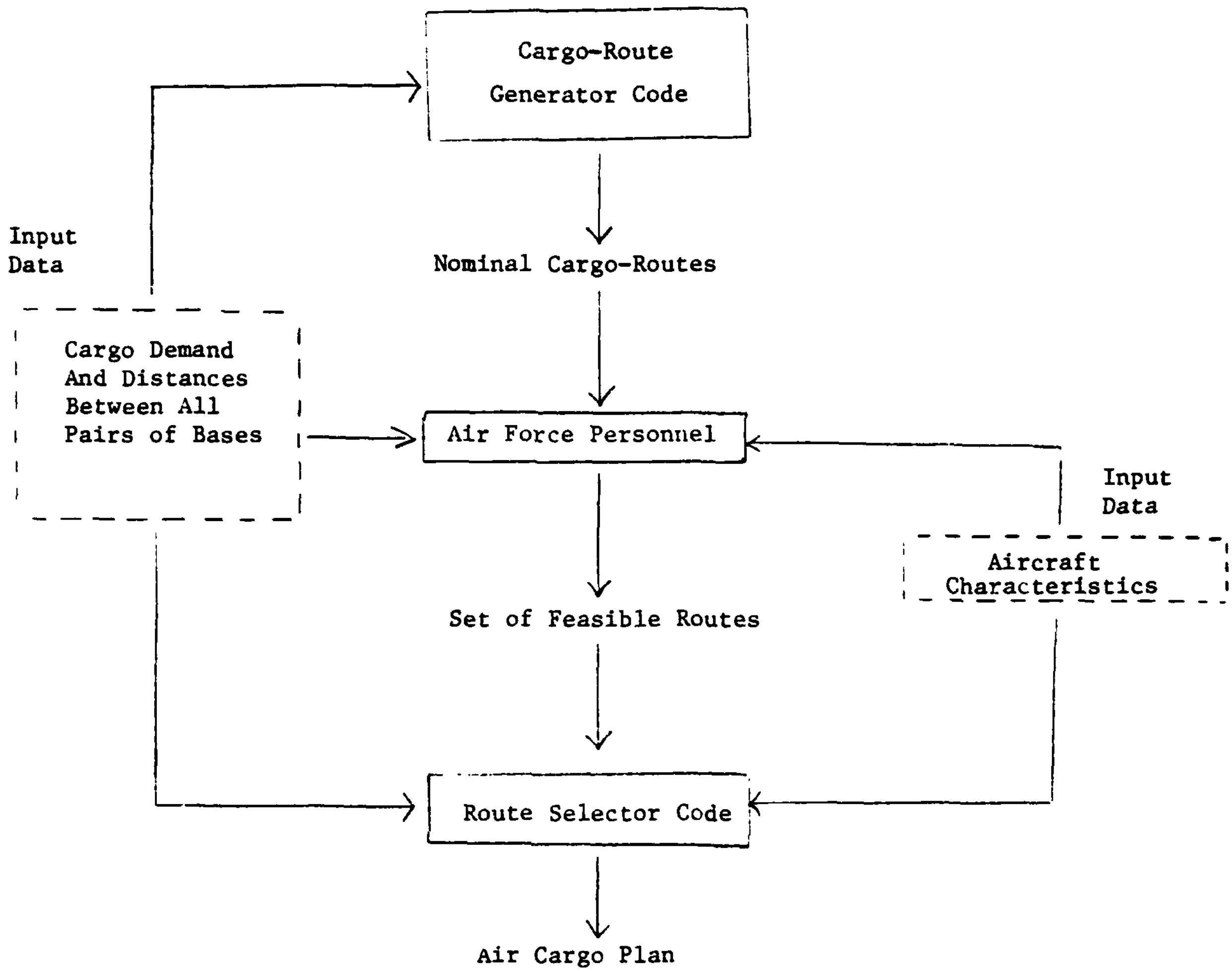


Figure 1. Data and codes used to develop an Air Cargo Plan.

Letting N denote the number of bases, there are a potential of N(N-1) arcs (i.e. total arcs in a complete network). Suppose $M \leq N(N-1)$ arcs are selected for consideration and let A denote the corresponding N by M node-arc incidence matrix. Let c_j for $j = 1, \ldots, M$, denote the flying distance associated with each of these arcs, and let $\underline{c}$ denote the vector of distances. For each pair of bases, (i,j), let d_{ij} denote the total quantity of serviceable spares to be shipped from base i to base j in units of pounds per day. Since the capacity of the aircraft must be shared by all goods with various origin-destination pairs, these must be distinguished in the model. For our models the commodities are associated with the nodes of origin. We let the node length vector $\underline{r}^k$ denote the requirement vector for commodity k. If $r_i^k > 0$, then node i is called a supply point for commodity k with supply of r_i^k. If $r_i^k < 0$, then node i is called a demand point for commodity k with demand of $|r_i^k|$. For the Air Force problem the requirement vectors are defined as follows:

$$r_i^k = \begin{cases} -d_{ki}, & k \neq i \\ \sum\limits_{\substack{j=1 \\ j \neq i}}^{N} d_{ij}, & k = i \end{cases} \qquad \begin{array}{l} k = 1, \ldots, N, \\ i = 1, \ldots, N. \end{array}$$

Letting x_j^k denote the flow of commodity k in arc e_j with corresponding vector $\underline{x}^k$, the Flow Generator Model seeks a flow $\underline{y}$ which satisfies the demand while minimizing system pound-miles. Mathematically the Flow Generator Model may be stated as follows:

$$\left.\begin{array}{ll}\min & \underline{cy} \\ \text{s.t.} & A\underline{x}^k = \underline{r}^k,\ \underline{x}^k \geq \underline{0}, \quad k = 1, \ldots, N \\ & A\underline{y} = \underline{0}\ ,\ \underline{0} \leq \underline{y} \leq \underline{u} \\ & \sum_{k=1}^{N} \underline{x}^k \leq \underline{y}\,. \end{array}\right\} \tag{1}$$

Given an optimal solution to (1), say $(\bar{\underline{x}}^1, \bar{\underline{x}}^2, \ldots, \bar{\underline{x}}^N, \underline{y}^*)$, we decompose $\underline{y}^*$ into a set of vector-circuits, $\underline{z}^1, \ldots, \underline{z}^p$, and positive multipliers $\alpha_1, \ldots, \alpha_p$, such that $\underline{y}^* = \sum_{i=1}^{p} \alpha_i \underline{z}^i$. The vector-circuits define nominal routes for the system. The algorithm used to obtain the vector-circuits may be found in Ali [5].

Using the nominal routes as a guide, a set of approximately 25 feasible routes are input to an integer program for final route selection. Suppose there are L routes in the feasible set. Let the set $R_\ell = \{e_{j_1}, e_{j_2}, \ldots, e_{j_q}\}$ denote the arcs in route ℓ. Let the arc set be given by $A_{\ell=1,\ldots,L} \bigcup R_\ell$. Then the network used in the Route Selector Model is $[N, \hat{A}]$ where $N = \{1, \ldots, N\}$. Let $\hat{A}$ denote the node-arc incidence matrix associated with $[N, \hat{A}]$. Letting f_ℓ and b_ℓ denote the fixed charge and aircraft capacity for route ℓ, respectively; the Route Selector Model is given by

$$\min \sum_{k=1}^{N} \underline{cx}^k + \sum_{\ell=1}^{L} f_\ell y_\ell \tag{2}$$

$$\text{s.t.} \quad \hat{A}\underline{x}^k = \underline{r}^k, \qquad k = 1, \ldots, N \tag{3}$$

$$\sum_{k=1}^{N} x_j^k \leq b_\ell, \qquad \text{for all } e_j \in R_\ell \text{ and } \ell = 1, \ldots, L \tag{4}$$

$$\sum_{e_j \epsilon R_\ell} \sum_{k=1}^{N} x_j^k \leq M^* y_\ell, \quad \ell = 1, \ldots, L \qquad (5)$$

$$y_\ell \; \epsilon \; \{0, 1\}, \quad \ell = 1, \ldots, L \qquad (6)$$

$$\underline{x}^k \geq 0, \quad k = 1, \ldots, N, \qquad (7)$$

where M^* is a large positive number. Constraint (4) insures that the aircraft capacity is not exceeded and constraint (5) forces the binary route variable y_ℓ to assume the value of 1 if route ℓ is used. The above model is a multicommodity fixed charge network flow problem. Solution of (2) - (7) provides a set of optirum routes from the set of L feasible routes which if flown daily will guarantee that the daily demand is met subject to aircraft capacity constraints. The underlying assumptions associated with this model are as follows:

(i) All cargo has the same priority.

(ii) Loading and unloading costs have been ignored.

(iii) Cargo volume restrictions have been ignored. (However, these can be incorporated into the model at the expense of increasing the number of constraints).

(iv) Circuitous routing is allowed to meet the demand constraints.

The integrated man-machine system used to develop an air cargo plan is illustrated in Figure 2.

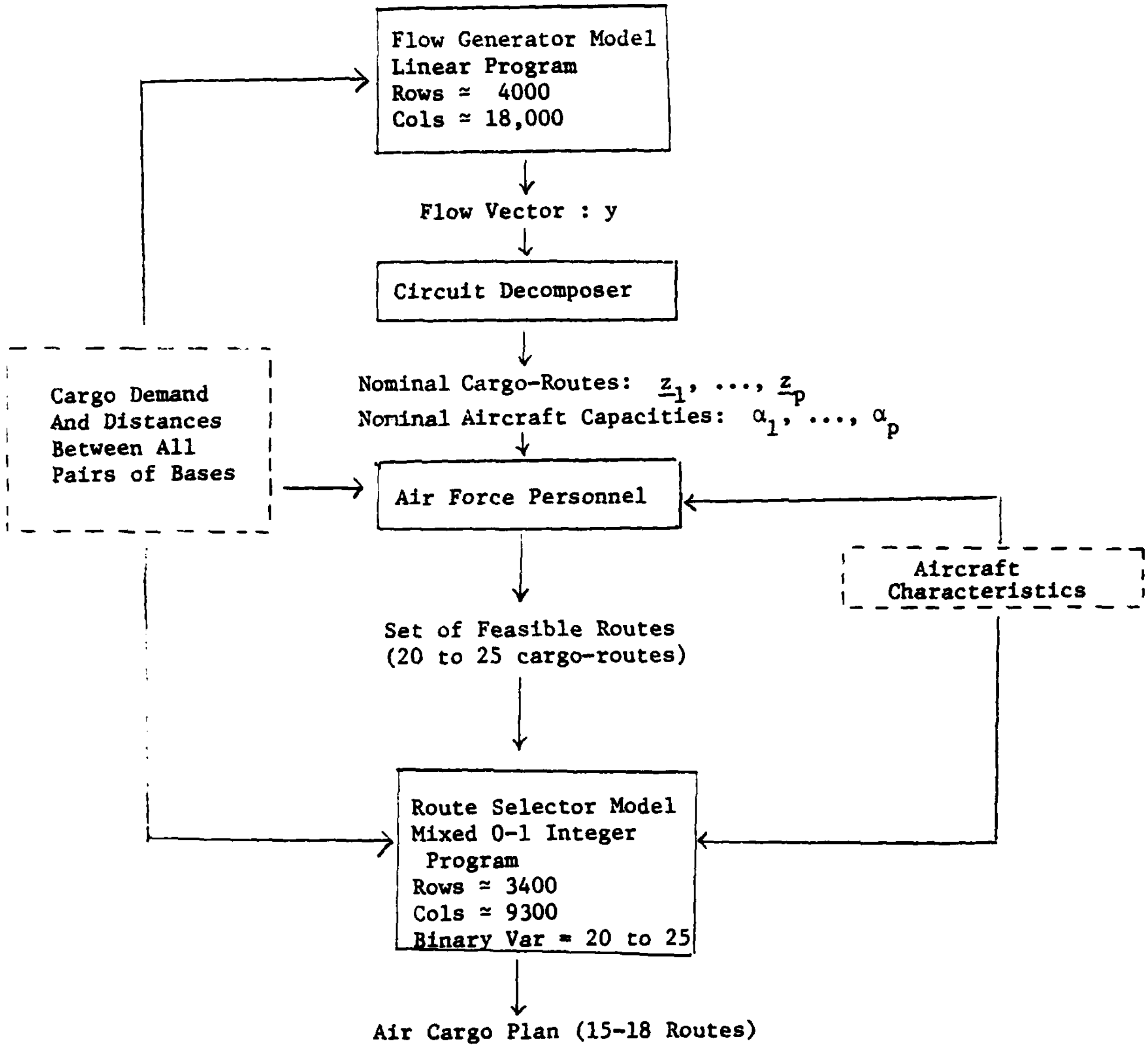


Figure 2. Procedure used to develop an Air Cargo Plan.

2.5 Computational Experience

The primal partitioning code for solving multicommodity network flow problems reported in [4] has been specialized for the Flow Generator Model (see [31] for a complete description of the primal partitioning algorithm). This system carries the inverse of the working basis in product form using the technique described in [31]. The reinversion routine is based on the work of Hellerman and Rarick [36] and uses the spike swapping procedure described in [32]. A simple circuit identifier algorithm has also been coded. Both codes are written in standard FORTRAN and have been run on a CDC Cyber 73.

The Civil Aeronautics Board provided the distance matrix for the 60 Air Force Bases in the continental U. S. A. and the Air Force Logistics Command provided the point-to-point demands (d_{ij}) for the fiscal years 1979 and 1980. From this data two test problems for each year were generated. The two test problems differ only in the number of arcs used to define the network used in the Flow Generator Model.

The termination criterion used for problems 2 and 4 was to check the objective function every 1000 iterations and terminate if the objective function value became less than 5×10^8. This number was selected arbitrarily, though keeping in mind that the routes generated by AFLC personnel for 1980 yielded a cost of 222.5×10^8 pound-miles. The two smaller problems, problems 1 and 3, were solved to optimality. Table 2 summarizes relevant information obtained in solving these problems. Note that the vector-circuit generator takes only a few seconds while the multicommodity code requires more than 20 minutes to obtain an acceptable solution.

TABLE 2. COMPUTATIONAL EXPERIENCE WITH ROUTE GENERATOR MODEL

Row Description	Problem Number			
	1	2	3	4
PROBLEM CHARACTERISTICS				
Nodes	60	60	56	56
Arcs	249	313	237	305
Commodities	61	61	57	57
LP rows	3967	4031	3483	3551
LP columns	15009	18913	14277	18425
Linking Constraints	246	310	234	302
Data Source Year	1979	1979	1980	1980
SOLUTION STATISTICS				
Objective function value x 10^9	.36235	.41256	.15632	.44889
Termination Criterion*	optimal	conditional	optimal	conditional
Time in CP minutes	21	29	19	26
Iterations	11,605	13,000	11,349	12,000
Reinversions	77	87	76	76
Time for vector-circuit selector in seconds	6.3	8.1	6.2	7.8

*Conditional termination when obj value < 5 x 10^9.

We also designed and implemented a large-scale FORTRAN computer code to be used in obtaining the solution of the Route Selector Model. The code employs a branch-and-bound scheme with separation and candidate selection guided by heuristic rules. The free integer variable furthest from an integral value is chosen for separation. The candidate subproblems most recently created are chosen first with preference given to those whose separation variable was fixed at 1 when created. The branch-and-bound tree was kept on disk in groups of 16 nodes, making use of the CDC mass storage input/output subroutines. The system was designed to allow the user to terminate a run with the current branch-and-bound tree and later restart with that tree.

It is shown in [33] that the continuous relaxation of a candidate subproblem can be formulated as a minimum cost multicommodity network flow problem. Thus we make use of a specialization of the primal partitioning code of [4] for efficient solution of the relaxed candidate subproblems. The route selector system was tested on the 1980 data using 17 routes supplied by AFLC personnel and 8 routes developed using the Flow Generator Model. This yielded a fixed charge multicommodity model having 25 binary variables, 9349 continuous variables, and 3355 constraints.

Beginning with an initial feasible solution supplied by AFLC personnel, the system was used to generate a branch-and-bound tree having 1023 nodes. This required 46 restarts and took approximately 23 hours of computer time over a 3 week period. At the termination of the run, there were 15 nodes remaining in the candidate list. Only one new incumbent was developed during the computation but the estimated cost savings of this incumbent was approximately $800,000. The new route structure involved the substitution of one of the 8 routes generated by the Flow Generator Model for one of the original 17 supplied by the Air Force.

2.6 Implementation

The two models and specialized software systems described above evolved over the period 1976 - 1980. All code development was done at Southern Methodist University by the authors for the Directorate of Transportation located at Wright-Patterson Air Force Base in Dayton, Ohio. Transportation personnel had many years of experience with the Air Force system, but they had little background in mathematical analysis and no background in either mathematical or computer programming. Even though AFLC personnel were unfamiliar with optimization models, the problem was ideally suited for operations research analysis. The important characteristics which made this study feasible are as follows:

(i) The problem was well-defined.

(ii) It was a planning (as opposed to an operational) problem in which the plan was reevaluated annually.

(iii) The problem involved a large cash outlay, \$50,000,000. Hence a 1% savings was very significant.

(iv) Most of the data was already being collected and stored on magnetic tape. There was essentially no new data which had to be collected by the client.

(v) The client had been attempting to solve the problem manually and had an appreciation for the complexity of the problem.

Rather than implement both models simultaneously, we chose to install only the Route Selector Model in which all binary variables are fixed by the user. The user selects the routes and the system loads the routes to optimally satisfy the demand. An elaborate report generator was attached to this system to provide the client with detailed information about flow in the system. In particular, legs of routes running at 100% capacity and underutilized legs are highlighted. This system has been implemented at Wright-Patterson and was used to develop the air cargo plan for fiscal year 1981. The client was very pleased with this basic system and was able to run the system and interpret the results without the aid of the authors. The system is currently being used to develop the annual routing plan.

III TRANSFORMING LINEAR PROGRAMS INTO NETWORKS WITH SIDE CONSTRAINTS

Since the development of the primal simplex method by George B. Dantzig in 1947, linear programming has been used as a fundamental planning tool for solving a wide variety of problems in industry and government. Due to the development of extremely efficient solution algorithms, a special class of linear programs known as network models have emerged as one of the most important models available to operations research analysts. Since the constraint matrices of real world linear programs usually have only a few nonzero elements (i.e. more than 98% of the matrix elements are zero) we are convinced that these problems either contain large embedded networks or can be transformed to a problem which contains a large network. If this is the case, then techniques which combine linear programming technology with network technology can be used to solve such problems.

The underlying hypothesis of this project is that general linear programs can best be solved by transforming them to networks with side constraints. Unlike the theory of linear programming which is based on mathematical results from linear algebra and convex analysis, this investigation is cast in the mathematical framework of matroid theory.

3.1 Linear Network Models

A network is composed of two types of entities: arcs and nodes. The arcs may be viewed as unidirectional means of commodity transport, and the nodes may be interpreted as locations or terminals connected by the arcs. Hence, arcs may represent aircraft flights in a distribution system, streets and highways in an urban transportation network, pipes in a water distribution network, telephone lines in a communication network, and so on. The structure of a network can be displayed by means of a labeled drawing in which nodes are represented by circles and arcs are represented by line segments incident on two nodes. An arrowhead on the line segment indicates the arc direction.

The structure of a network may also be described by a matrix, defined as follows:

$$A_{ij} = \begin{cases} +1, & \text{if arc } j \text{ is directed away from node } i \\ -1, & \text{if arc } j \text{ is directed toward node } i \\ 0, & \text{otherwise} \end{cases}$$

The matrix A defined above is called a node-arc incidence matrix. A characteristic of this matrix is that each column has exactly two non-zero entries, one being +1 and the other a -1. Any matrix (regardless of origin) having this characteristic is called a node-arc incidence matrix.

The minimal cost network flow problem is a linear program whose constraint matrix is a node-arc incidence matrix. Mathematically this problem may be stated as follows:

$$\min \quad \underline{c}\underline{x}$$

$$\text{s.t.} \quad A\underline{x} = \underline{r}$$

$$\underline{0} \leq \underline{x} \leq \underline{u}$$

where $\underline{c}$ is a known 1 x n vector,

$\underline{u}$ is a known n x 1 vector,

$\underline{0}$ is an n x 1 vector of zeroes,

$\underline{r}$ is a known m x 1 vector,

A is an m x n node-arc incidence matrix, and

$\underline{x}$ is an n x 1 vector of decision variables.

Since the expository papers by Ellis Johnson [39, 40] in 1965, tremendous advances have been made in the area of solution techniques for network related problems. This work was led primarily by Glover and Klingman and their colleagues at the University of Texas (see [3, 17, 22, 23, 24, 25, 26, 27, 28, 29]). Contributions have also been made by Srinivasan and Thompson [51, 52] and by Bradley, Brown, and Graves [14]. The author and his colleagues at Southern Methodist University have been actively extending these ideas to the more complicated network structure found in multicommodity network flow problems (see [3, 4, 32, 42, 43, 44]). Computational experimentation has shown that the new methodology is approximately 200 times faster on pure networks and as much as 25 times faster on more complex embedded network problems.

We call a matrix M a <u>network matrix</u> if it has the following three properties:

(P1) The nonzero entries of M are either +1 or -1.

(P2) No column of M has more than two nonzero entries.

(P3) If a column of M has two nonzero entries, then one is a +1 and the other is a -1.

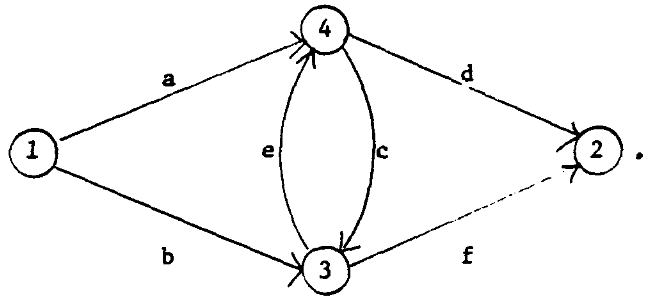
A network matrix M can be transformed to a node-arc incidence matrix by simply appending a row which is the negative of the sum of all other rows. Consider

$$M = \begin{array}{c|cccccc|c} & a & b & c & d & e & f & \\ \hline & 1 & 1 & & & & & \text{row 1} \\ & & & & -1 & & -1 & \text{row 2} \\ & & -1 & -1 & & 1 & 1 & \text{row 3} \end{array}$$

Appending a row which is the negative of the sum of the other rows yields the node-arc incidence matrix,

$$A = \begin{array}{c|cccccc|c} & a & b & c & d & e & f & \\ \hline & 1 & 1 & & & & & \text{row 1} \\ & & & & -1 & & -1 & \text{row 2} \\ & & -1 & -1 & & 1 & 1 & \text{row 3} \\ & -1 & & 1 & 1 & -1 & & \text{row 4} \end{array}$$

which corresponds to the network



Hence, any linear program whose constraint matrix is a network matrix may be solved as a minimal cost network flow problem.

3.2 Reducibility of a Linear Program

Since pure network problems are at least two orders of magnitude easier to solve than general linear programs, several researchers have addressed the following problem:

> *"When is a general linear program reducible (i.e. transformable) to a minimal cost network flow problem?"*

Consider the general linear program in the following form:

$$\min \ \underline{c}\underline{x} \tag{8}$$

$$\text{s.t.} \ \bar{A}\underline{x} = \underline{b} \tag{9}$$

$$\underline{0} \le \underline{x} \le \underline{u}, \tag{10}$$

where $\underline{c}$ is a known 1 x n vector,

$\underline{u}$ is a known n x 1 vector,

$\underline{0}$ is an n x 1 vector of zeroes,

$\bar{A}$ is an m x n matrix, and

$\underline{x}$ is an n x 1 vector of decision variables.

Suppose $\bar{A}$ takes the form $\bar{A} = [I \mid A]$. This form is always obtainable by the addition of artificial variables with corresponding bound, u_i, equal zero. Let T be an m x m nonsingular matrix, let R be an m x m nonsingular diagonal matrix, and let D be an n x n nonsingular diagonal matrix. Letting $\underline{x} = D\underline{y}$ and premultiplying (9) by TR yields the equivalent linear program,

$$\min \ \hat{\underline{c}}\underline{y} \tag{11}$$

$$\text{s.t. } \hat{A}\underline{y} = \hat{\underline{b}} \tag{12}$$

$$\underline{0} \leq D\underline{y} \leq \underline{u} \tag{13}$$

where $\hat{\underline{c}} = \underline{c}D$,

$\hat{A} = TR\bar{A}D$, and

$\hat{b} = TR\underline{b}$.

The problems (8) - (10) and (11) - (13) are equivalent in the sense that *if* $\underline{x}^*$ *solves* (8) - (10), *then* $\underline{y}^* = D^{-1}\underline{x}^*$ solves (11) - (13) and if $\underline{y}^*$ solves (11) - (13), then $\underline{x}^* = D\underline{y}^*$ solves (8) - (10). Furthermore, we say that (8) - (10) is reducible to a network problem if and only if $TR\bar{A}D$ is a network matrix (see properties (P1) - (P3) on page 22). But

$$TR\bar{A}D = TR[I \mid A]\begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix} = T[RD_1 \mid RAD_2].$$

Without loss of generality we may require that $D_1 = R^{-1}$. Then $T[RD_1 \mid RAD_2] = T[I \mid RAD_2]$. Therefore, the Reducibility Problem may be stated as follows:

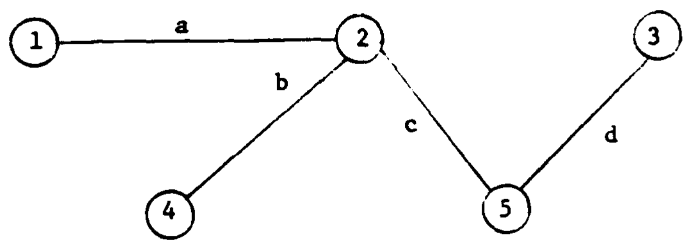
> *"Given a matrix $\bar{A} = [I \mid A]$, does there exist a nonsingular matrix T and nonsingular diagonal matrices R and D such that $[T \mid TRAD]$ is a network matrix?"*

3.3 Necessary Conditions for Reducibility

Several researchers have addressed the r· ucibility problem and necessary conditions on T, R, and D are known. It is clear that if $T[I \,\vdots\, RAD]$ is a network matrix, then T is itself a network matrix. It is shown in Bartholdi and Ratliff [8] that a nonsingular network matrix corresponds to a tree (i.e. a connected graph having one less arc than node). For example the nonsingular matrix

$$T = \begin{array}{cc} \begin{array}{cccc} a & b & c & d \end{array} & \\ \left[\begin{array}{cccc} -1 & & & \\ 1 & -1 & -1 & \\ & & & 1 \\ & 1 & & \end{array}\right] & \begin{array}{l} \text{row } 1 \\ \text{row } 2 \\ \text{row } 3 \\ \text{row } 4 \end{array} \end{array}$$

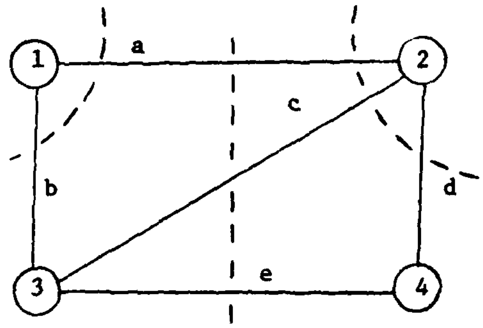
corresponds to the tree



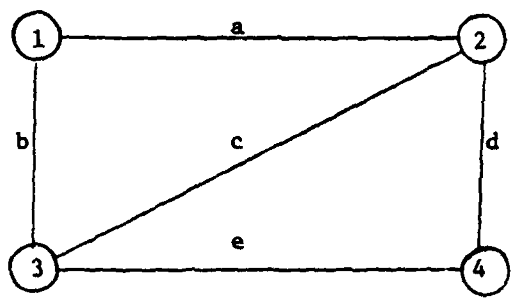
,

where node (row 5) does not appear in T. Therefore, $T[I \,\vdots\, RAD]$ is a network matrix only if T is a tree.

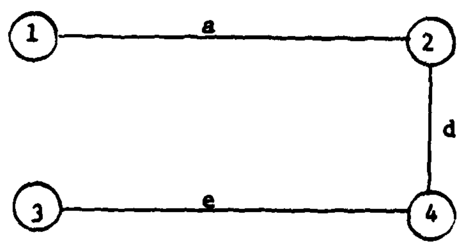
A <u>cut-set</u> for a connected graph G is a set of edges whose removal results in a disconnected graph and is minimal with respect to this property. For example in the graph



{a, b}, {a, c, e}, and {a, c, d} are all cut-sets. There is a dual relationship between the spanning tree of a graph and a cut-set. Recall that a spanning tree is a minimal set of edges which connects all vertices of a graph, whereas a cut-set is a minimal set of links which disconnects some vertices from others. From this observation it is obvious that any spanning tree must have at least one link in common with every cut-set. The set of <u>fundamental cut-sets</u> associated with a spanning tree having n arcs is composed of the n cut-sets each having one of the n edges from the tree. For example, for the graph



with spanning tree



the fundamental cut-sets are {a, b}, {d, b, c}, and {e, b, c}. The corresponding cut-set matrix is given by

$$\begin{array}{c|c} & \begin{array}{ccccc} a & d & e & b & c \end{array} \\ \begin{array}{c} a \\ d \\ e \end{array} & \left[\begin{array}{ccc|cc} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{array}\right], \end{array} \begin{array}{l} \text{cut-set } \{a, b\} \\ \text{cut-set } \{d, b, c\} \\ \text{cut-set } \{e, b, c\} \end{array}$$

where the rows correspond to cut-sets and the columns correspond to edges. The following result gives a relationship between a cut-set matrix and the corresponding network matrix.

Proposition 1.

Let T be a nonsingular network matrix corresponding to a tree. Then $T[I \mid E]$ is a network matrix only if $[I \mid E] \bmod 2$ is a fundamental cut-set matrix corresponding to T.

We now state another necessary condition for reducibility.

Proposition 2.

Let T be a nonsingular network matrix corresponding to a tree. Then $T[I \mid R_1AD_1]$ is a network matrix only if the nonzero elements of R_1AD_1 are ± 1.

The above two propositions provide the basis for a transformation algorithm given below.

TRANSFORMATION ALGORITHM

0. Begin with the constraint matrix $[I \mid A]$.
1. Does there exist nonsingular diagonal matrices R_1 and D_1 such that the nonzero elements of R_1AD_1 are ± 1?

   No - Then $[I \mid A]$ is not transformable by Proposition2.

   Yes - Continue with step 2.

2. Is $[I \mid R_1AD_1]$ mod 2 a fundamental cut-set matrix for some graph, say G?

No - Then $[I \mid A]$ is not transformable by Proposition 1.

Yes - Continue with step 3.

3. Direct the arcs of G arbitrarily and define the corresponding node-arc incidence matrix by N_1. Let N_2 be formed from N_1 by omitting one row. Let T correspond to the columns of N_2 associated with some spanning tree of G. Partition N as $[T \mid N_3]$. Convert N_2 to standard form by premultiplying by T^{-1}. This gives $[I \mid T^{-1}N_3]$.

4. Do there exist diagonal nonsingular matrices R_2 and D_2 having nonzero elements ± 1 such that $R_2[I \mid R_1AD_1] \left[\begin{array}{c|c} R_2^{-1} & \\ \hline & D_2 \end{array}\right] = [I \mid T^{-1}N_3]$?

Yes - Then $[I \mid A]$ is transformable to a network matrix using

$R = R_2R_1$, $D = D_1D_2$ and T. That is, $TR_2R_1[I \mid A]$

$\left[\begin{array}{c|c} R_1^{-1} & \\ \hline & D_1 \end{array}\right] \left[\begin{array}{c|c} R_2^{-1} & \\ \hline & D_2 \end{array}\right]$ is a network matrix.

No - Transformation algorithm is not successful.

3.4 Matroid Theory

The question of when a matrix, $[I \mid E]$ is the cut-set matrix for some graph has been addressed by Tutte [54] in the mathematical framework known as matroid theory. A matroid (Welsh [57]) is a mathematical structure consisting of a finite set E and a finite set C of nonempty subsets of E such that two properties hold.

(M1) If $X \neq Y \in C$, then $X \not\subseteq Y$.

(M2) If X, Y are distinct members of C and $a \in X \cap Y \neq \phi$ there exists $Z \in C$ such that $Z \subseteq (X \cup Y) - \{a\}$.

For example E = {1, 2, 3, 4, 5} and C = {1, 4}, {2, 4, 5}, {3}, {1, 2, 5}} is a matroid.

An m x n matrix R is called a <u>binary matrix</u> if the nonzero elements of R are ones. Let F denote the set of binary m-vectors generated by all modulo 2 row sums of the binary matrix R. That is, F = {$\underline{r} = \underline{\alpha}R$ and mod 2 : α_i is integer for all i}. The elements of F are called <u>chains</u> by Tutte [55]. Define the support of a <u>chain</u>, $\underline{f} \in F$, denoted by $||\underline{f}||$ as follows:

$$||\underline{f}|| = \{i : f_i \neq 0\}.$$

A chain, $\underline{f} \neq \underline{0}$, of F is said to be <u>elementary</u> if there is no chain of F which is a proper subset of F. Letting E = {1, ..., n} and C equal the set of elementary supports, we have what Tutte [55] calls a <u>binary matroid</u>, M(R). For example, let

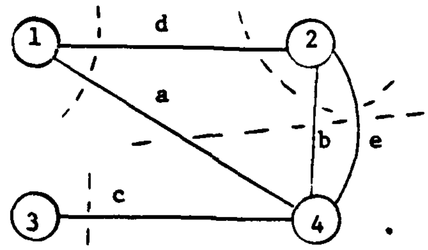
$$R = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

Then the set of binary vectors generated by the mod 2 sum of rows of R is given by

Number	$\underline{\alpha}$	$\underline{f}$	$\vert\vert\underline{f}\vert\vert$
1	[1 0 0]	[1 0 0 1 0]	{1, 4}
2	[0 1 0]	[0 1 0 1 1]	{2, 4, 5}
3	[0 0 1]	[0 0 1 0 0]	{3}
4	[1 1 0]	[1 1 0 0 1]	{1, 2, 5}
5	[1 0 1]	[1 0 1 1 0]	{1, 3, 4}
6	[0 1 1]	[0 1 1 1 1]	{2, 3, 4, 5}
7	[1 1 1]	[1 1 1 0 1]	{1, 2, 3, 5}

The elementary supports are {1, 4}, {2, 4, 5}, {3}, and {1, 2, 5}, and the matroid is given by E = {1, 2, 3, 4, 5} and C = {{1, 4}, {2, 4, 5}, {3}, {1, 2, 5}}.

Another procedure for generating a binary matroid is to consider the cut sets of a given finite graph having n edges. Letting E = {1, ..., n} and C = the set of cut sets for G, we also have a binary matroid. For example, suppose G is



Then E = {a, b, c, d, e} and C = {{a, d}, {b, d, e}, {c}, {a, b, e}} forms a binary matroid. A matroid formed in this way is also called the <u>bond matroid</u> of the graph G. A binary matroid is said to be <u>graphic</u> if it can be represented as the bond matroid of some graph. All graphic matroids are binary but the converse is false. The question of whether a given binary matroid is graphic is equivalent to the question, when is a binary matrix, [I ¦ E], the cut-set matrix some graph? Tutte [54] provided an algorithm to determine when a binary matroid is graphic and Bixby and Cunningham [13] described a precise procedure based on Tutte's work to generate the corresponding graph. Heller [34, 35], and Iri [37, 38], have also discussed the reducibility problem but no one has ever experimentally tested any of these procedures on a set of large real world linear programs.

3.5 Practical Significance

We conjecture that at least 90% of the real world linear programs are <u>not</u> reducible to minimal cost network flow problems. Therefore, from a strictly practical point of view, the rich theory developed to date can not be applied directly to help solve a large class of real world linear programs. However, modification of these results to obtain networks with side constraints could be extremely valuable.

3.6 A Heuristic Algorithm

The reducibility algorithms of Bixby-Cunningham [13] and Iri [38] either find the matrices T, R, and D required for the transformation or conclude that no such matrices exist. We have modified these ideas so that they can be used to produce an embedded network. This work could drastically change the way we view linear programs and could substantially improve the state-of-the-art software for solving linear programs.

Consider the following linear program which has been called the <u>network problem with side constraints</u>,

$$\min \quad \underline{c}\underline{x} + \underline{d}\underline{z} \tag{14}$$

$$\text{s.t.} \quad A\underline{x} = \underline{r} \tag{15}$$

$$S\underline{x} + P\underline{z} = \underline{b} \tag{16}$$

$$\underline{0} \le \underline{x} \le \underline{u} \tag{17}$$

$$\underline{0} \le \underline{z} \le \underline{r} \tag{18}$$

where

$\underline{c}$ is $1 \times n_1$,

$\underline{d}$ is $1 \times n_2$,

$\underline{r}$ is $m_1 \times 1$,

$\underline{b}$ is $m_2 \times 1$,

$\underline{u}$ is $n_1 \times 1$,

$\underline{r}$ is $n_2 \times 1$,

A is $m_1 \times n_1$,

S is $m_2 \times n_1$,

P is $m_2 \times n_2$, and

A is a network matrix. If $m_1/(m_1 + m_2) \geq .5$, then we believe that a special algorithm applied to (14) - (18), such as the Simplex Son [27], will be substantially superior to the standard primal simplex method applied to this problem.

We shall say that an m x n matrix $[I \mid C]$ is <u>p-reducible</u> if there exists a nonsingular p x p matrix T, and there exists nonsingular diagonal p x p matrices R and D, such that for some p rows of $[I \mid C]$, say $[I \mid C_1]$, $TR[I \mid C_1] \left[\begin{array}{c|c} R^{-1} & \\ \hline & D \end{array}\right]$ is a network matrix. That is, the matrix $[I \mid C]$ can be transformed as follows:

$$\left[\begin{array}{c|c} T & \\ \hline & I \end{array}\right] \left[\begin{array}{c|c} R & \\ \hline & I \end{array}\right] \left[\begin{array}{c|c|c|c} I & & C_1 & \\ \hline & I & C_2 & C_3 \end{array}\right] \left[\begin{array}{c|c|c|c} R^{-1} & & & \\ \hline & I & & \\ \hline & & D & \\ \hline & & & I \end{array}\right] =$$

$$\left[\begin{array}{c|c|c|c} T & & TRC_1D & \\ \hline & I & C_2D & C_3 \end{array}\right]. \tag{19}$$

The matrix (19) is, of course, the constraint matrix for a p-node network with side constraints.

Theoretically, the problem of partial network transformation can be described as follows:

> *"Given any matrix $\bar{A} = [I \mid A]$, find the largest p such that $\bar{A}$ is p-reducible."*

The above described problem is NP-complete and we believe that there is little hope of finding an exact solution which could be used to enhance linear programming software. However, a good heuristic based upon the rich theory available could prove to be extremely useful.

We now present a heuristic algorithm which we believe holds the best hope for development of an automatic procedure for converting linear programs to networks with side constraints.

0. Initialization

Begin with the linear system

$$A^1\underline{x} = \underline{b}^1 \tag{20}$$

1. Scale To $\pm$ 1's

Using only row and column scaling,

let

$$A^2\underline{x} = \underline{b}^2 \tag{21}$$

denote a subset of rows of (20) which have been scaled to the elements 0, $\pm$ 1.

2. Apply Brown-Wright Heuristic

Let $A^3\underline{x} = \underline{b}^3$ (22)

denote the rows of (21) which correspond to a network matrix as obtained by the Brown-Wright heuristic.

3. Transformation

Let

$$A^4\underline{x} = \underline{b}^4 \tag{23}$$

denote the rows of (21) not appearing in (22). Apply a modification of the Bixby-Cunningham algorithm to attempt to build a tree which transforms part of (23) to a network matrix.

The above algorithm will be fast; however, no results concerning the size of the network generated are available. No computational experience with this algorithm is available at this time.

REFERENCES

1. Agin, N. I., and D. E. Cullen, "An Algorithm for Transportation Routing and Vehicle Loading", North-Holland/TIMS Studies in The Management Sciences, 1, (1975), 1-20.

2. Aho, A. V., J. E. Hopcroft, and J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison, Wesley, Reading, Mass., (1974).

3. Ali, A., R. Helgason, J. Kennington, and H. Lall, "Primal Simplex Network Codes: State-Of-The-Art Implementation Technology," Networks, 8, 315-339, (1978).

4. Ali, A., R. Helgason, J. Kennington, and H. Lall, "Computational Comparison Among Three Multicommodity Network Flow Algorithms," Operations Research, 28 (4), (1980), 995-1000.

5. Ali, A. I., "Two Node-Routing Problems," unpublished dissertation, *Department of Operations Research*, Southern Methodist University, Dallas, Texas (1980).

6. Angel, R. D., W. L. Caudle, R. Noonan, and A. Whinston, "Computer Assisted School Bus Scheduling", Management Science, 18(6), (1972), B279-B288.

7. Barr, R., F. Glover, and D. Klingman, "A New Optimization Method for Large-Scale Fixed Charge Transportation Problems," Operations Research, 29 (3) (1981), 448-463.

8. Bartholdi, J. J., and H. D. Ratliff, "A Field Guide to Identifying Network Flow and Matching Problems," Research Report No. 77-12, University of Florida, Gainesville, Florida, (1977).

9. Bellmore, M., G. Bennington, and S. Lubore, "A Multivehicle Tanker Scheduling Problem", Transportation Science, 5, (1971), 36-47.

10. Berge, C., The Theory of Graphs and its Applications, John Wiley, New York, (1962).

11. Berge, C. and A. Ghouila-Houri, Programming, Games and Transportation Networks, John Wiley and Sons, New York (1965).

12. Billheimer, J. W., and P. Gray, "Network Design With Fixed and Variable Cost Elements", Transportation Science, 1(1), (1973), 49-74.

13. Bixby, R. E., and W. H. Cunningham, "Converting Linear Programs to Network Problems," Mathematics of Operations Research, 5 (3), (1980), 321-357.

14. Bradley, G. H., G. G. Brown, and G. W. Graves, "Design and Implementation of Large Scale Primal Transshipment Algorithms," Management Science, 24 (1), (1977), 1-34.

15. Demmy, W. S. and K. E. Brant, "Design of a Military Air Cargo Transportation System by Use of a Large Scale Mathematical Programming Model", Working Note Number 5, Systems Studies Branch Air Force Logistics Command, Wright-Patterson AFB, (undated).

16. Dial, R., F. Glover, D. Karney, and D. Klingman, "A Computational Analysis of Alternative Algorithms and Labeling Techniques for Finding Shortest Path Trees," Research Report CCS 291, Center for Cybernetic Studies, The University of Texas at Austin, (1977).

17. Elam, J., F. Glover, and D. Klingsman, "A Strongly Convergent Primal Simplex Algorithm for Generalized Networks," Mathematics of Operations Research, 4 (1), (1979) 39-59.

18. Ferguson, A. R., and G. B. Dantzig, "The Allocation of Aircraft to Routes - An example of Linear Programming Under Uncertain Demand", Management Science, 3, (1957), 45-60.

19. Frank, H. and W. Chou, "Routing in Computer Networks", Networks, 1, (1971), 99-112.

20. Frank, H., I. T. Frisch, R. Van Slyke, and W. S. Chou, "Optimal Design of Centralized Computer Networks", Networks, 1, (1971), 43- 57.

21. Geoffrion, A. M., and G. W. Graves, "Multicommodity Distribution System Design by Benders Decomposition", Management Science, 20(5), (1974), 822-844.

22. Glover, F., D. Karney, D. Klingman, and A. Napier, "A Computational Study on Start Procedures, Basis Change Criteria, and Solution Algorithms for Transportation Problems," Management Science, 20 (5), (1974) 793-813.

23. Glover, F., D. Karney, and D. Klingman, "Implementation and Computational Comparisons of Primal, Dual, and Primal-Dual Computer Codes for Minimum Cost Network Flow Problems," Networks, 4 (3), (1974) 191-212.

24. Glover, F., D. Klingman, and J. Stutz, "Augmented Threaded Index Method for Network Optimization," INFOR, 12 (3), (1974), 293-298.

25. Glover, F., and D. Klingman, "New Advances in the Solution of Large-Scale Network and Network-Related Problems," Technical Report CCS 177, Center for Cybernetic Studies, The University of Texas at Austin, (1974).

26. Glover, F., J. Hultz, and D. Klingman, "Network Versus Linear Programming Algorithms and Implementations," CCS 306, The University of Texas, Austin, TX, (1977).

27. Glover, F., and D. Klingman, "The Simplex Son Algorithm for LP/Embedded Network Problems," CCS 317, University of Texas, Austin, Texas, (1977).

28. Glover, F., and D. Klingman, "Some Recent Practical Misconceptions About the State-of-the-Art of Network Algorithms," Operations Research, 2, (1978) 370-379.

29. Glover, F., D. Karney, and D. Klingman, and R. Russell, "Solving Singly Constrained Transshipment Problems," Transportation Science, 12 (4), (1978) 277-297.

30. Hadley, G., Linear Algebra, Addison-Wesley, Reading, Massachusetts, (1961).

31. Helgason, R. V., and J. L. Kennington, "A Product Form Representation of the Inverse of a Multicommodity Cycle Matrix", Networks, 7, (1977), 297-322.

32. Helgason, R. V., and J. L. Kennington, "A Product Form Representation of the Inverse of a Multicommodity Cycle Matrix", Networks, 7, (1977), 297-322.

33. Helgason, R. V., "A Lagrangean Relaxation Approach to the Generalized Fixed Charge Multicommodity Minimal Cost Network Flow Problem", unpublished dissertation, Department of Operations Research, Southern Methodist University, Dallas, Texas (1980).

34. Heller, I., "On a Class of Equivalent Systems of Linear Inequalities," Pacific Journal of Mathematics, 3 (1963), 1209-1227.

35. Heller, I., "On Linear Programmes Equivalent to the Transportation Program," Journal of the Society for Industrial and Applied Mathematics, 12, (1964), 31-42.

36. Hellerman, E., and D. Rarick, "Reinversion With the Preassigned Pivot Procedure", Mathematical Programming, 1, (1971), 195-216.

37. Iri, M., "A Criterion for the Reducibility of a Linear Programming Problem to a Linear Network-Flow Problem," RAAG Research Notes, Third Series, No. 98 (Feb. 1966).

38. Iri, M., "On the Synthesis of Loop and Cutset Matrices and the Related Problems," RAAG Memoirs, 4, (1968), pp. 376-410.

39. Johnson, E. L., "Programming in Networks and Graphs," Technical Report ORC 65-1, Operations Research Center, University of California at Berkeley (1965).

40. Johnson, E. L., "Networks and Basic Solutions," Operations Research, 14, (1966), 619-623.

41. Kalaba, R. E., and M. L. Juncosa, "Optimal Design and Utilization of Communication Network", Management Science, 3, (1957), 33-44.

42. Kennington, J. L., and M. Shalaby, "An Effective Subgradient Procedure for Minimal Cost Multicommodity Flow Problems," Management Science, 23 (9), (1977), 994-1004.

43. Kennington, J. L., "Solving Multicommodity Transportation Problems Using a Primal Partitioning Simplex Technique," Naval Research Logistics Quarterly, 24 (2), (1977) 309-325.

44. Kennington, J. L., "A Survey of Linear Cost Multicommodity Network Flows," Operations Research, 26 (2), (1978) 209-236.

45. Marsten, R. E., and M. R. Muller, "A Mixed-Integer Programming Approach to Air Cargo Fleet Planning," Management Science, 26 (11), (1980), 1096-1107.

46. Mateti, P., and N. Deo, "On Algorithms for Enumerating all Circuits of a Graph", SIAM J. Comput., 5(1), (1976), 90-99.

47. Murchland, J. D., "Construction of a Basis of Elementary Circuits and Cocircuits in a Directed Graph", Combinatorial Structures and Their Applications, Gordon and Bread, New York, 1970, 285-288.

48. Newton, R. M., and W. H. Thomas, "Bus Routing in a Multi-School System", Comput. and Ops. Res., 1. (1974), 213-222.

49. Richardson, R., "An Optimization Approach to Routing Aircraft", Transportation Science, 10 (1), (1976), 52-71.

50. Silman, L. A., Z. Barzily, and U. Passy, "Planning the Route System for Urban Buses", Comput. and Ops. Res., 1, (1974), 201-211.

51. Srinivasan, V. and G. L. Thompson, "Accelerated Algorithms for Labeling and Relabeling of Trees, with Applications to Distribution Problems," Journal of the Association for Computing Machinery, 19 (4), (1972), 712-726.

52. Srinivasan, V., and G. L. Thompson, "Benefit-Cost Analysis of Coding Techniques for the Primal Transportation Algorithm," Journal of the Association of Computing Machinery, 20, (1973) 194-213.

53. Tarjan, R., "Depth-First Search and Linear Graph Algorithms", SIAM J. Comput., 1(2), (1972), 146-160.

54. Tutte, W. T., "An Algorithm for Determining Whether a Given Binary Matroid is Graphic," Proceedings of American Mathematical Society, 11 (1960), 905-917.

55. Tutte, W. T., Introduction to the Theory of Matroids, American Elsevier, New York, NY (1971).

56. Verderber, W. J., "Automated Pupil Transportation", Comput. and Ops. Res., 1, (1974), 235-245.

57. Welsh, D. J. A., Matroid Theory, Academic Press, New York, NY (1976).

58. White, W. W., and E. Wrathall, "A System for Railroad Traffic Scheduling", Tech. Report No. 320-2992, IBM-Philadelphia Scientific Center, (1970).