

INTEGRATED RISK/COST PLANNING MODELS
FOR THE U.S. AIR TRAFFIC SYSTEM

John M. Mulvey

and

Stavros A. Zenios

(NASA-CR-177274) INTEGRATED RISK/COST
PLANNING MODELS FOR THE US AIR TRAFFIC
SYSTEM (Princeton Univ., N. J.) 41 p
HC A03/MF A01

N86-26312

CSSL 17G

Unclas

G3/04

43602

Department of Civil Engineering
Princeton University
Princeton, New Jersey 08544

Engineering - Economic Series
Report EES-85-9

Revised, December, 1985

This research was funded in part by NSF grant DCR-8401098 and
NASA grant NAG-1-520.

Abstract

~~We describe~~ a prototype network planning model for the U.S. Air Traffic control system. ^{ic DES} The model encompasses the dual objectives of managing collision risks and transportation costs where traffic flows can be related to these objectives. The underlying structure is a network graph with nonseparable convex costs; the model is solved efficiently by capitalizing on its intrinsic characteristics. Two specialized algorithms for solving the resulting problems are described: (1) truncated Newton, and (2) simplicial decomposition.

The feasibility of the approach is demonstrated using data collected from a control center in the Midwest. Computational results with different computer systems are presented -- including a vector supercomputer (CRAY-XMP). The risk/cost model will have two primary uses: (1) as a strategic planning tool using aggregate flight information, and (2) as an integrated operational system for forecasting congestion and monitoring (controlling) flow throughout the United States. In the latter case, access to a supercomputer will be required due to the model's enormous size.

AIR TRAFFIC CONTROL
NETWORK ANALYSIS
MATH. MODELS
RISK
COSTS
FLIGHT SAFETY

MID AIR COLLISIONS
COLLISION AVOIDANCE
ALGORITHMS
COMPUTER TECHNIQUES
SUPERCOMPUTER
CONGESTION

1. Introduction

By the year 2000, the U.S. Federal Aviation Administration (FAA) plans to spend over \$11 billion on a program to upgrade and consolidate the U.S. air-traffic system -- called the National Airspace System Plan [NASP]. On an average day in 1984 approximately 17,000 aircraft traveled these routes. It is expected that traffic volume will increase by 62% over the next decade. The FAA has responsibility for managing the air-traffic system, especially those elements affecting risk. Indeed, the agency has been able to reduce overall (and relative) risks during the past thirty years [1]. As new hardware and software technologies have been implemented, the FAA has provided mechanisms for monitoring factors associated with system risks. Recently a trend to deregulate the airline industry in the interest of greater efficiency and competitiveness has begun. Despite this trend the FAA must possess efficient procedures for assessing (and ultimately controlling) system risks.

In this report we discuss the development of a prototype network planning model for flights on high-altitude jet routes over the U.S. airspace, in conjunction with the NASP. The model encompasses the dual objective of assessing risk-related measures and transportation costs. The underlying mathematical model has the special structure of networks graph. Since safety components depend upon interacting variables, the proposed model falls in the category of a nonlinear network with nonseparable cost functions.

The model can serve as the building block for a management information system that can assist FAA in two basic settings:

- (a) As a strategic planning tool, or
- (2) As an operational planning program for air-traffic scheduling and routing.

In the former case it is essential that the optimization can be solved efficiently and accurately. In this way extensive sensitivity analyses can be carried out and answers to "what if" questions obtained in a comparative fashion and without regard to numerical instability. The use of the model as an operational planning tool depends on the ability to "solve" the model under real-time conditions.

By capitalizing on the special structure of the underlying network basis (a forest of 1-trees) we have developed highly efficient solution algorithms [2, 12, 13]. Thus large-scale problems can be handled within minutes or hours; repeated runs can be made and the use of the model for planning is feasible. Going a step further we have streamlined our algorithms for the architecture of vector supercomputers, in particular the CRAY-XMP at Boeing Computer Services. By taking advantage of the Cray's pipeline features we were able to solve the test cases in a matter of seconds -- thus demonstrating feasibility of the network model as an operational planning tool.

The rest of this paper is organized as follows. The mathematical models are defined in Section 2. A general description of two solution algorithms -- (1) Truncated Newton, and (2) simplicial decomposition -- appears in Section 3. Data requirements for the model are listed in Section 4. Next, Section 5 presents the modeling of a representative air-traffic control sector (Indianapolis); some computational results are given in Section 6. Finally, we discuss limitations of the current model and directions for future research.

2. Air Traffic Control Modeling

A stochastic programming model for the air-traffic control problem was proposed in the early work by Ferguson and Dantzig [5]. More recently, general aspects of air-traffic systems planning and design were the focus of a conference at Princeton University [1]. We propose here a network formulation for the air-traffic planning problem.

The generic nonlinear network model takes the form:

[NLGN] Minimize $F(\bar{x})$

Subject to:

$$\sum_{j \in \delta_i^+} x_{ij} - \sum_{k \in \delta_i^-} m_{ki} x_{ki} = b_i, \quad \text{for } i \in N \quad (2.1)$$

$$l_{ij} \leq x_{ij} \leq u_{ij}, \quad \text{for } (i,j) \in E \quad (2.2)$$

where

$F(\bar{x})$ = convex function

$\{N\}$ = set of nodes

$\{E\}$ = set of arcs (edges)

x_{ij} = flow over arc (i,j)

m_{ki} = multiplier* on arc (k,i)

b_i = supply/demand for node i

$\bar{x} = \{x_{ij} \mid (i,j) \in E\}$

$\delta_i^+ = \{j \mid (i,j) \in E\}$

$\delta_i^- = \{j \mid (j,i) \in E\}$

$u_{ij} (l_{ij})$ upper (lower) bound on arc (i,j)

$X = \{x_{ij} \mid x_{ij} \text{ satisfies constraints (2.1) and (2.2)}\}$

*Multipliers are indicated so that airport congestions and other factors can be modeled. These features will be discussed in Section 7.

We may rewrite [NLGN] in the following form:

[NLGN] Minimize $F(\bar{x})$

Subject to $x \in X$

where

$$X \equiv \{\bar{x} | \bar{A} \cdot \bar{x} = b, \ell \leq \bar{x} \leq \bar{u}\}$$

Since the [NLGN] basis is a "forest of 1-trees" (collection of subtrees with one extra arc per subtree creating a single loop), efficient procedures are available for storing and updating the basis and other aspects of the algorithm [9, 12].

Figure 1 depicts a simplified example graph of the first planning model. This example consists of five airports and interconnecting routes, representing an aggregate of individual flights into and out of the designated airports. Note that each airport has three triangular nodes, indicating net traffic, number of incoming flights and the number of outgoing flights. Network arcs points in the obvious direction of traffic movements.

The corresponding optimization model is shown below:

$$[\text{OPT1}] \text{ Minimize } \{w_1 \cdot \left[\sum_{(i,j) \in A_1} \sum_{r \in R_{ij}} c_{ij}^r z_{ij}^r \right] + w_2 \cdot s(\bar{z})\} \equiv F(\cdot)$$

Subject to:

$$x_i = \sum_{k \in L_i} x_{ik} \quad i \in A$$

$$y_j = \sum_{k \in M_j} y_{kj} \quad j \in A$$

$$\ell_{ik} \leq x_{ij} \leq u_{ij} \quad (i,k) \in A_1$$

$$\ell_{kj} \leq y_{kj} \leq u_{kj} \quad (k,j) \text{ such that } j \in A, k \in M_j$$

$$x_{ik} = \sum_{r \in R_{ik}} z_{ik}^r \quad i \in A, k \in L_i$$

ORIGINAL PAGE IS
OF POOR QUALITY

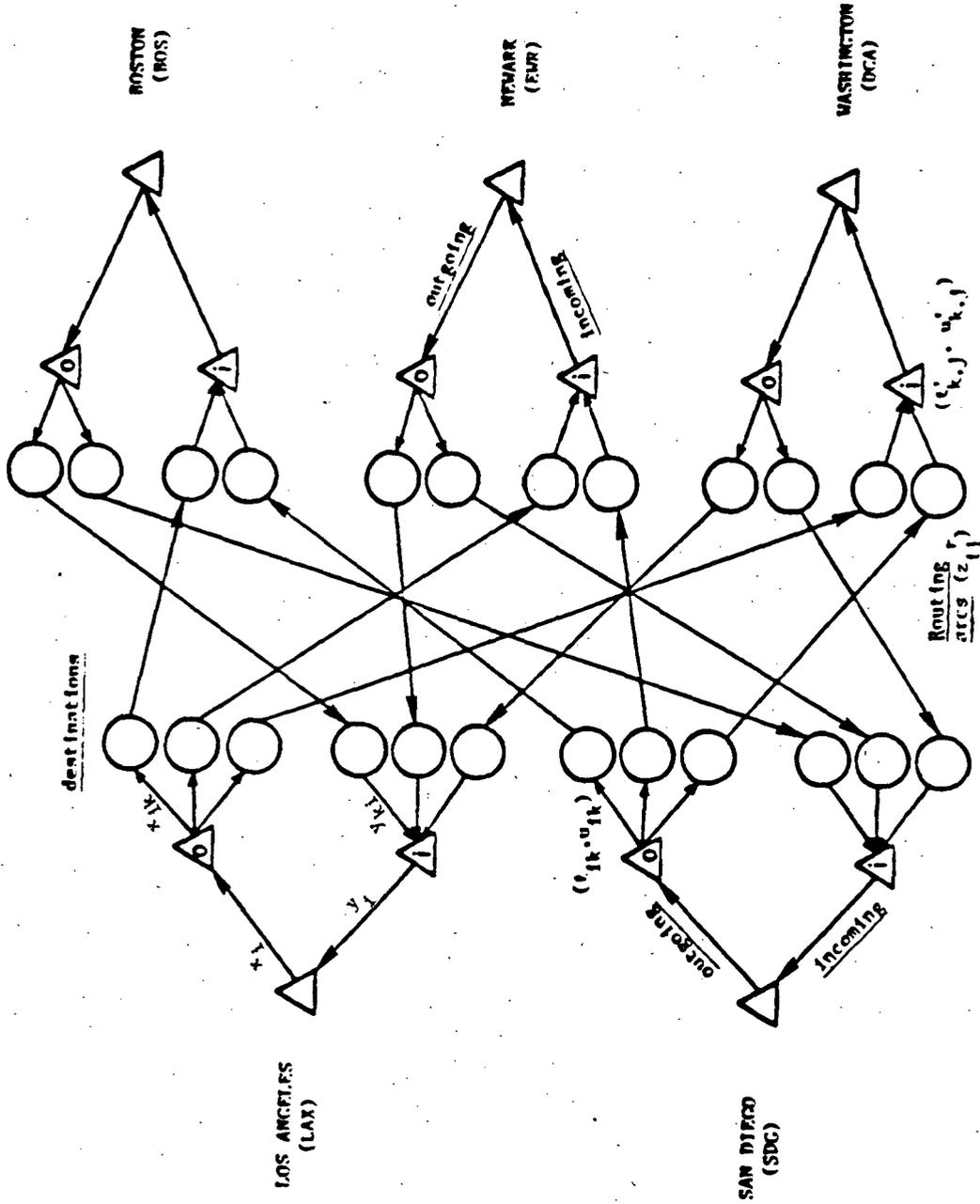


Figure 1: Example Air-Traffic Network (Strategic Model)

$$y_{kj} = \sum_{r \in R_{kj}} z_{kj}^r \quad j \in A, k \in M_i$$

$$0 \leq z_{ij}^r \leq u_{ij}^r \quad r \in R_{ij} \text{ and } (i,j) \in A_1$$

$$x_i - y_i = b_i \quad i \in A$$

$$w_1 + w_2 = 1$$

$$x_i, y_j, x_{ik}, y_{kj}, z_{ij}^r \geq 0 \quad i, j, k, r$$

$$w_1, w_2 \geq 0$$

where we have used the following decision variables:

$x_i(y_i)$ = number of total outbound (inbound) flights, airport $i \in A$

x_{ik} = number of outbound flights from airport i to airport k

z_{ij}^r = number of flights over route r , airport i to airport j

y_{ki} = number of inbound flights from airport k to airport i .

Notation:

A = set of airports

(i,j) = feasible pair (linking airport i to airport j)

R_{ij} = all feasible routes for pair (ij)

L_i = outbound destinations for airport i

M_j = inbound airports for airport j

$A_1 \equiv \{(i,j) | i \in I \text{ and } j \in L_i\}$

c_{ij}^r = marginal transportation cost on route $r \in R_{ij}$

b_i = net traffic at airport i

$F(\bar{z})$ = convex nonseparable function, measuring risk

This single period planning model aggregates all individual flights between airport-pairs over the planning horizon. While extending the model to a large number of multiple periods will considerably increase the model's size, it would be, conceptually, straightforward to depict several periods. As such, the [OPT1] model can be used for supporting airport resource decisions, such as opening new runways and expanding the capabilities of control towers. The primary aims are to identify bottlenecks and to predict imbalances for the U.S. air-traffic system, given a variety of scenarios.

In this representation the optimization model takes a bi-criteria objective function consisting of risk and systemwide cost, with respective weights w_2 and w_1 . It should be stressed that the weights are used only as part of the sensitivity analyses to identify the efficient frontier and are not meant to be set a priori. Transportation cost is easily quantified in terms of traveled distance and fuel burn rates of the different aircraft models. A monetary value can also be assigned on factors like customer dissatisfaction due to flight delays and so on. Quantifying the risk component of the system, however, is a critical and controversial issue. At best we may consider optimizing a relative risk measure: the key idea is to compare some risk norm under different scenarios or system states, or with other similar systems. Odoni and Endoh [15] consider a probabilistic analysis of risk. In Section 5 we touch upon the issue of modeling risk as a function of congestion in the target sector during time intervals of interest.

The issue of identifying an acceptable point on the efficient risk-cost frontier is complex. It is obviously difficult for society to compare human risks and congestion delays against financial criteria. See, for example, Rowe [18]. Ultimately the solution depends upon societal tradeoffs and can only be derived through an informed political process. The network framework provides essential input data for this process by tracing out the efficient frontier; see Section 5.

The aggregate model [OPT1] is unable to provide adequate details for operational planning purposes.

Therefore, a second network model has been developed and an example is shown in figure 2. Here, decision variables monitor possible delays and alternative altitudes for every flight departing to or arriving at the airports of interest. The corresponding mathematical representation is defined below. In addition to optimizing transportation and delay costs this model limits the number of flights traversing regions of interest -- e.g., control sectors -- as a surrogate for minimizing risk. Including these aspects causes [OPT2] to be considerably larger than [OPT1].

$$[OPT2] \text{ Minimize } \sum_{f \in F} \sum_{p \in P_t} \sum_{t \in T_f} c_f^{pt} \cdot x_f^{pt} + \sum_{f \in F} \sum_{t \in T_f} c_f^t \cdot d_f^t + \sum_{a \in A} c^a \sum_{k \in K} c_k^a \cdot w_k^a$$

Subject to:

$$d_f^0 = 1 \quad \text{for all } f \in F$$

$$d_f^{t-1} = d_f^t + \sum_{p \in P_j} x_j^{pt} \quad \text{for all } f \in F, t \in T_f$$

$$\sum_{f \in D_k^a} x_f^{pt} = y_k^a + w_k^a \quad \text{for all } a \in A, k \in K$$

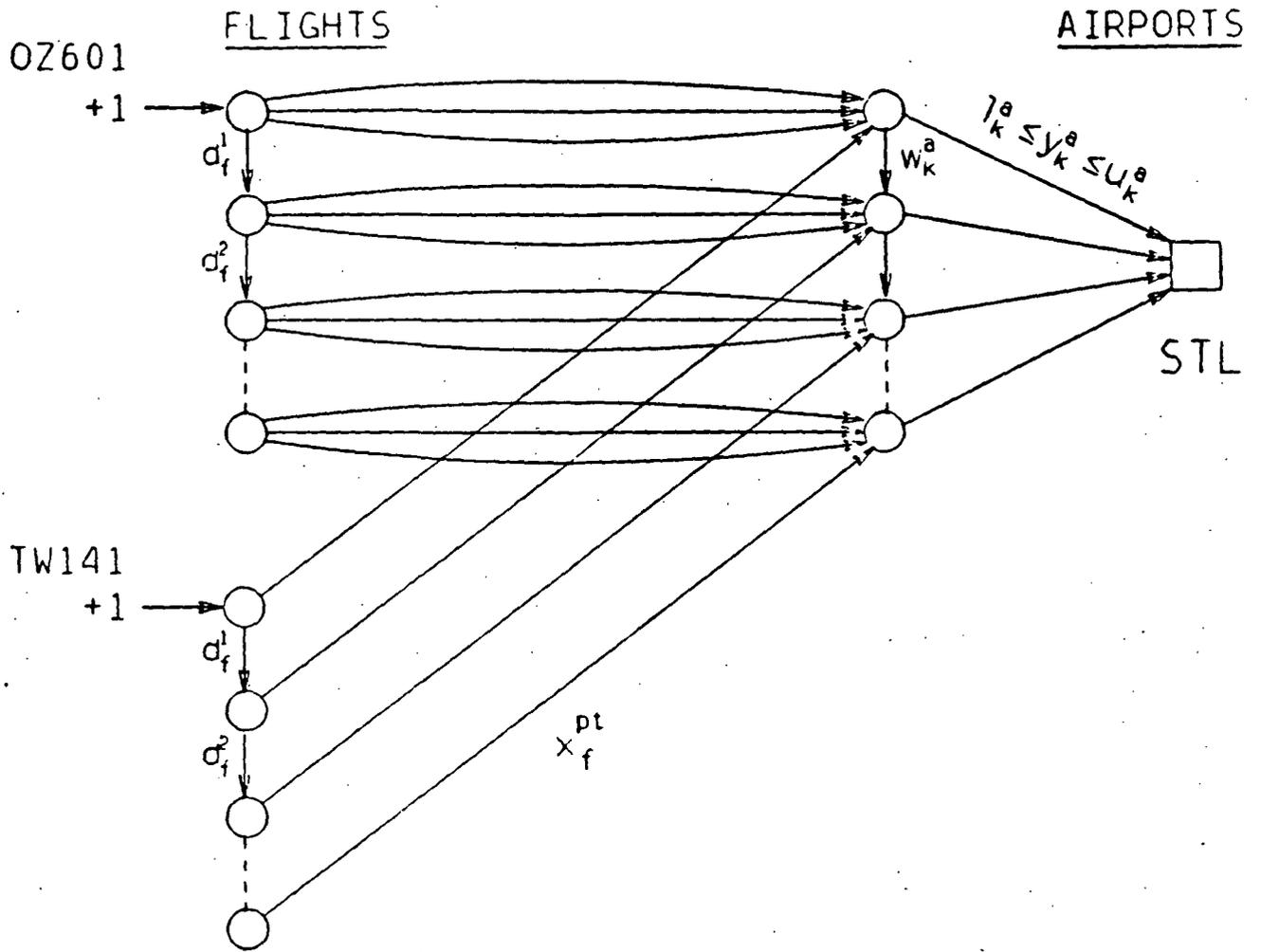


Figure 2: Example Air-traffic Network (Operational Model)

$$l_k^a \leq y_k^a \leq u_k^a \quad \text{for all } a \in A, k \in K$$

$$\sum_{x_f^{pt} \in R_r^{kf}} x_f^{pt} \leq L_r^k \quad \text{for all } r \in S, k \in K$$

where we have used the following decision variables:

- $x_f^{pt} = 0, 1$: flight f follows route p during time period t
 $d_f^t = 0, 1, t \in T_f$: flight f delays its departure during period t
 w_k^a for $k \in K$: number of planes delayed at destination airport a during period k
 y_k^a $k = 1, 2, \dots, K_{\max}$: number of planes landing at destination airport a during period k .

Notation:

- $a \in A$: set of airports
 $f \in F$: set of flights
 $p_f \in P_f$: set of possible routes for flight f
 $k = 1, 2, \dots, K_{\max}$: time of periods in planning horizon
 $t_f \in T_f$: set of possible time periods for departure of flight f
 R_r^k : set of flights traversing control sector r at period k
 D_k^a : set of flights whose destination is airport a , and arrival time is k
 c_f^{pt} : marginal transportation cost for flight f on route p during period t
 c_f^t : marginal departure delay cost for flight f during period t

c_k^a : marginal landing delay cost at airport a
during period k

$u_k^a(l_k^a)$: upper (lower) limit on the total number of
flights landing at airport a during period k

L_r^k : limit on the number of flights traversing control
sector r during period k

S : set of control sectors

This model controls individual flights: for every flight it specifies possible altitudes, routes to be followed and any departure/landing delays. The objective function incorporates the marginal cost of transportation, and associated delay costs.* Transportation risk is assessed by imposing a limit on the number of flights through a particular sector during every time interval of interest. As mentioned this formulation provides more details than [OPT1] since it controls flights on an individual basis. The added details result in a more complex model: the model is not only larger than [OPT1] but also it has integer variables. It is, however, equivalent to [OPT1] in the risk minimization aspects where congestion in an identified region provides a surrogate for risk. By varying the limit on the number of flights traversing a control section during every time period we may achieve the same result as by varying the weights in the multi-objective formulation in [OPT1]. Mathematically the first formulation is more tractable: it deals with a continuous nonlinear network model, while the second model deals with a linear multicommodity network problem with integer variables.

*Note that the model minimizes total transportation costs, following a utilitarian economies. If this approach adversely impacts individual carriers, the model can be adjusted through multiple weightings or by means of side payments between carriers (or the FAA).

The two models are expected to serve different purposes. The strategic planning model is intended to assess aggregate data, while the operational planning model deals with more microscopic aspects of air-traffic control.

3. Solution Algorithms

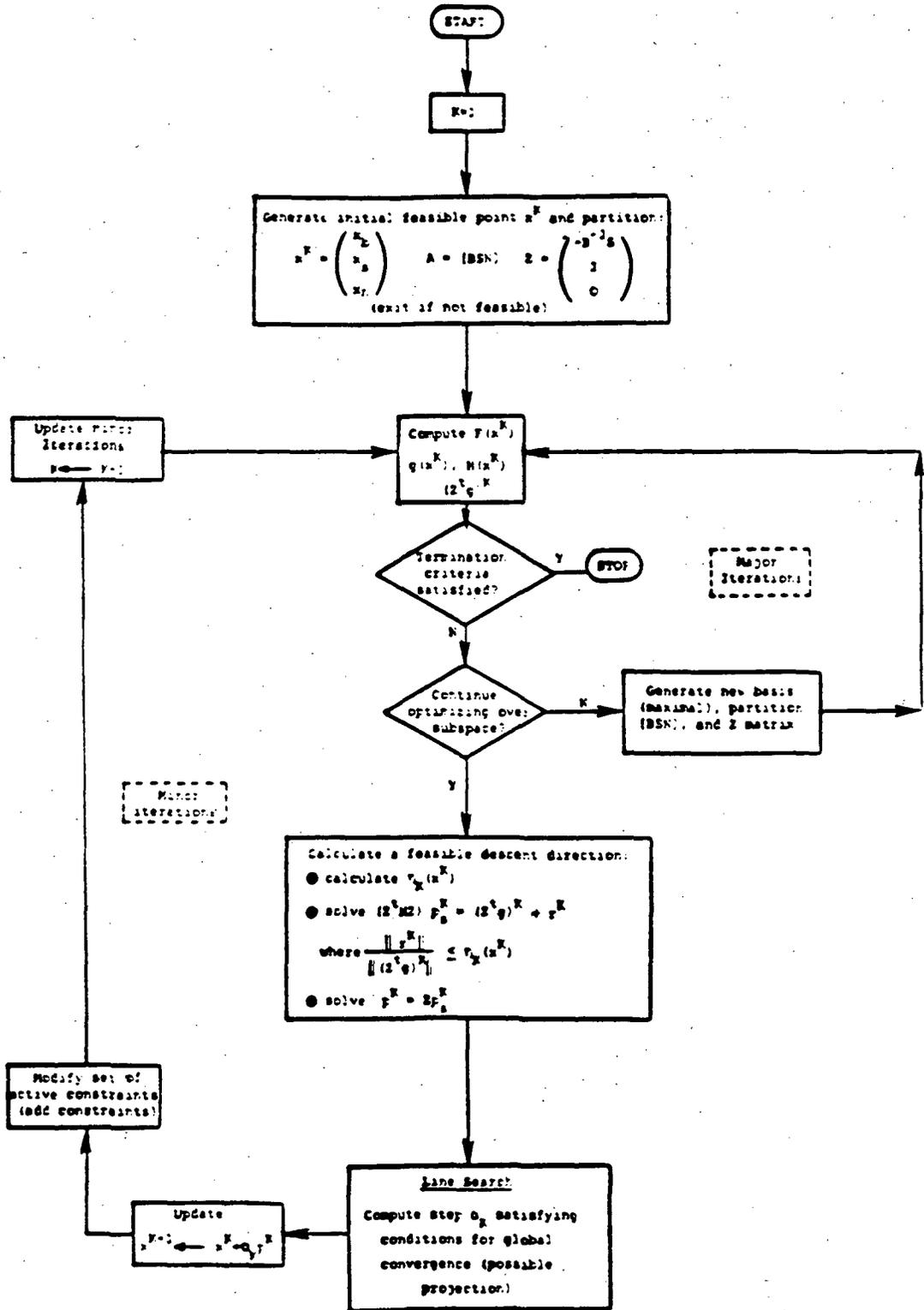
We describe here the main features of two nonlinear programming algorithms that have been specialized to solve [NLGN]. The methods are: (1) A second order algorithm based on truncated Newton search directions (TN), and (2) A first order algorithm based on the simplicial decomposition of the feasible region (SD). Both algorithms possess distinct characteristics: the truncated Newton algorithm can identify solutions to a high degree of accuracy; the simplicial decomposition algorithm can quickly converge to an approximate solution. The interested reader should refer to the papers by Ahlfeld et al. [2] and Mulvey et al. [13] for more detailed description of these algorithms and some computational results.

3.1 Truncated Newton Algorithm

In a manner similar to most nonlinear programming procedures, each (TN) iteration consists of two stages: (1) a search direction routine, and (2) a step length routine. Table 1 depicts the overall flow, in which the notation refers to the model [NLGN] presented in Section 2.

The search direction must fulfill certain essential features so that the overall algorithm will converge and so that performance efficiencies are attained. First, the direction must both maintain feasibility and point downhill (in a minimization context). Defining the search direction as \bar{p}^{-k} and given a feasible point \bar{x}^k at the k^{th} iteration, the usual Newton method for calculating \bar{p}^{-k} would solve the following quadratic programming problem:

Table 1: The Truncated Newton Algorithm



$$[QP] \quad \text{Minimize}_{\bar{p}^{-k}} \frac{1}{2} (\bar{p}^{-k})^t G(\bar{x}^{-k}) \bar{p}^{-k} + g(\bar{x}^{-k})^t \bar{p}^{-k}$$

Subject to:

$$\bar{A} \cdot \bar{p}^{-k} = 0$$

$$p_j^k \geq 0 \quad \text{if } x_j^k = l_j$$

$$p_j^k \leq 0 \quad \text{if } x_j^k = u_j$$

where

$g(\bar{x}^{-k})$ gradient of $F(\bar{x}^{-k})$ at \bar{x}^{-k}

$G(\bar{x}^{-k})$ Hessian of $F(\bar{x}^{-k})$ at \bar{x}^{-k}

By restricting our attention to a special projected matrix Z , whose columns form a basis for the null space of A , i.e., $A \cdot Z = 0$, the problem [QP] can be solved using the following two formulae:

$$\begin{aligned} (Z^t G Z) \bar{p}_s^{-k} &= Z^t g^{-k} \\ \bar{p}^{-k} &= Z^t \bar{p}_s^{-k} \end{aligned} \quad (3.1.i)$$

where

$$Z \equiv \begin{bmatrix} -B^{-1}S \\ S \\ N \end{bmatrix} \quad \begin{matrix} m \\ s \\ n-s-m \end{matrix}$$

G semi-positive approximation to the Hessian at point \bar{x}^{-k}

g^{-k} gradient of objective function at point \bar{x}^{-k}

and where the decision variables have been partitioned into three sets:

$$\bar{x} = [x_b | x_s | x_n]$$

$$A = [B | S | N]$$

$$g(\bar{x}) = [g_b | g_s | g_n]$$

$$\bar{p}^{-k} = [p_b | p_s | p_n]$$

The benefits of the Newton direction \bar{p}^{-k} are greatest in the neighborhood of a solution; however, it is expensive to calculate the solution of equation (3.1.1). In response, we adjust in a dynamic fashion the degree of accuracy of solving [QP]. A forcing sequence $\{\eta^k\} \rightarrow 0$ is employed in this regard. Accuracy is defined according to the relative residual in equation (3.1.1), $\frac{\|r^k\|}{\|Z_g^{t-k}\|}$, in which $r^k \equiv (Z^t G Z) \cdot \bar{p}_s^{-k} + Z_g^{t-k}$ and is a vector norm in R^n . The minor iteration (see table 1) continues only until the required accuracy is attained. Thus, $\frac{\|r^k\|}{\|Z_g^{t-k}\|} \leq \eta^k$ defines the termination criteria for the minor iterations.

When the algorithm is far from the solution the reduced gradient $\|Z_g^{t-k}\|$ -- is large and little work is required to locate a direction satisfying the acceptance criteria. Only the basic and super basic variables are optimized. If one of these variables hits a bound, the constraining variable is transferred into the set of nonbasics $[x_n]$. As $\|Z_g^{t-k}\|$ is reduced the acceptance criteria becomes more restrictive and the current solution to the direction finding problem lies closer to the Newton direction.

At this point, the nonbasic variables must be tested for optimality. First order estimates for the Lagrange multipliers are computed as follows:

$$\bar{u}_b^t = \bar{g}_b^t \cdot B^{-1}$$

$$\bar{u}_n^t = [\bar{g}_n^t - \bar{u}_b^t N]$$

$$\bar{u}_s^t = \bar{u}_b^t S$$

In this environment non-basic variables that reduce the objective function when moving away from their bounds (i.e., if $\mu_n^j < 0$ and $x_n^j = u_n^n$, or if $\mu_n^j > 0$ and $x_n^j = \ell_n^j$) along with free non-basic variables (i.e., $\ell_n^j < x_n^j < u_n^j$) are called eligible. Eligible variables are transferred to the superbasic set $[x_s]$ in conjunction with a maximal basis [4], and the TN algorithm continues the next major iteration with the new partition.

A sizable portion of the algorithm's execution time involves computing the search direction \bar{p}_s^k . While the truncated-Newton method can use any iterative method for solving equation (3.1.i), we have chosen the linear conjugate-gradient [CG] method. Although the reduced Hessian matrix $Z^t G Z$ is typically dense, the product required by [CG], $(Z^t G Z)\bar{v}$, is easily computable due to the sparsity of the large-scale components.

The success of the conjugate-gradient method depends upon locating a "good" search direction in a small number of iterations. Thus, preconditioning the reduced Hessian by the matrix P is important so as to reduce the number of CG iterations. Whereas the usual initial element of the CG sequence is \bar{g}_s^k , the vector $P \cdot \bar{g}_s^k$ becomes an initial element when preconditioning, where P is a positive-definite matrix. See [2] for further details.

3.2 Simplicial Decomposition Algorithm

While the TN algorithm is capable of solving large nonlinear network problems, we felt that a first order approximation would be better suited for ultra-large examples -- problems with more than 10,000 nodes and 100,000 arcs. The simplicial decomposition algorithm was selected by us to meet this goal. This algorithm is best examined in the context of general

decomposition methods. Following Geoffrion [6] we may place the algorithm in the class of "inner linearization/restriction" methods. The SD algorithm iterates between (1) solving a linearized subproblem and (2) solving a nonlinear master problem on a restricted space subject to non-negativity constraints. Table 2 depicts the overall flow. The algorithm was first presented in this form by von Hohenbalken [12,20]. Holloway [8] proposed the same algorithm, as an extension of the Frank-Wolfe method, and recently Lauphongpanich and Hearn [10] devised a restricted version for the traffic assignment problem. We have developed a version of SD specialized to handle [NLGN], whereby the master problem is solved inexactly [13].

The following theorem due to Caratheodory provides the necessary theoretical foundation for the algorithm.

Theorem:

Let $X \in R^n$ be a non-empty convex polytope. Then every $x \in X$ lies in the relative interior of one of a finite number of simplices whose vertices are extreme points of X .

See [19] for a proof. The main idea behind SD is simple in principle:

(1) First solve a linearized subproblem to get the extreme points of X .

We need not generate all the extreme points of the feasible region; this would result in a problem as complex as [NLGN]. Instead we generate extreme points as needed -- in a manner reminiscent of Dantzing's column generation method. A $(K-1)$ dimensional simplex is defined from K extreme points, and the search for the optimum is now restricted on the generated simplex.

ORIGINAL PAGE IS
OF POOR QUALITY

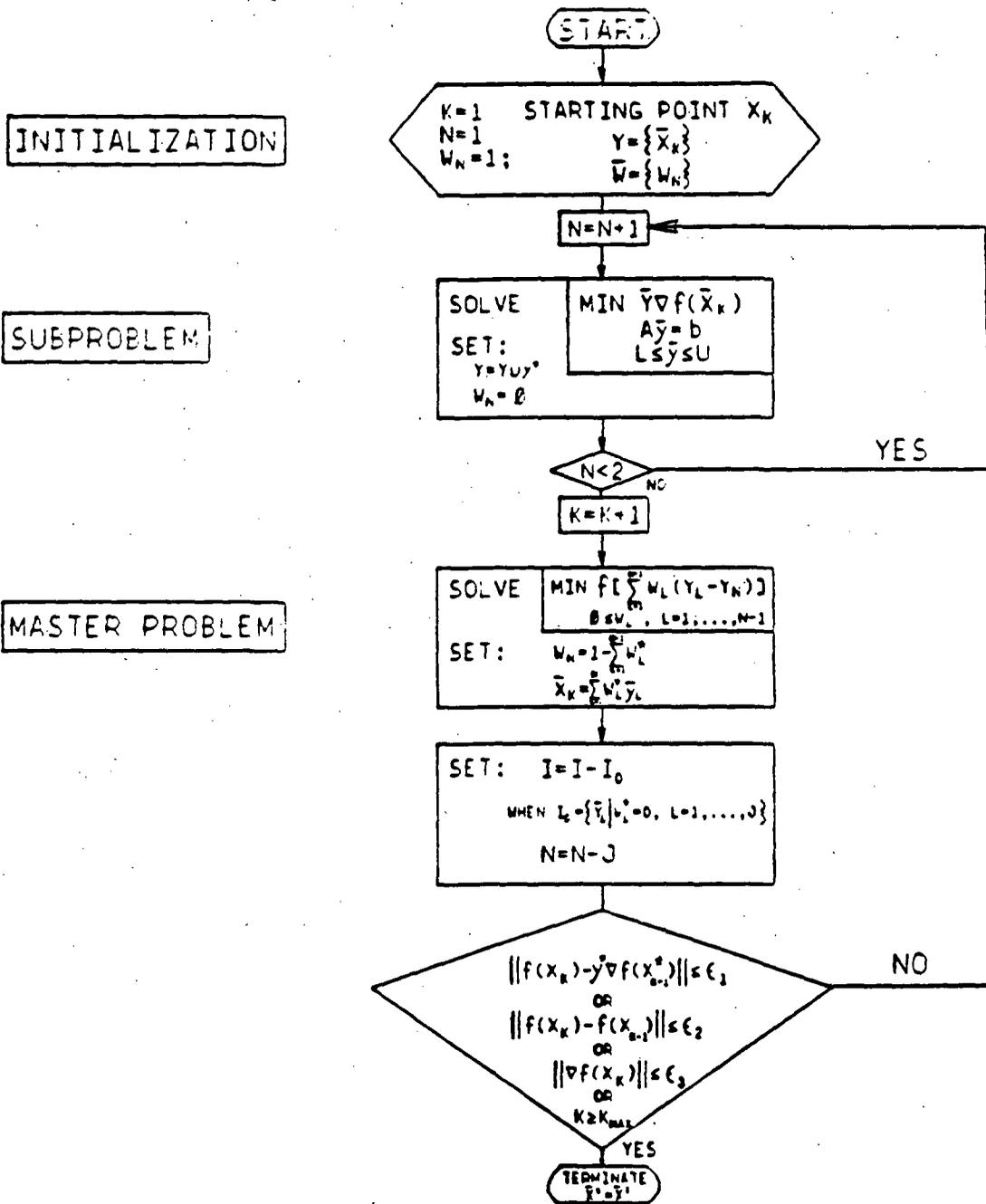


Table 2 : The Simplicial Decomposition Algorithm

(2) The master problem attempts to optimize the original function on the simplex generated by the subproblem. We have

$$\text{Minimize } F(\bar{w} \bar{y})$$

$$\bar{w}$$

Subject to:

$$\sum_{i=1}^K w_i = 1$$

$$0 \leq w_i \leq 1$$

where $\bar{y} = y_1, y_2, \dots, y_K$ is the set of generated extreme points and $\bar{w} = w_1, w_2, \dots, w_K$ are associated weights. We may reduce further the dimension of the master problem making use of the implicit function theorem:

$$\text{Minimize } F\left(\sum_{i=1}^{K-1} w_i (y_1 - y_K)\right)$$

Subject to:

$$0 \leq w_i \quad i = 1, \dots, K-1$$

Thus we have a nonlinear problem in dimension (K-1) subject to simple non-negativity constraints.

Reducing the master problem to a sequence of unrestricted (K-1) dimensional problems, this problem can be solved by any unconstrained algorithm. Refer to the work by Mulvey et al. [13] for computational experiments. The solution to the master problem is more important when the solution to [NLGN] lies in the current simplex. Thus we adjust in a dynamic fashion the accuracy in solving the master problem. Again a forcing sequence

$\{r^k\} \rightarrow 0$ is employed and the master problem terminates when $\|D^T \cdot g\| \leq r^k$ -- where D is the derived linear basis representing the current simplex. Once this degree of accuracy is achieved we return to the subproblem.

Simplicial decomposition provides us with a modular algorithm for solving [NLGN]. It converges rapidly to a good approximate solution, represented as a linear combination of a few extreme points ($K \ll n$). If high accuracy is required -- i.e., exact representation of the optimum solution -- then we need generate $n + 1$ extreme points and the master problem becomes as difficult as the original problem. In addition the subproblem preserves any special structure that may be present in the original problem. For [OPT] the subproblem iterations consists of solving a linear generalized network problem. Code LPNETG [12] was employed, modified to allow restarting from the basis of the last subproblem. The master problem was solved using a Quasi-Newton algorithm. Reference [13] provides further details for the implementation of SD and accompanying results from computational experiments.

4. Data Requirements

One of the most important aspects of modeling real world systems is the ability to collect the required data in a timely fashion. In the case of the Models [OPT], data may be classified in two categories: static and dynamic. By static we mean data that do not change over a long period of time (e.g., airport locations) and by dynamic we refer to data that change with time (e.g., flights scheduled), or with technological innovation (e.g., aircraft fuel burn rates, navigation systems and flight management). For the model to be useful the input data must be readily available. A key component of the comprehensive National Airspace System Plan is a centralized data-base of aircraft scheduled for, or actually flying the high altitude jet routes [7]. This data-base can serve on a real-time basis for updating the data required for [OPT].

For the prototype model developed the following data were required:

- (I) Airports information
- (II) Flights information
- (III) Fuel burn data.

Table 3 provides more details. Some data, like the airport coordinates, were available through sources used in the past by FAA, while other data had to be collected for the network model. The following sources were employed:

- (I) International Official Airlines Guide (IOAG) tape,
providing information about the airports
- (II) Flight Progress Strip data collected by the Control
Center, providing flights information
- (III) Fuel Burn Model developed by the FAA providing data
about fuel burn rates for different types of aircrafts.

ORIGINAL PAGE IS
OF POOR QUALITY

<ul style="list-style-type: none">• Airports Information1. Airport ID code2. Geographical coordinates
<ul style="list-style-type: none">• Flight Information1. Flight ID2. Origin airport3. Destination airport4. Cruise altitude on entering target sector5. Cruise altitude on exit from the target sector6. Time flight enters the target sector7. Time flight exits from the target sector8. Flight Hemi code defining legitimate cruise altitudes
<ul style="list-style-type: none">• Fuel Burn Data1. Aircraft type2. Fuel burn rate per hour for every legitimate cruise altitude3. Fuel burn rate per nautical mile for every legitimate cruise altitude

Table 3: Model data requirements

5. Modeling a Sector of the Indianapolis Control Center

A network model was built for the airspace controlled by a sector of the Indianapolis center. The purpose of the model is to serve as a prototype to illustrate the use of the optimization algorithms described earlier as well as the feasibility of the proposed model. Data were collected for a high traffic period on January 9, 1985 in which a total of 185 aircrafts crossed the sector over a 6-hour period. The duration of flight through the sector ranged from 4 to 23 minutes. Five distinct cruise altitudes above 29,000 feet (FL 290) were selected by the planes.

The model was built as a multi-period network as shown in figure 3.

The following provisions were made in the model:

- (I) Allow for delays at the origin airport, up to three 10-minute intervals and similar delays at the destination airports.

This time grid can be made finer by considering a larger number of progressively smaller delay intervals (e.g., six 5-minute intervals). The added accuracy will be balanced by the larger network that has to be solved.

- (II) Allow for every plane to follow one or two alternative cruise altitudes besides the one currently followed. Choice was restricted to the cruise altitudes one level above and one level below the primary altitude. Again, this restriction can be relaxed at the expense of generating larger network models -- the aircrafts could be instructed to follow any one of the four or five legitimate cruise altitudes, specified for the particular flight.

ORIGINAL PAGE IS
OF POOR QUALITY

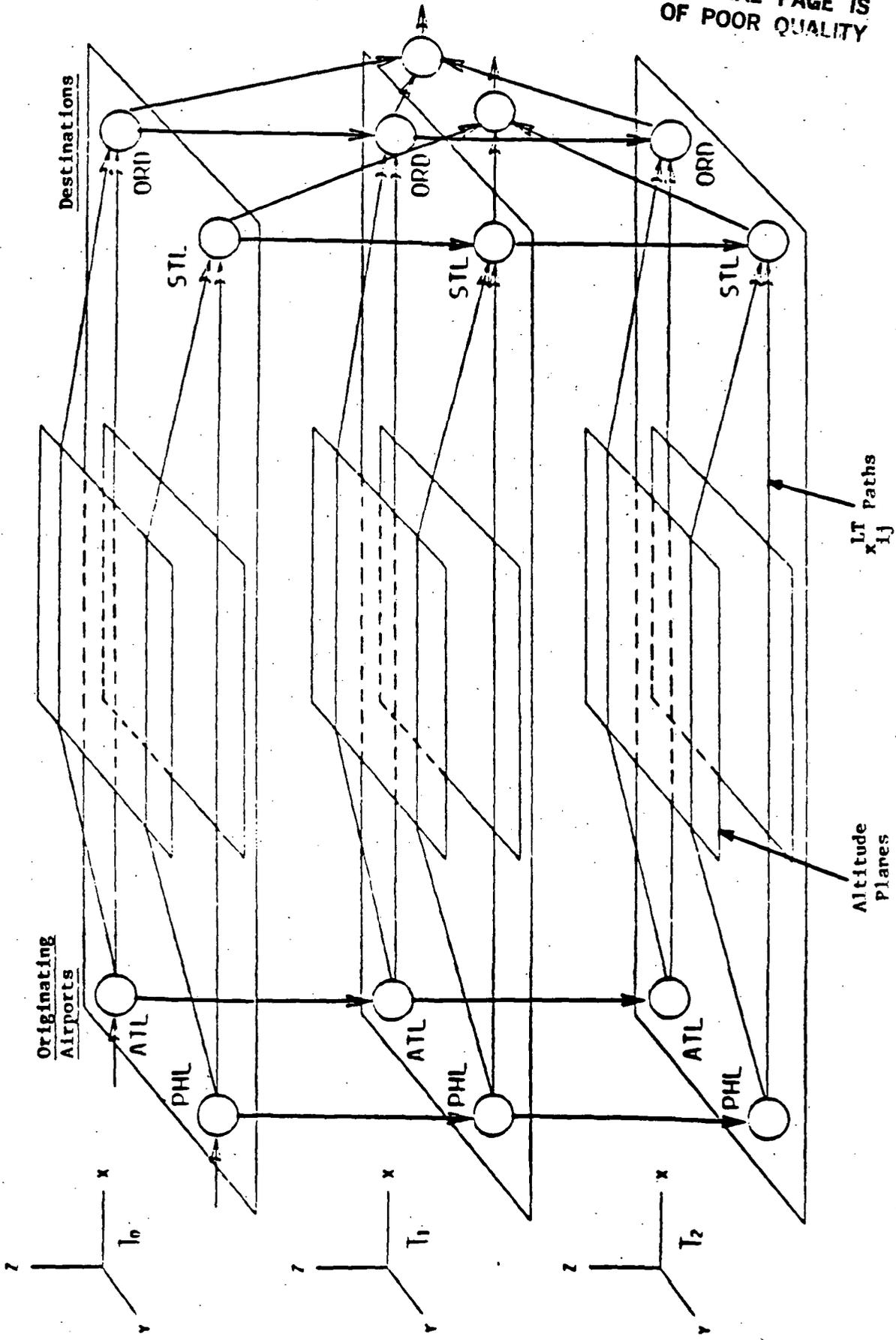


Figure 3 : Prototype Network Model for Risk/Cost Management

The model includes the dual objective of assessing risk and cost, as proposed in Section 2. Delay cost was considered as a function of fuel burn data. Other delay costs like crew salaries and a surrogate for customer dissatisfaction could also be included. The risk analysis was based on occupancy rates (congestion) at the same altitude during time intervals of interest.

We define the occupancy at level L during a period T as:

$$\alpha_{ij}^{LT} = \frac{\text{Time spent by aircraft on route } i \text{ on the same altitude (L) as aircraft on route } j \text{ during period T}}{\text{Time spent by aircraft on route } i \text{ altitude L during period T}}$$

The system-wide relative risk was then defined as:

$$\sum_T \sum_L \sum_{i,j} \alpha_{ij}^{LT} (x_i \cdot x_j)^2$$

where x_i, x_j indicate the number of flights on arcs i and j respectively. Summation was taken over all legitimate cruise altitudes (L), and over ten 36-minute intervals that cover the six hour planning period. Higher interactions -- between more than two planes -- were ignored. Figure 4 depicts the different phases of the modeling procedure.

Example:

- (1) If a single plane travels at altitude 39000 feet, and the plane is in the target section during the whole interval of $T = 30$ min., its contribution to the risk function is 0.
- (2) If a second aircraft flies at 30000 feet for $T = 30$ min., the corresponding risk coefficient is 1.
- (3) If the second plane stays in the target sector for 15 min., the risk coefficient is 0.5.

ORIGINAL PAGE IS
OF POOR QUALITY

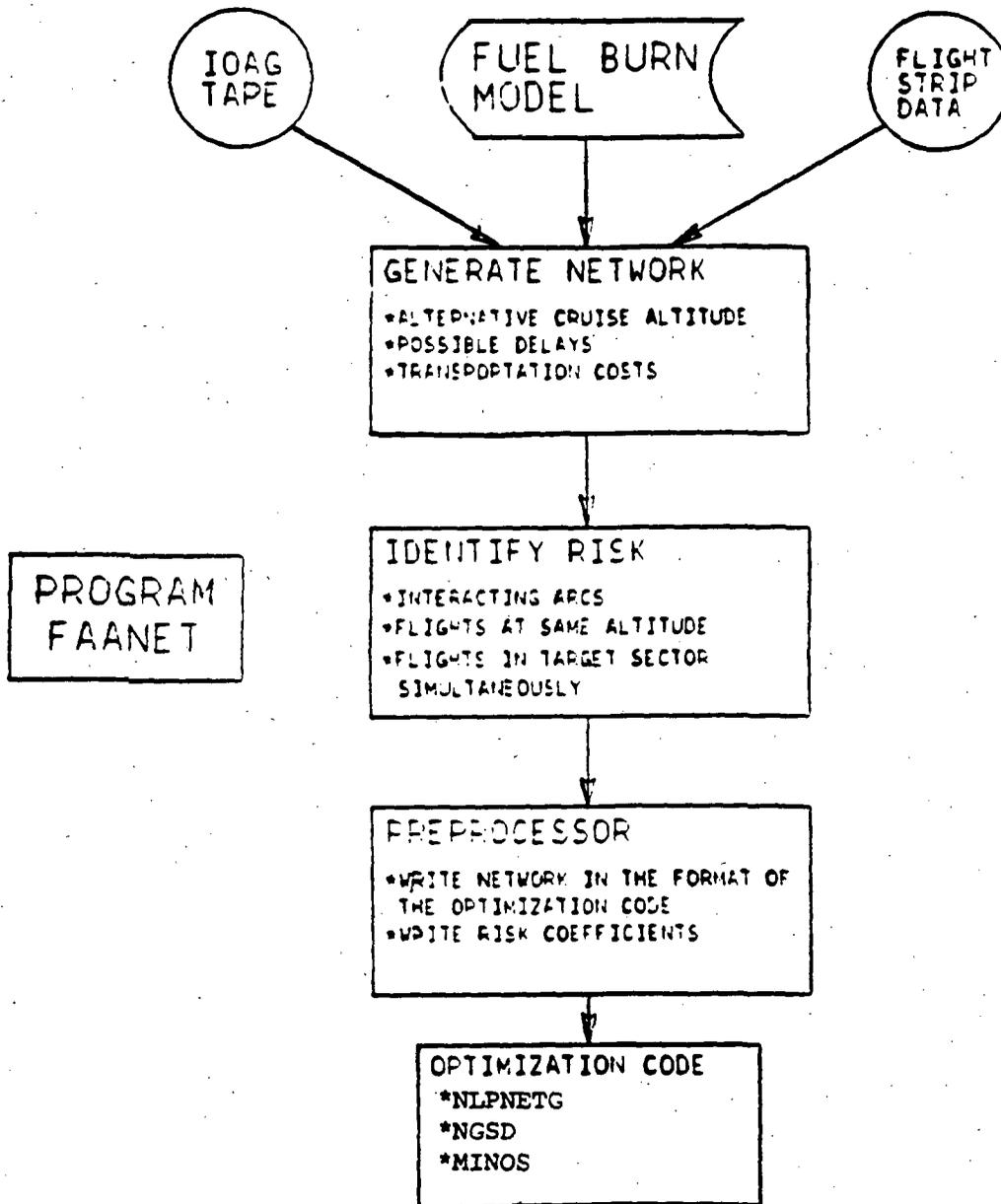


Figure 4: Modeling the Indianapolis Control Sector

This formulation provides for two degrees of freedom in the decision making process: delay the departure/arrival times in an optimal fashion and instruct aircraft to follow alternative altitudes to reduce congestion. The resulting network problem has 1295 nodes, 2873 arcs and approximately 15000 non-zero coefficients describing interacting flights. The test case was solved using code NLPNETG. By varying relative weights on the transportation and risk function in a systematic fashion the risk/cost efficient frontier was traced (figure 5). Again, the efficient frontier is not meant to serve as a direct way of comparing risk with cost. Instead it guides one in evaluating alternative modes of operation of the air-traffic control system, as generated by the model, or with currently followed procedures.

The major advantage of this methodology is that it generates a sequence of alternatives that are efficient; i.e., both risk and cost values cannot be improved simultaneously. This is easier to understand if we notice the location of point A in figure 5 -- this point was obtained by solving the optimization problem inexactly. From point A we may move to a series of alternative solutions for which the system is better off, both with respect to transportation cost and risk.

To study the effect of airplane congestion, we developed a histogram of all planes flying at a particular altitude, during the ten time intervals of interest. Figure 6 summarizes the results for three particular altitudes, before and after the optimization model was used. Note that as expected planes were diverted from a highly congested altitude (35000 ft.) to less congested routes (31000 ft. and 39000 ft.). This result was obtained with relative weights 0.5/9.5 on both risk and transportation costs.

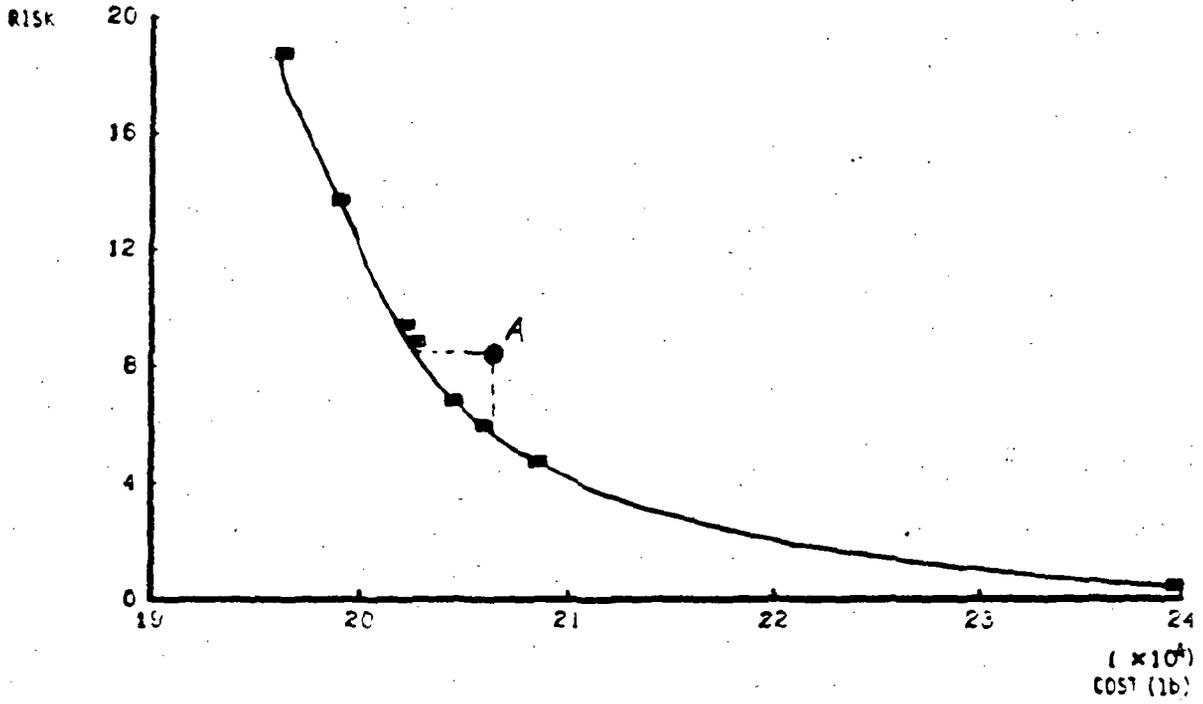


Figure 5: Efficient Frontier for Risk / Cost Tradeoffs

ORIGINAL PAGE IS
OF POOR QUALITY

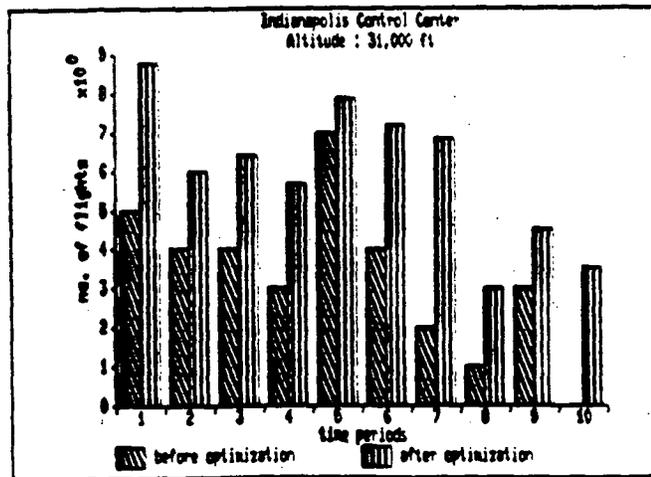
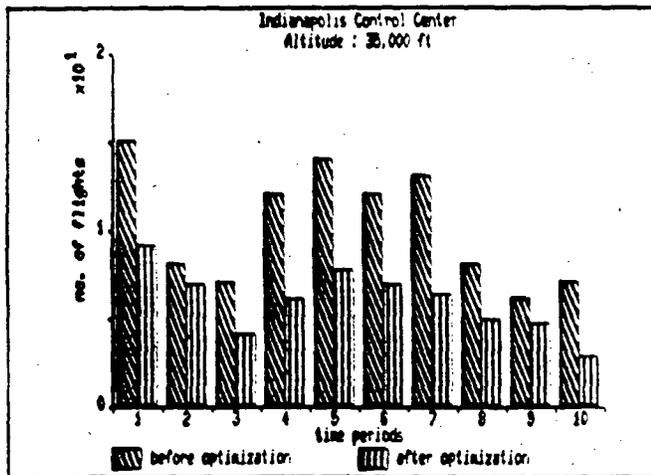
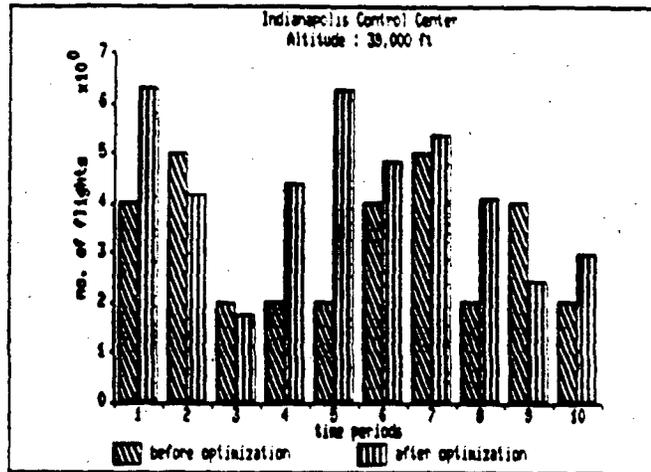


Figure 6: Airport Congestion Before and After Optimization

6. Computational Results

The FAA Control model can grow in size to challenge the capabilities of general purpose algorithms and conventional mainframe computer systems. The developed Indianapolis prototype consists of 1295 nodes and 2873 arcs. If instead we had chosen six 10-minute intervals for possible delays, and five alternative cruise altitudes the resulting problem would consist of 2405 nodes and 8510 arcs. If we consider simultaneously ten control sectors, with the same number of flights as the Indianapolis center, the network grows to approximately 20000 nodes and 80000 arcs.

To demonstrate the efficiency of the algorithms described in Section 3, we have solved several problems arising from a wide range of applications. Table 4 presents relevant information concerning the test problems. First, eighteen test problems were solved with the general purpose code MINOS [14]. The results from this program formed a benchmark against which to compare the truncated Newton -- code NLPNETG -- and the simplicial decomposition -- code NGSD -- algorithms. Problems that cannot be solved efficiently with the general purpose code can be solved with minimal computational resources using the specialized network algorithms. Tables 5 and 6 summarize the results. We observe that efficiency of the specialized network codes increases with problem size.

Advances in parallel processing computers are expected to ensure the feasibility of new applications. Specifically the air-traffic control model will benefit from the use of supercomputers in two domains:

- (i) Efficiently solve larger models that cover more than one control center and ultimately extent to cover the whole continental U.S. airspace, with finer time discretization
- (ii) Solve the operational planning model, under real-time conditions.

To illustrate the situation we have specialized NLPNETG for the CRAY-XMP vector computer at Boeing Computer Services. The optimizing compiler proved to be only marginally effective, due to the sparsity of the network problems. We had to analyze the algorithm in a way to take advantage of the pipeline features, and the presence of multiple vector functional units. Table 7 summarizes some of the results. We observe that much can be gained by specializing the network algorithm for the architecture of vector supercomputers. Comparisons with a VAX 11/750 and an IBM 3081 large mainframe are highlighted in table 8.

Finally the network model was solved for a range of relative weights, with code NLPNETG. We observe (table 9) that a complete analysis can be carried out within a few hours, even on a VAX minicomputer. Giving more emphasis to the risk function (nonlinear component) causes the problem to become, algorithmically, more difficult. This difficulty reflects the price we have to pay in going from a linear, transportation cost minimizing model, to a nonlinear model that incorporates the nonlinear form of risk minimization.

7. Conclusions and Future Research

We have discussed in this paper a general optimization planning model for the U.S. air-traffic system. The use of this model has been demonstrated using air-traffic data from a representative sector. By developing specialized algorithms, we were able to solve the resulting problem efficiently, thus demonstrating the feasibility of the approach for strategic planning. In addition, taking advantage of the latest technological developments in supercomputer design we were able to solve very large problems in a matter of seconds; thus, the network model can be used as an operational planning tool under real-time conditions.

This research has established that network models can be used as the basis for assessing some aspects of the U.S. airspace. The technology -- in terms of computer systems and algorithms -- is available, and the required data can be collected. Further research is needed, however, in modeling the air-traffic system. We have considered a deterministic model of risk. In general a stochastic analysis which also considers the systemwide optimization aspects would be more appropriate. While congestion was used as a risk surrogate, other criteria like expected number of aircraft conflicts, or expected number of controller intervention should be evaluated as alternatives. This is an area of potential interface between the optimization model described here and the probabilistic models of Odoni and Endoh [15].

Other aspects of the air-traffic control system can be examined within a network optimization framework, such as the flow control problem [1], or aircraft scheduling problems arising from aggregate solutions of the current

model. Issues like the uncertainty involved in determining projected demand for air transportation merit investigation, and the end effects due to the finite planning horizon have to be examined. As another extension, resource limitations can be imposed by introducing arc multipliers on traffic arriving/departing from an airport, thus controlling the total number of passengers an airport facility can handle.

While substantial progress has been made towards building and verifying the model, additional work needs to be done in model validation. Alternative strategies generated by the model have to be compared with existing methodologies to establish the correspondence of the model and its results to the perceived reality. This is another area of potential interface between the optimization model and the general probabilistic framework presented in Powell, Mulvey and Babu [16].

ORIGINAL PAGE IS
OF POOR QUALITY

PROBLEM	Size (Nodes/Arcs)	Free arcs at optimum	Condition No. of Reduced Hessian	Objective value	L_{∞} Norm of Reduced Gradient	Description
PTN30	30/ 46	15	$\geq 10^4$	-.3239322E5	.0045	Dallas Water Distribution models
PTN150	150/ 196	44	$\geq 10^4$	-.4819730E5	.0020	
PTN660	666/ 906	240	$\geq 10^6$	-.2061074E6	.0160	
SMBANK	64/ 117	54	$\geq 10^4$	-.7129290E7	.0003	Matrix balancing models
BIGBANK	1116/2230	946	$\geq 10^8$	-.4205693E7	.0004	
RANBANK	800/2000	13	$\geq 10^4$.3018788E6	.4025	
STICK1	209/ 454	246	$\geq 10^2$.6934392E1	.0001	Stick percolation models, electrical networks
STICK2	650/1412	763	$\geq 10^2$.3124563E1	.0001	
STICK3	782/1686	905	$\geq 10^2$.1117978E2	.0006	
STICK4	832/2264	1433	$\geq 10^2$.1566195E1	.0010	
GROUPlac	200/ 500	100	$\geq 10^3$.1011792E5	.0060	Randomly generated, strictly convex networks
GROUPlad	400/1000	119	$\geq 10^4$.3834884E5	.0049	
GROUPlac	800/2000	230	$\geq 10^5$.2265714E6	.4070	
MARK1	23/ 42	4	$\geq 10^2$	-.1145214E6	.0020	Markowitz portfolio construction
MARK2	53/ 102	10	$\geq 10^2$	-.3271077E6	.2420	
MARK3	857/1710	35	$\geq 10^2$.1150057E8	.0710	
FAA.TEST	28/ 64	6	$\geq 10^2$.7725644E5	.0500	Air traffic control model
FAA.IND3	1295/2873	787	$\geq 10^2$.2556325E1	.0501	

Table 4: Test Problems

Problem	NLPNETG Solution		MINOS Solution	
	Time (sec)	L_{∞} norm	Time (sec)	L_{∞} norm
PTN30	7.33	4.5E-3	44.08	8.8E-6
PTN150	23.86	2.0E-3	305.01	8.8E-6
PTN660	297.93	1.6E-2	9647.35	4.0E-4
SMBANK	21.50	3.0E-4	287.20	4.9E-4
BIGBANK	9100.00	4.0E-4	28800.00*	4.0E-1
RANBANK	245.37	4.0E-1	5477.97*	1.0E+2
STICK1	19.05	1.0E-4	12392.00	3.9E-6
STICK2	103.12	1.0E-4	**	
STICK3	73.75	6.0E-4	**	
STICK4	166.50	1.0E-3	**	
GROUPlac	1652.00	6.0E-3	2312.69	5.9E-2
GROUPlad	10227.00	4.9E-3	20347.00	5.9E-1
GROUPlac	7376.00	4.1E-1	16115.00	2.8E-1
MARK1	3.07	2.0E-3	12.23	1.3E-1
MARK2	21.60	2.4E-1	33.03	1.4E-1
MARK3	204.43	7.1E-2	1341.53	1.5E-1
FAA.TEST	1.24	5.0E-2	15.22	3.4E-9
FAA.IND3	707.00	5.0E-2	111600.00	2.8E-4

* Terminated when within 0.02% from optimum

** Did not converge after 8 hours

Table 5: Comparison of NLPNETG with MINOS

Problem	NGSD Solution		MINOS Solution	
	Time (sec)	% optimality	Time (sec)	l_{∞} norm
PTN30	13.25	0.02	44.08	8.8E-6
PTN150	83.90	0.50	305.01	8.8E-6
PTN660	590.35	3.50	9647.35	4.0E-4
RANBANK	245.37	4.0E-1	5477.97*	1.0E+2
STICK1	50.25	0.40	12392.00	3.9E-6
STICK2	517.48	0.94	**	
STICK3	1387.25	0.08	**	
STICK4	2688.72	0.15	**	
GROUPlac	314.85	1.60	2312.69	5.9E-2
GROUPlad	606.90	0.40	20347.00	5.9E-1
GROUPlac	395.38	0.52	16115.00	2.8E-1
MARK1	3.07	2.0E-3	12.23	1.3E-1
MARK2	21.60	2.4E-1	33.03	1.4E-1
MARK3	204.43	7.1E-2	1341.53	1.5E-1

* Terminated when within 0.02% from optimum

** Did not converge after 8 hours

Table 6 : Comparison of GNSD with MINOS

Problem	NLPNETG Solution times (sec)		
	Without vectorization	Compiler vectorization	User vectorization
PTN150	0.328	0.317	0.165
PTN660	2.435	2.316	1.402
SMBANK	0.358	0.327	0.177
BIGBANK	244.107	225.900	58.896
STICK4	*	*	2.925
GROUPlac	18.668	17.215	4.983
GROUPlac	*	*	49.454
MARK3	*	*	2.131

* Problem not solved with this option

Table 7 : Vectorization of NLPNETG on the CRAY/XMP

ORIGINAL PAGE IS
OF POOR QUALITY

Problem	NLPNETG Solution times (sec)		
	IBM 3081	VAX 11/750 (Unix)	CRAY/XMP
PTN150	1.37	23.86	0.165
PTN660	36.68	297.93	1.402
SMBANK	2.33	21.50	0.177
BIGBANK	884.53	9100.00	58.896
GROUP1ac	218.46	1652.00	4.983
GROUP1ac	1320.82	10227.00	49.454
MARK3	24.87	204.43	2.131
Average	281.19 (17)	3075.25 (184)	16.744 (1)

Table 8 : Testing NLPNETG on different computer systems

Weight (w_2)	Time (sec)	l_{∞} norm
1.0	1211	0.020
0.999999	707	0.054
0.999997	396	0.094
0.999995	396	0.150
0.999992	406	0.187
0.999990	367	0.467
0.999975	357	0.069
0.0	18	0.0

Table 9 : Tracing the efficient frontier with NLPNETG

References

- [1] *Proceedings of the Conference of Safety Issues in Air Traffic Systems Planning and Design*, Princeton University, Princeton, New Jersey, September, 1983.
- [2] D. P. Ahlfield, R. S. Dembo, J. M. Mulvey and S. A. Zenios, "Nonlinear Programming on Generalized Networks," Engineering Management Systems Working Paper EES-85-7, Princeton University, June, 1985.
- [3] P. Beck, L. Lasdon, M. Engquist, I. Balberg, N. Binnebaum and D. P. Bertsekas, "On the Goldsten-Levitin-Polyak Gradient Projection Method," *Physical Review*, Vol. 28, No. 7, pp. 3799-3812, 1983.
- [4] R. S. Dembo and J. G. Klincewicz, "Resolving Degeneracy Using a Maximal Basis," School of Organization and Management Working Paper Series B #63, July, 1983.
- [5] A. R. Ferguson and G. B. Dantzig, "The Allocation of Aircraft to Routes," *Management Science*, Vol. 3, pp. 45-73, 1957.
- [6] A. M. Geoffrion, "Elements of Large-Scale Mathematical Programming," *Management Science*, Vol. 16, No. 11, pp. 652-691, 1970.
- [7] J. L. Helms, "Implementing the National Airspace System Plan," Safety Issues in Air Traffic Systems Planning and Design, Princeton University Conference 155, September, 1983.
- [8] C. A. Holloway, "An Extension of the Frank-Wolfe Method of Feasible Directions," *Mathematical Programming*, Vol. 6, No. 1, pp. 14-27, 1974.
- [9] J. L. Kennington and R. V. Helgason, *Algorithms for Network Programming*, John Wiley and Sons, New York, 1980.
- [10] S. Lawphongpanich and D. W. Hearn, "Simplicial Decomposition of the Asymmetric Traffic Assignment Problem," *Transportation Research*, Vol. 18B, No. 2, pp. 123-133, 1984.
- [11] B. McDonald, unpublished report to the FAA, March, 1985.
- [12] J. M. Mulvey and S. Z. Zenios "Solving Large Scale Generalized Networks," *Journal of Information and Optimization Science*, Vol. 6, pp. 95-112, 1985.
- [13] J. M. Mulvey, S. A. Zenios and D. P. Ahlfield, "Simplicial Decomposition for Convex Generalized Networks," Engineering Management Systems Working Paper EES-85-8, Princeton University, 1985.
- [14] B. Murtagh and M. Saunders, "Large-Scale Linearly Constrained Optimization," *Mathematical Programming*, Vol. [bf 14], pp. 41-72, 1978.

- [15] A. Odoni and S. Endoh, "A General Model for Predicting the Frequency of Air Conflicts," *Safety Issues in Air Traffic Systems Planning and Design*, Princeton University Conference #115, September, 1983.
- [16] W. B. Powell, J. M. Mulvey and S. Babu, "Development and Validation of Models of Air Traffic Control Performance," Report, DOT DTFA 03-82-c-400046, Princeton University, September, 1984.
- [17] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, New Jersey, 1970.
- [18] W. D. Rowe, *An Anatomy of Risk*, John Wiley and Sons, New York, 1977.
- [19] B. von Hohenbalken, "A Finite Algorithm to Maximize Certain Pseudoconcave Functions on Polytopes," *Mathematical Programming*, Vol. 13, pp. 189-206, 1975.
- [20] B. von Hohenbalken, "Simplicial Decomposition in Nonlinear Programming Algorithms," *Mathematical Programming*, Vol. 13, pp. 49-68, 1977.