

AN OPTIMIZATION APPROACH TO ROUTING
AND SCHEDULING COMMUTER AIRCRAFT

by

Thomas A. White

Thesis submitted to the Graduate Faculty of
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Industrial Engineering and Operations Research

APPROVED:

D. M. Miller

J. W. Schmidt

J. A. Nachlas

August 1977
Blacksburg, Virginia

ACKNOWLEDGMENTS

The author would like to express his appreciation to the members of his committee, Drs. J. A. Nachlas and J. W. Schmidt and especially my advisor Dr. D. M. Miller for their assistance in preparing this thesis. I would also like to thank the National Aeronautics and Space Administration which provided funding for this effort under NASA Grant NGR-47-004-090 and technical guidance from Dr. T. Walsh. In addition I would like to thank Mr. J. W. Presburg of Pilgrim Airlines for his assistance in data collection. Finally, I would like to acknowledge my wife, Ro, for her understanding and support whenever I needed it.

TABLE OF CONTENTS

		<u>Page</u>
	Acknowledgments	ii
	List of Figures	vi
	List of Tables	vii
	Chapter	
1	INTRODUCTION	1
	Purpose of Study	2
	Problem Description	2
	Scope of Research	4
	Research Approach	4
	Results	5
	Thesis Organization	5
2	COMMUTER AIRLINES	7
3	THE ROUTING AND SCHEDULING PROBLEM . .	14
	Survey Results	14
	Routing and Scheduling Objectives . . .	15
	Scheduling Objectives	16
	Routing Objectives	17
	Summary	19
4	LITERATURE SEARCH	20
	Route Development	20
	Schedule development	23
	Route and Schedulement Development . .	26
5	MATHEMATICAL MODEL	28
	Problem Description	28

Chapter		<u>Page</u>
	Assumptions	30
	Scheduling Assumptions	30
	Routing Assumptions	30
	General Assumptions	32
	Mathematical Formulation	33
	The Objective Function	33
	The Penalty Constraints	35
	Route Feasibility Constraints	36
	Service Constraints	38
	Timing Constraints	38
	Mathematical Form of the Constraints	39
	Model Transformations	40
	Mathematical Model for the Afternoon System	43
	Summary	45
6	SOLUTION PROCEDURE	46
	Branch and Bound Methods	46
	General Description	46
	Application to Models (P^1) and (P)	47
	The Quadratic Programming Algorithm	48
	The Kuhn Tucker Conditions	48
	Beale's Algorithm	49
	Convexity	51
	Artificial Constraints	51
	Problem Simplifications	54

Chapter		<u>Page</u>
	The Passenger Choice Variables . .	54
	The Starting Point Variables . . .	55
	The Community-Connecting Variables	56
	Summary	57
7	APPLICATION	59
	Pilgrim Airlines	59
	Results	66
	Comparison of Solutions	69
	Sensitivity	70
	Conclusions	71
8	SUMMARY AND EXTENSIONS	72
	Extensions	75
	Cited References and Bibliography	76
	Appendix A	78
	Appendix B	83
	Appendix C	85
	Vita	160

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
2.1	Route Structure of Golden West Airlines	11
5.1	Passenger Choice Problem	37
6.1	An Example of an Artificial Constraint	52
7.1	Route Structure of Pilgrim Airlines . .	60
7.2	Subset of Pilgrim Airlines Route Structure	61

LIST OF TABLES

<u>Table</u>	<u>Page</u>
2.1 Type of Services Performed by Commuters	8
2.2 Airports and City Pairs Served by Commuters	9
7.1 Approximate Distance Matrix	62
7.2 Flight Times	63
7.3 Relative Demand as a Function of Time of Day	65
7.4 Solution to the Pilgrim Airlines Routing and Scheduling Problem	67
7.5 Approximate Pilgrim Airlines Time table	68
Appendix Table	
A.1 Relative Importance of Measures of Effectiveness	82

CHAPTER 1

INTRODUCTION

Currently, there are three major types of airlines serving the public sector of the United States. They are domestic trunks, local service carriers and commuters. Domestic trunks operate without federal subsidy payments and primarily serve long haul markets. Local service carriers also serve large population centers but may also serve smaller communities. The local carriers are given federal subsidy payments when operations are unprofitable on routes over which they hold certificates of public convenience and necessity. Commuter airlines are usually found in communities where the population cannot support timely service by the larger airlines. The Civil Aeronautics Board (CAB) has defined a commuter as an air taxi operator which satisfies one of two conditions [4]:

- (1) Performs five or more round trips between two or more points and publishes schedules stating the times, days of week and points between which these round trips are flown, or
- (2) Transports mail in accordance with a current contract with the U.S. Postal Service.

This paper addresses the problems of supplying timely service to small communities by commuters.

PURPOSE OF STUDY

Commuter airlines remain in business by carrying passengers, cargo, mail or a combination of these three. This research is directed toward those commuters which derive their primary income from passenger service or those which believe passenger preferences to be very important in developing routes and schedules. The purpose of this research is to develop a decision making aid useful to the managers of commuter airlines. By application of the methodology developed the airline will be better able to meet the demands of the customers, thereby increasing the demand for commuter airlines and increasing revenues.

PROBLEM DESCRIPTION

An airline "route" is a group of cities to be visited, in a specific order, by a single plane. A "schedule" is an assignment of a set of departure (and therefore arrival) times to each flight leg (i.e., path between two airports) of each route. The problem addressed here is to develop a procedure to find the optimal routes and schedules for the type of commuter discussed in the previous section.

The problem may be concisely stated as follows:

Design a set of routes and a corresponding schedule to minimize passenger inconvenience. Constraints on the system should limit the

operating costs, achieve a desired level of service, limit the number of intermediate stops on any route, and place the schedule within defined time limits.

The factors to be considered in modeling this problem are:

- (1) Passenger preferences for timing
- (2) Operating costs
- (3) Connecting schedules with domestic trunks and local service carriers
- (4) Level of service (i.e., frequency of stops) for each community
- (5) The number of intermediate stops on a route
- (6) Plane availability, and
- (7) Crew availability.

In solving the routing and scheduling problem, it is assumed that the following data is available: (a) the demand between city pairs as a function of time of day, (b) the travel time between city pairs, (c) the city pairs to be served, (d) the number of daily flights between each city pair, (e) the earliest time that a morning flight may be begin, and (f) the number of planes available to the commuter.

SCOPE OF RESEARCH

As mentioned previously, this research is restricted to those airlines which consider passenger preferences to be the main criteria in evaluating a set of routes and schedules. The research is further restricted to those commuter airlines which provide a feeder service into a single large hub airport from several outlying communities.

This study does not attempt to determine the demand between city pairs. It is assumed that this data can be estimated by the individual airlines. Also, the prospect of an airline purchasing additional aircraft is not considered.

RESEARCH APPROACH

Before developing a solution procedure it is necessary to concisely define the underlying decision problem. According to St. Mark [25] it is necessary to concentrate on certain types of commuter airlines because the scheduling and routing needs of commuters differ greatly. Hence, those commuters primarily transporting passengers are selected. A survey of commuter airlines believed to be this type is conducted to determine the objectives, constraints, and decision processes involved in determining routes and schedules.

After this data is gathered, the problem is formulated as a mathematical model. This model is transformed into

a mixed integer quadratic programming model. Taking advantage of the zero-one nature of the integer variables, a branch and bound method with bounds set by continuous valuable quadratic programming is used. The bounds are set using Beale's algorithm because of its efficiency [3] and its availability [12].

RESULTS

The mathematical model and solution procedure developed in this thesis are successfully applied to an operating commuter airline. An exact, globally optimal solution is found which satisfies all feasibility criteria. The solution is computed using a FORTRAN program developed for commuter airline routing and scheduling problems.

THESIS ORGANIZATION

The second chapter of this thesis deals with the background and general operating environment of commuter airlines. Chapter 3 discusses the problems of routing and scheduling as they relate to commuters. This chapter presents some conclusions drawn from a survey of commuter airlines. Chapter 4 presents a discussion of the literature available on route and schedule development.

The fifth chapter deals with the mathematical model development. A verbal description of the problem is fol-

lowed by a mathematical formulation. This formulation is then transformed into a mixed-integer quadratic programming model.

Chapter 6 deals with the solution procedure for the model. A discussion of branch and bound procedures is followed by a discussion of Beale's algorithm. Chapter 7 deals with an application of the solution procedure to Pilgrim Airlines, a commuter airline operating in New England. Finally, Chapter 8 provides a summary of the research effort, conclusions drawn, and proposed extensions.

CHAPTER 2

COMMUTER AIRLINES

In 1969, the CAB defined a "COMMUTER AIR CARRIER as an air taxi operator which (1) performs at least five round trips per week between two or more points and publishes flight schedules which specify the times, days of week and places between which such flights are performed, or (2) transports mail by air pursuant to a current contract with the (U.S. Postal Service)" [4] . Thus, commuter airlines may be classified as those which fly passengers, cargo and mail, passengers and cargo, passengers and mail, passengers only, cargo and mail, cargo only, and mail only. Tables 2.1 and 2.2 [4] provide the number of commuters as well as the number of airports and city pairs in each of the six classes.

Commuters are usually found in communities where the population density is not great enough to justify the expense involved in timely service by a local carrier or domestic trunk. Commuters extend scheduled air service to hundreds of communities that CAB certificated carriers choose not to serve or serve with poorly timed schedules [4]. The commuter airlines are able to operate profitably in these small markets because they operate smaller, more efficient aircraft.

TABLE 2.1
TYPE OF SERVICES PERFORMED
BY COMMUTERS

	Carriers	
	<u>No.</u>	<u>Percent</u>
Passenger, Cargo and Mail	49	22.1
Passenger and Cargo	83	37.4
Passenger and Mail	2	0.9
Passenger Only	28	12.6
Cargo and Mail	10	4.5
Cargo Only	27	12.2
Mail Only	<u>23</u>	<u>10.3</u>
	222	100%

TABLE 2.2
AIRPORTS AND CITY PAIRS
SERVED BY COMMUTERS

Airports Served
By Commuters

	Carriers	
	<u>No.</u>	<u>Percent</u>
Passenger, Cargo and Mail	198	27.3
Passenger and Cargo	268	37.0
Passenger and Mail	16	2.2
Passenger Only	85	11.7
Cargo and Mail	36	5.0
Cargo Only	45	6.2
Mail Only	<u>77</u>	<u>10.6</u>
	725	100%

City-Pairs Served
By Commuters

	<u>No.</u>	<u>Percent</u>
Passenger, Cargo and Mail	162	8.7
Passenger and Cargo	665	36.0
Passenger and Mail	25	1.3
Passenger Only	405	21.9
Cargo and Mail	55	3.0
Cargo Only	301	16.3
Mail Only	<u>236</u>	<u>12.8</u>
	1,849	100%

In general, commuters are regulated under a "laissez faire" policy by the CAB. That is, commuter ticket prices are not subject to CAB approval and the commuters may enter or leave any market. The decision to enter a market is based on the airlines management's evaluation of the potential return from such a venture. The decision to leave a market may be based on a present net loss from the market, or on a belief that greater returns will be achieved elsewhere.

The principle obligations of commuters are that they register with the CAB, carry the required passenger liability and personal property insurance, provide the CAB with copies of their schedules and fares, and file periodic reports covering their operations in the transportation of passengers, cargo and mail. In addition, some states place intrastate restrictions on commuter airlines. Such restrictions may concern routes as well as fares and other charges. These regulations affect only intrastate traffic [4].

Many commuter airlines are located in the vicinity of a large hub airport. The commuter's routes form spoke-like patterns with the hub as the center. These commuters primarily provide a feeder service into the hub from the surrounding communities. The commuter schedules are designed to meet the CAB certificated carriers at the hub to ensure that convenient connections are provided for passengers, mail, and cargo going to and from the hub

Key

- 0. Los Angeles
- 1. Santa Anna
- 2. Fullerton
- 3. San Bernadino
- 4. Ontario
- 5. Palmdale
- 6. Inyokern
- 7. Mojave
- 8. Santa Barbara
- 9. Oxnard

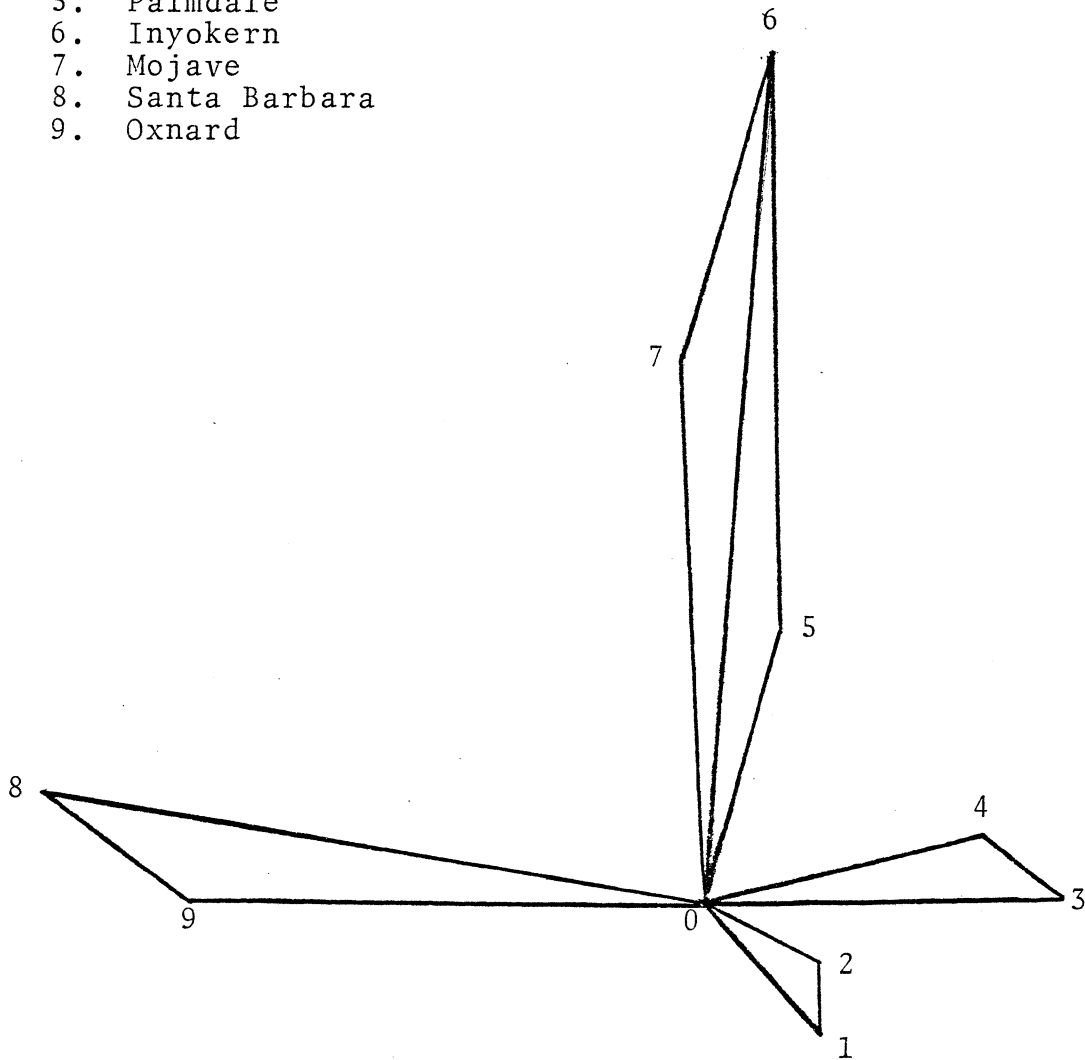


FIGURE 2.1
ROUTE STRUCTURE FOR GOLDEN
WEST AIRLINES

airport. See Figure 2.1 for an example of this type of commuter.

On the other hand, some commuters provide mainly intercity or origin-destination service. Such commuters take the passenger to his final destination as opposed to a connecting hub. Commuters will usually adapt the service they provide to the market they serve. Of the 1849 city-pairs served in 1974, 1257 had passenger service. Of the passengers about 65 percent were connecting and about 35 percent were origin-destination passengers [4].

In general, commuters will not be found in city pair markets which are being served by CAB certificated carriers. However, commuters may enter these markets if the CAB carrier's service is infrequent or poorly timed. In this case, the commuter may be able to generate sufficient revenue to make the market profitable. The fares of most commuters are about the same as those of the CAB carriers. Thus, convenience is the only advantage commuters offer passengers in these markets [24].

When local carriers are serving markets for which their aircraft are too large they may arrange for a commuter airline to assume service to the market. This is done only with the approval of the CAB and usually the approval of the affected local government. The CAB will usually act in the public interest to assure that the commuter's rates, fares and charges are unchanged, a specific

number of flights are offered per day, and that the local carrier will resume service to the affected markets if the commuter terminates service.

The commuter airline industry, although less than ten years old, is growing rapidly. From 1973 to 1974 the number of passengers flown increased by 14.3 percent while the tonnage of cargo increased by 29.9 percent [4]. Commuters carried 6,295,025 passengers in the year from June 30, 1973 to June 30, 1974. In spite of this sizeable growth there are no known quantitative tools available to commuter airlines which have been designed specifically to aid them in designing routes and schedules [25]. The needs of these airlines have generally been overlooked by those conducting research in favor of the needs of CAB certificated carriers.

CHAPTER 3

THE ROUTING AND SCHEDULING PROBLEM

SURVEY RESULTS

In order to more fully understand the routing and scheduling problems faced by commuter airlines, a survey of operating commuter airlines is necessary. A questionnaire concerning the relevance of various decision variables, objectives, and constraints was sent to 48 commuter airlines. A sample questionnaire is included in Appendix A. Twelve of the airlines responded to the survey.

From this survey, it is apparent that the timing of a schedule is the most important factor in evaluating the effectiveness of a schedule. Most airlines (10 of 12) feel that the timing of a schedule is either important or very important in increasing the market for commuter airlines. One respondent did not answer that particular question and another feels timing is fairly important but that "marketing, pricing, and aircraft size are equal factors." The airlines generally feel that it is important for the commuter schedule to minimize the time that a passenger must wait at a hub airport between connections with a CAB certificated carrier and a commuter airline. The degree of importance, however, is related to the percentage of connecting passengers. Pilgrim Airlines, which has

about 80 percent of its passengers connecting with CAB flights, feels that minimizing this delay is "vital on short, surface competitive routes." Air Florida Lines, with about 60 percent of its passengers connecting, feels that "except for those cities where there is no competition (the minimization of wait time at a hub airport) is the only thing that commuters have to promote." On the other hand, Royal Hawaiian Airways, with 22 percent connecting passenger, feels the minimization of delay at the hub is only moderately important.

Most of the airlines feel that the maximum number of intermediate stops on a route should be one or two. Five airlines responded that one should be the limit, four said two should be the limit, two said zero, and one said three to four should be the upper limit on intermediate stops.

One final observation from the survey is that "slotting" (i.e., airport officials allowing commuters to land and takeoff only during certain time slots) is not a problem. Eleven airlines responded that slotting is not a problem at the present time. Suburban Airlines said slotting is only a problem at Washington, D.C. (National).

ROUTING AND SCHEDULING OBJECTIVES

By combining the survey results with background literature it is possible to state and justify the general

objectives of commuter airlines in developing routes and schedules. The objectives stated in this section are used to develop the mathematical model in Chapter 5.

Scheduling Objectives

As determined from the survey results, the timing of a schedule is very important. If a potential passenger cannot fly at a time sufficiently close to his desired flight time, he will not utilize the commuter airline. For example consider the case where Air Wisconsin (a commuter) replaced North Central Airlines (a local service carrier) on a route from Appleton, Wisconsin, to Chicago, Illinois. During the first year of replacement, Air Wisconsin flew ten daily flights and 29,000 passengers. North Central Airlines had flown two daily flights and carried 6,630 passengers during the previous year. This is an increase of 337.5 percent [4].

The reader should keep in mind that examples such as this demonstrate the effect of a schedule only when all other factors remain constant. Factors such as advertising or changes in overall demand between communities have not been evaluated directly and are confounded with the schedule effect.

Since commuters operate primarily in the short haul, they compete with auto, bus and train transportation. Scheiner [18] discusses the competitive relationship between

short haul air transportation and automobile transportation. First, both means of travel stress convenience over economy. Secondly, with the auto no change of travel mode is required. Here, Scheiner is considering only intercity short haul traffic and not traffic connecting at an airport for some other final destination. In the case of connecting traffic, commuter airlines offer the advantage of a single mode of transportation. In any case, there is a competitive relationship between commuter airlines and automobiles. In certain areas such as the Northeast commuter airlines must also compete with trains.

According to Presburg [23] commuters usually will respect the territory of other commuter airlines and not engage in direct scheduling competition. Furthermore, commuters will usually avoid direct competition with CAB certificated carriers [4].

Commuter airlines also face problems relating to flight crews and ground support when designing schedules. Crews must work regular hours and be allowed sufficient time for meals and rest [23].

Routing Objectives

In developing routes, commuter airlines attempt to assign planes to a certain route and have that plane repeat that same route throughout the day in a manner similar to a limousine service vehicle [23]. Commuters also prefer

to minimize the distance traveled in connecting any group of cities; thus minimizing the direct operating costs for a given route. A plane will always move to a city that is closer to the hub than the last city visited. This is true on trips to the hub. On trips from the hub, the opposite is true. A plane is farther from the hub at the end of each successive flight leg.

Commuters wish to minimize the number of intermediate stops even if the time span between landing and takeoff is very short. This is because some travelers experience anxiety during landings and takeoffs. From the survey results it is determined that most airlines limit the number of intermediate stops on a route to one or two.

Commuters may also wish to preassign certain cities to certain routes. This may be to even the demand over the routes or it may reflect variations in fuel cost [23]. Airlines may have low cost fuel contracts with fuel suppliers at some but not all airports. In this case, the airline would want each route to include at least one stop at an airport where low cost fuel is available. Commuters may also attempt to group cities which experience similar weather together. The reason being that if delay is caused due to poor weather, a passenger will be more sympathetic to the airline's problem if his location also has poor weather.

SUMMARY

In developing routes and schedules commuters attempt to achieve many objectives. The survey results show that, to attract passengers, the airlines must provide timely service. The airlines also attempt to minimize direct operating costs. The survey results combined with information from Presburg [23] and St. Mark [25] are used to develop the mathematical models of the routing and scheduling problem presented in Chapter 5.

CHAPTER 4

LITERATURE SEARCH

Recall that, a route is defined as a group of cities, communities or stops served in a specific order by a single plane (or any vehicle in the more general case). A schedule is an assignment of a specific time(s) that the plane is to stop at each community on the route. This chapter deals with past efforts in the area of route and schedule development.

ROUTE DEVELOPMENT

This section surveys the literature available on route development alone. Following sections deal with literature available on schedule development and simultaneous route and schedule development.

Ferguson and Dantzig [6] develop an algorithm to assign aircraft to routes. The basic problem is to determine the number of aircraft of each type to assign to each route. The objective is to minimize the operating costs of aircraft and the cost of lost revenue when not enough aircraft are assigned to a route to meet the demand on that route.

In this problem x_{ij} denotes the quantity of the i^{th} type of aircraft assigned to the j^{th} route. The total

demand on the j^{th} route is d_j and the total number of the i^{th} type of aircraft is a_i . The number of passengers that may be carried by the i^{th} type of aircraft on the j^{th} route is denoted by p_{ij} . The cost of a flight is represented by c_{ij} . The mathematical formulation is as follows:

$$\min_{\bar{x}} z = \sum_i \sum_j c_{ij} x_{ij}$$

$$\text{s.t. } \sum_j x_{ij} = a_i \quad \text{for all } i$$

$$\sum_i p_{ij} x_{ij} = d_j \quad \text{for all } j$$

$$x_{ij} \geq 0 \quad \text{for all } i, j$$

Ferguson and Dantzig further develop their model in [7] to solve problems where demand is uncertain but follows a known discrete probability distribution. A similar model is developed for this case.

Actually, the work of Ferguson and Dantzig is not concerned with the development of routes. It is more of an assignment methodology once the routes have been determined.

Gagnon [8] develops a "Passenger Allocation Model" to allocate a forecast number of passengers traveling from any point to any other point to the various paths offered to them. This process involves three major steps. The

first step uses an "attractive path generator." A path is attractive if no other path will lead to the destination in less time and the flight is scheduled at a popular time of day. The second step involves a "traffic distribution estimator." This step consists of evaluating passenger preferences concerning time of day and day of week to obtain the volume of passengers wishing to travel during each half hour of each day. The third step involves input from the first two steps. This data is transformed under allocation rules into a final allocation of passengers to routes. Gagnon does not develop routes or schedules, but rather finds the demand for routes once they are determined.

Barcum et. al. develop a model in [1] to determine the optimal number of available seats that an airline should provide in any market to maximize profits. A non-linear relationship is developed between seat share (i.e., the percentage of the total seats available that one airline provides) and market share (i.e., the percentage of all seats purchased that passengers purchase with one airline). By relating load factors to market share and seats flown it is possible to develop equations to optimize the number of seats to fly. Barcum, like Ferguson and Dantzig, is concerned with how many planes to fly on each route after the routes have been determined.

Richardson [15] develops and solves a mixed integer linear programming model to maximize profits over a long

haul market from Sidney to London. Each route is broken into sectors with the requirement that each plane must stop at the end of each sector. In addition, a plane may stop at intermediate points between sector ends.

Richardson's method for forcing aircraft to make connected trips is used (with modifications) in developing the routes for the commuter airline routing and scheduling problem, model (P), in Chapter 5.

SCHEDULE DEVELOPMENT

This section discusses literature available on the problem of scheduling aircraft (or other vehicles). The routes are assumed to already be known.

Salzborn develops an optimization procedure in [16], for scheduling trains on a single double track railway line without branches. The problem is that, given a single path with multiple stops and multiple trains (of equal capacity), which train is to make what stops and in what order are the trains to depart. Algorithms are developed for two objectives and a combination. The first objective is to minimize the overall number of intermediate stops passengers have to be delayed by. The second is to minimize the total number of carriage miles. All problems are solved using dynamic programming.

Nemhauser [14] solves a train scheduling problem between two points with dynamic programming. The objective

is to maximize the difference between income and expenses. This is done by assigning express service (non-stop) or local service (intermediate stops) to discrete departure times. A passenger whose origin or destination is an intermediate point must use a local train. Passengers traveling between end points may use either express or local service. A passenger may choose local service when traveling between end points even though the service takes longer if the train arrives at the final destination at a time more convenient for the traveler.

Salzborn [17] also develops a method of scheduling buses on a route from the central business district of a city to the suburbs. The objective is to determine the bus departure rate (which varies over time) which will minimize the passenger waiting time subject to constraints which are necessary and sufficient conditions for the minimum fleet size.

Barnett and Kleitman [2] investigate the problem of scheduling service with one vehicle and several service points. The objective is to minimize the expected waiting time for users of the service. A constant arrival rate and infinite vehicle capacity are assumed. The travel times between service points are random variables and the holding times between consecutive routes are the decision variables. An important conclusions from this research is that very little improvement results from holding the

vehicles for any specific amount of time. It is almost as effective to have the bus begin its next route as soon as it finishes its present route.

Miller [13] develops a scheduling algorithm for aircraft that considers the time of day during with a flight is made. However, since the algorithm is developed for only one city pair it has little applicability to real-world scheduling problems. In this procedure the day is broken into 2.5 hour periods. Departures are permitted only at the beginning of each period. The decision variables are the number of each type of aircraft to allow to leave at the beginning of each period. It is assumed that every plane can make a round trip in 2.5 hours but will not try to make more than one trip in any period. Hence, the decision to schedule a plane in one period has no effect on scheduling the plane in other periods. The objective is to maximize the revenue plus a measure of consumer satisfaction minus the direct, indirect and capital costs. This problem is solved using IBM's mixed integer code BBMIP.

Salzborn, Barnett, Kleitman, Nemhauser, and Miller have developed scheduling algorithms but by the nature of their particular applications have not had to address themselves to the scheduling problem simultaneously with the problem of routing.

ROUTE AND SCHEDULE DEVELOPMENT

This section deals with simultaneous route and schedule development. This literature deals with problems, of CAB certificated carriers, which are analogous to the commuter problem undertaken in this thesis.

Young [21] uses a dynamic programming model to develop a timetable for common-carrier transportation systems. The objective function to be maximized is a combination of revenues, costs and passenger willingness to pay. The method of solution is to find a currently optimal solution for each vehicle. Each vehicle timetable is designed to improve the overall fleet effectiveness. After all fleet vehicles have been given a timetable passenger allocations to vehicles are modified. At this point a new currently optimal timetable for each vehicle is again computed with each new timetable improving system effectiveness. Passenger allocations are again modified. These iterations continue until the overall fleet timetable repeats itself. Solutions obtained are optimal within a "first order passenger exchange." However, there is the possibility that a three or four way exchange of passengers between vehicles will lead to a better solution.

Hyman and Gordon develop a method, in [10], of routing and scheduling aircraft, given the route structure of an airline, the number and type of aircraft, passenger preferences, frequency of service on each route, management

policies, passengers on each route, and maintenance requirements. Initially, an aircraft is assigned to a starting city and a departure time. Dynamic programming is used to determine the best flight time if only one flight is to be made. Next, the program iterates to determine the best routine if two flights are to be made. This continues until the number of flights exhausts the available time during a day. All possible chains (group of connected cities) are compared to determine the one which provides the maximum return. An important assumption is that the optimal solution for an entire fleet of aircraft scheduled simultaneously will be in the neighborhood of the solution found if the planes are scheduled sequentially (plane one is scheduled, then plane two is scheduled, etc.).

A major disadvantage of the methods of Hyman, Gordon and Young is that these methods do not guarantee a global optimum. In addition the commuter airline systems studied in this paper must have routes with a predetermined stopping point (the hub airport) rather than a predetermined starting point as does the model developed by Hyman and Gordon.

The next chapter presents the mathematical formulation of the commuter airline routing and scheduling problem. Following chapters present the solution procedure and an application to an actual operating commuter.

CHAPTER 5

MATHEMATICAL MODEL

PROBLEM DESCRIPTION

The mathematical model presented in this chapter is designed to represent a system which has a majority of passengers connecting with CAB certificated carriers at a single hub airport. The model does not, in general, apply to systems which have a large percentage of origin-destination passengers. The model can represent origin-destination traffic from the outlying communities to the hub by representing the time the passengers desire to arrive at or depart from the hub as departure or arrival times of fictitious CAB carriers. The failure of the model to represent origin-destination traffic between the outlying communities does limit the applicability of the model but not severely in that there are many commuters which primarily are a feeder service. Approximately two-thirds of all commuter passengers connect to or from other flights [4].

During the morning hours, there are relatively few CAB certificated carriers bringing passengers into a hub to connect with the commuters [23]. This is mainly because most travelers do not wish to fly in the early morning hours. For instance, if a passenger wants to connect to

a commuter at Laguardia Airport at 9:00 a.m. from a CAB carrier, the CAB carrier must arrive at Laguardia by 8:15 a.m. (Local airport authorities specify a minimum slack time between connections.) This means that the passenger will have to begin the CAB flight at a very early hour if the CAB flight takes more than a few minutes. For this reason, most travelers prefer to arrive at the hub the night before or take a CAB flight that originates in mid-morning and arrives at its destination (the hub) later in the day. Another explanation for the existence of this one way flow in the morning is the general trend of traffic into large cities in the morning.

Because of this one way traffic flow, the schedule for the morning will be concerned only with traffic into the hub from the outlying communities. During the afternoon hours passengers arrive at the hub and want to use commuters to reach the outlying communities. In addition, there are still passengers in the outlying communities that want to reach the hub to connect with afternoon and evening CAB flights. For this reason, separate models have been developed for the morning and afternoon. The afternoon model is an extension of the morning model and slightly more complex. The afternoon model is, in fact, a generalization of the morning model. This more generalized model may be used with either a morning or afternoon system.

ASSUMPTIONS

Scheduling Assumptions

In developing the timing of the schedule it is assumed that once a plane begins a trip it will complete that trip as rapidly as possible. That is, the plane will only stop at intermediate airports long enough for the passengers to embark and debark, refuel, if necessary, and perform any other needed operations. Normally, there will no slack or hold time at the intermediate airports. The schedule is developed with this assumption because time is so critical. Commuters want to provide convenient connect times for their passenger but also want to provide fast service to the hub. A passenger will not continue to use a commuter airline on a short trip if he must wait long periods of time at intermediate cities. It is understood that traffic and weather conditions may cause unavoidable delays. This, however, will be placed into the schedule by increasing the time allotted for flight between cities. If this is done, the aircraft may have to wait at airports along the route when traffic is light and/or weather is good to avoid premature departures.

Routing Assumptions

One principal assumption concerning route development is that a plane, once assigned to a route, will repeat that route over and over during the day. This assumption

is based on industry provided information [23]. Another assumption is that if a plane is flying into a hub from an outlying community it will not make an intermediate stop at any city whose Euclidean distance from the hub is more than the Euclidean distance to the hub from the last city visited. For example, if the distance from city A to the hub is ten miles while the distance from city B to the hub is five miles then a plane may fly from city A to city B to the hub but not city B to city A to the hub. This applies on trips to the hub only. The reverse is true on trips from the hub to the outlying communities. It is assumed that all inbound trips terminate at the hub and all outbound trips originate at the hub and terminate elsewhere. These assumptions force any group of cities in a single route to be visited in the order which will minimize the total distance flown (and thus the order which will tend to minimize direct operating costs).

In addition, a plane will retrace its steps from the hub before it begins its next route. For example, if a plane flies from city A to city B to the hub it will fly from the hub to city B to city A and then at the proper time fly from city A to city B to the hub again if another trip into the hub is desired.

One final assumption is that one community is pre-assigned to each route. This is desirable for several reasons. First, it will reduce the probability of a search

procedure finding alternate optimal solutions which are really identical. Suppose the optimal solution consists of cities A and B comprising route one with hub arrival times of 8:00 and 10:00 and cities C and D comprising route two with hub arrival times of 7:30 and 9:00. Then an alternate optimal is cities C and D comprising route one with hub arrival times of 7:30 and 9:00 and cities A and B comprising route two with hub arrival times of 8:00 and 10:00. However, if one preassigns city A to route one and D to route two, the search procedure will not waste time searching for a solution it has already found. Secondly, this assumption allows one some control in leveling demand among the routes. Finally, this may permit the commuter to force the route to contain a community which has a low cost fuel agreement with the commuter.

General Assumptions

Concerning the size of planes, no assumptions are necessary. One may determine the routes and if all planes are not identical assign the larger planes to those routes with the greatest expected demand. Likewise, if all crews cannot fly all planes then care must be taken to assign the proper crew to each plane. Also, a reasonable limit must be placed on the number of hours that a crew can work in a day. Finally, it is assumed that a traveler will be able to purchase a ticket for the commuter flight of his choice.

MATHEMATICAL FORMULATION

The mathematical formulation for the morning system consists of an objective function and ten sets of constraints. The original formulation is later transformed into a mixed integer quadratic programming model. Finally, a mathematical model for the afternoon system is presented.

The Objective Function

The objective function is designed to minimize a weighted measure of the time that passengers wait at the hub between connections. The constant "c" is used as a lateness penalty and should be set relative to a penalty of one for a unit of time early. For example, if commuter airline executives feel that a commuter plane one minute late for a CAB flight is as equally undesirable as a commuter plane ten minutes early for that same CAB flight then "c" should be set to ten. In the opinion of the author "c" should normally be a large number. If a commuter flight arrives after a CAB flight then there is almost no chance that a passenger will fly with the commuter knowing he will miss his connection. It is conceivable that this would happen, however, because of airport regulations. For instance, at JFK International Airport travel agents may not book passengers on connecting flights when the time between connections is less than 45 minutes. Thus, if passengers on a commuter flight wish to connect with

an 11:30 a.m. CAB flight out of the hub they must arrive at the hub by 10:45 a.m., if they purchase all of the tickets from a single travel agent in advance. If the commuter is scheduled to arrive at 11:00 a.m., it is too late for most passengers. However, suppose a traveler purchases one ticket for the commuter from one source (say the commuter airline) and a ticket for the CAB flight from another source (say a travel agent). He may purchase any tickets he chooses and thus may use an 11:00 a.m. commuter flight to connect with an 11:30 a.m. CAB flight.

In addition to the weighting of earliness versus lateness, there is a set of weights relating the routes to the CAB flights. For each route there is a set of CAB flights that passengers in the city preassigned to that route wish to connect with. A weight is applied to each combination of route and CAB flight provided there are passengers on that route connecting with the CAB flight. The weights will be determined by the commuter airline's management to reflect the relative importance of providing close connect times between the plane on the route and the CAB flight.

One method of determining these weights is to define the weights to be the average number of passengers connecting with each CAB flight. The main problem here is that the commuter airlines may not know which flights the passengers are connecting with at the hub. However, the commuters

usually have a general knowledge of the relative importance of arrival times at the hub. In this case, they could group CAB flights leaving the hub at approximately the same time together and qualitatively determine the weights based on time of day without considering the CAB flights individually.

To model the objective function, let

w_{ip} = the weight of the p^{th} CAB certificated carrier flight for route i ,

c = the relative weight of a unit of time late compared to a unit of time early,

d_{pi} = the departure time of the CAB certificated carrier flight p for route i ,

t_{ij} = the time that the plane on route i ends its daily trip j into the hub, and

x_{ipj} = 1 if daily trip j into the hub provides a more desirable connect time for CAB certificated carrier flight p on route i than any other daily trip, 0 otherwise.

Then the objective function is:

$$\min_{x, t} q = \sum_i \sum_p \sum_j x_{ipj} w_{ip} (|d_{ip} - t_{ij}|^+ + c|d_{ip} - t_{ij}|^-).$$

The Penalty Constraints

The first set of constraints forces all routes to be penalized at most once for each departure from the hub by a CAB certificated carrier. A route is penalized once

for a CAB flight if there is a demand for that CAB flight from the community preassigned to that route and not penalized otherwise. For example, suppose city A is on route one and there are three daily trips into the hub on route one arriving at 8:00 a.m., 10:00 and 12:00 (see Figure 5.1).

If a passenger wishes to connect with CAB flight one his best choice is the first commuter flight. Thus the penalty should be applied against the first commuter flight and not against the second and third commuter flights.

Now, suppose a passenger wishes to connect with CAB flight two. If he uses the first commuter flight he is two hours and forty-five minutes early. If he uses the second commuter flight he is forty-five minutes early, and if he uses the third commuter flight he is one hour and fifteen minutes late. Thus, his best choice is the second commuter flight and the penalty for the second CAB flight is measured against the second commuter flight only.

Route Feasibility Constraints

The second set of constraints forces each route to have one and only one starting point. The third constraint set requires that all routes be feasible. In other words, a plane must start its route at city m or fly into city m before it may fly out of city m to some other city. The fourth set of constraints requires all routes to have a flight leg into the hub airport.

First Commuter Arrival	- - -	8:00
	8:45	- - - CAB Departure One
Second Commuter Arrival	- -	10:00
	10:45	- - - CAB Departure Two
Third Commuter Arrival	- - -	12:00

FIGURE 5.1
PASSENGER CHOICE PROBLEM

Service Constraints

The fifth set of constraints forces all cities to be included in at least one route. The sixth set of constraints limits the number of stops in a route. The seventh set of constraints is used to preassign cities to routes.

Timing Constraints

The eighth constraint set restricts the time at which a plane on a given route may arrive at the hub provided the plane is not on its first trip of the day into the hub. Constraint set nine restricts the time at which a plane may arrive at the hub during its first trip of the day. Two sets of constraints are required because of the method of routing. On the first trip into the hub the plane flies from city to city and into the hub. On later trips the plane must retrace its steps to the starting point and then fly back to the hub again. Thus, the time to fly between cities must be multiplied by two for the second, third, etc. trips into the hub. The tenth set of constraints limits the earliest time that a commuter plane may begin its first trip into the hub.

In order to represent the constraints described above consider the following notation.

Let $z_{mi} = 1$ if route i begins at city m , 0 otherwise,

$y_{kmi} = 1$ if the plane on route i flies non-stop between cities k and m , 0 otherwise,

t_{ij} = the time that the plane on route i finishes daily trip j into the hub,

s_{ij} = the slack time for the plane on route i between trips j and $j-1$,

L = the limit or the number of stops in any route,

a_{km} = the flight time between cities k and m ,

and

b = the earliest time that a commuter plane may begin its first daily trip into the hub.

Mathematical Form of the Constraints

Using this notation the constraints may be represented as follows:

- (1) $\sum_j x_{ipj} = 1$ for all i, p
- (2) $\sum_m z_{mi} = 1$ for all i
- (3) $z_{mi} + \sum_{k>m} y_{kmi} - \sum_{m>n} y_{mni} \geq 0$ for all m, i
- (4) $\sum_k y_{khi} = 1$ for all i where h represents the hub airport
- (5) $\sum_i \sum_{k>m} y_{kmi} + \sum_i z_{mi} \geq 1$ for all m
- (6) $\sum_k \sum_{k>m} y_{kmi} \leq L$ for all i
- (7) $z_{mi} + \sum_{k>m} y_{kmi} = 1$ if city m is preassigned to rout i
- (8) $t_{ij} - s_{ij} - t_{i,j-1} + 2 \sum_k \sum_{k>m} y_{kmi} a_{km} = 0$
- (9) $t_{i1} - s_{i1} - \sum_k \sum_{k>m} y_{kmi} a_{km} = 0$ for all $i, j = 1$
- (10) $s_{i1} \geq b$ for all i .

MODEL TRANSFORMATIONS

By making the necessary variable transformations and noting certain characteristics of the problem the model may be transformed into a mixed integer quadratic programming problem. The variable transformation is similar to one found in Francis and White [20].

Let earliness = $e_{ipj} = d_{pi} - t_{ij}$ if $d_p \geq t_{ij}$ and 0 otherwise,

let lateness = $l_{ipj} = t_{ij} - d_{pi}$ if $t_{ij} > d_p$ and 0 otherwise.

The objective function may now be represented by a new objective function and two sets of constraints as follows:

$$\min_{\bar{x}, \bar{e}, \bar{l}} q = \sum_i \sum_p \sum_j x_{ipj} (e_{ipj} + c l_{ipj}) w_{pi}$$

Subject to

$$d_p - e_{ipj} + l_{ipj} - t_{ij} = 0 \text{ for all } ijp$$

$$e_{ipj} \times l_{ipj} = 0 \text{ for all } ijp$$

$$e_{ipj} \geq 0, l_{ipj} \geq 0 \text{ for all } ijp$$

All other constraints and variable definitions are unchanged.

In its present form the model does not represent a simple mixed integer quadratic programming problem. The constraints of the form $e_{ipj} \times l_{ipj} = 0$ are not linear. This would normally prevent the use of standard quadratic programming algorithms.

Fortunately, due to the nature of this problem, the constraints $e_{ipj} \times l_{ipj} = 0$ are redundant and may be ignored. This is due to the relationship between the objective function, the constraints of the form $d_{pi} - e_{ipj} + l_{ipj} - t_{ij} = 0$ and the requirement that e_{ipj} and l_{ipj} are both greater than or equal to zero. For the purpose of an illustration, suppose a CAB flight leaves at 9:00 a.m. ($d_{pi} = 9$) and a commuter arrives at the hub at 8:00 a.m. ($t_{ij} = 8$). One wants to minimize the quantity $e_{ipj} + c l_{ipj}$. This quantity satisfies the constraints $9 - e_{ipj} + l_{ipj} - 8 = 0$, $e_{ipj} \geq 0$, $l_{ipj} \geq 0$ and attains its minimum if $e_{ipj} = 1$ and $l_{ipj} = 0$. Thus, $e_{ipj} \times l_{ipj} = 0$. Suppose $l_{ipj} > 0$, say some positive quantity α . Now, $e_{ipj} = 1 + \alpha$. Previously, the objective function ($e_{ipj} + c l_{ipj}$) was $1 + c(0)$. If $l_{ipj} = \alpha$, the objective function becomes $1 + \alpha + c(\alpha)$ which is greater than one.

Now consider a case where the commuter flight is late for the CAB flight. Suppose the CAB flight leaves at 8:00 a.m. while the commuter flight arrives at 9:00 a.m. ($d_{pi} = 8$ and $t_{ij} = 9$). In this case $e_{ipj} + c l_{ipj}$ must be minimized subject to $8 - e_{ipj} + l_{ipj} - 9 = 0$, $e_{ipj} \geq 0$ and $l_{ipj} \geq 0$. The objective is minimized if $e_{ipj} = 0$ and $l_{ipj} = 1$. Suppose $e_{ipj} = \alpha > 0$. In this case $l_{ipj} = 1 + \alpha$. The objective function becomes $\alpha + c(1 + \alpha)$ compared to c if $e_{ipj} = 0$. Thus $e_{ipj} \times l_{ipj} = 0$ if the affected portion of the objective function is to be minimized. Thus, independently of whether a commuter flight is early

or late for a CAB flight the constraints $e_{ipj} \times l_{ipj} = 0$ hold automatically.

Implicit in these illustrations is the assumption that $x_{ipj} = 1$. If $x_{ipj} = 0$ there is no penalty for earliness or lateness being applied. Thus e_{ipj} and l_{ipj} may assume any value without affecting the objective function and one need not worry about the constraints of the form $d_{pi} - e_{ipj} + l_{ipj} - t_{ij} = 0$ or $e_{ipj} \times l_{ipj} = 0$. For simplicity it was assumed in this illustration that w_{ip} is one.

At this point it may be shown that the problem is a mixed integer quadratic programming problem. The objective function needs to be represented in the form $\bar{y}^T B + \frac{1}{2} \bar{y}^T A \bar{y}$, where A is a symmetric $m \times m$ positive semidefinite matrix and B is a $m \times 1$ dimensioned matrix. Let B be defined as 0 (a $m \times 1$ matrix of zeros) and $\bar{y}^T$ be defined as:

$$[x_{111} \ x_{112} \ \dots \ x_{ipj} \ \dots (e_{111} + c \ l_{111}) \ \dots (e_{ipj} + c \ l_{ipj}) \ \dots]$$

Let A be a matrix of the form:

$$\begin{bmatrix} 0 & \dots & 0 & w_{11} & 0 & \dots & 0 \\ & & & 0 & w_{11} & \dots & \\ \vdots & \vdots & \vdots & \vdots & & & \vdots \\ \vdots & \vdots & \vdots & \vdots & w_{12} & \dots & \vdots \\ 0 & \dots & 0 & 0 & & \dots & w_{ip} \\ w_{11} & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & w_{11} & \dots & & & & \\ \vdots & & & \vdots & \vdots & & \vdots \\ \vdots & w_{12} & \dots & \vdots & \vdots & & \vdots \\ \vdots & & & \vdots & \vdots & & \vdots \\ 0 & \dots & w_{ip} & 0 & & \dots & 0 \end{bmatrix}$$

By inspection it may be determined that A is symmetric and that

$$\frac{1}{2} \bar{y}^T A \bar{y} + \bar{y}^T B = \sum_i \sum_p \sum_j x_{ipj} (e_{ipj} + c l_{ipj}).$$

Due to the definition (0,1) of each variable x_{ipj} , these variables will be greater than or equal to zero. Also, from nonnegativity constraints in the model, e_{ipj} and l_{ipj} are greater than or equal to zero. Therefore the product $x_{ipj} (e_{ipj} + c l_{ipj})$ is greater than or equal to zero and furthermore the sum of these products is greater than or equal to zero. Thus, the matrix A is positive semidefinite over the domain of the variables even though the matrix A is not, in general, positive semidefinite.

MATHEMATICAL MODEL FOR THE AFTERNOON SYSTEM

The afternoon model may be formulated in the same manner as the morning model except that the commuters attempt to meet CAB arrivals into the hub as well as CAB departures from the hub. Define d_{pi} to be the time CAB carrier p arrives at the hub if the flight has passengers connecting to the commuter or the time CAB flight p departs from the hub if passengers are connecting from the commuter. It should be noted that the values of d_{pi} will differ from the actual CAB flight times by the required spack time between connections at the airport. For example, suppose

the first CAB flight to arrive at the hub with commuter passengers arrives at 3:00 p.m. (actual $d_{pi} = 15$) and the required slack is 45 minutes. Then the value of d_{pi} used in the model will be 15.75.

To formulate the objective function, consider the following definitions.

Let earliness = $e_{ipj} = d_p - t_{ij}$ if $d_{pi} \geq t_{ij}$
and 0 otherwise for p contained in P_1

lateness = $l_{ipj} = t_{ij} - d_{pi}$ if $t_{ij} > d_{pi}$ and 0 otherwise
for p contained in P_1

earliness = $e_{ipj} = t_{ij} + s_{i,j+1} - d_{pi}$ if $t_{ij} + s_{i,j+1} \geq d_p$
0 otherwise for p contained in P_2

lateness = $l_{ipj} = d_{pi} - t_{ij} - s_{i,j+1}$ if $d_{pi} > t_{ij} + s_{i,j+1}$
0 otherwise for p contained in P_2

where P_1 is the set of departure times from the hub by CAB carriers and P_2 is the set of arrivals at the hub by CAB carriers with passengers connecting to/from the commuters for route i .

The objective function becomes:

$$(P^1) \min_{\bar{x}, \bar{e}, \bar{l}} z = \sum_i \sum_p \sum_j x_{ipj} (e_{ipj} + c l_{ipj}) w_{pi}$$

All original constraints remain unchanged.

A principal assumption for the afternoon model is that a commuter will not begin its return trip to the

outlying communities as soon as possible after landing at the hub if it has some slack time. If a commuter aircraft has some free time or slack time between its trips into the hub it is assumed that the plane will fly its route with no waiting anywhere except at the hub. This forces the commuter to remain at the hub longer and allows more passengers the opportunity to connect with the commuter. In the morning model one need assume only that the trip into the hub is made without holding at any airport on trips into the hub. If the plane has some slack time before its next trip it may wait at the hub, any of the cities along the return route to the starting point or the starting point itself.

SUMMARY

This chapter presents the mathematical models (P) and (P¹) for the morning and afternoon systems. Industry provided information is used to model the commuter airline routing and scheduling problem. This model is then transformed into a mixed integer quadratic programming problem. Next, the model is modified for an afternoon system. Chapter 6 presents the solution procedure for solving the models developed in this chapter.

CHAPTER 6

SOLUTION PROCEDURE

Due to the fact that the model (P) developed in Chapter 5 involves $(0, 1)$ integer variables, a quadratic objective function and linear constraints, a branch and bound procedure may be utilized as a solution approach. The procedure uses continuous variable quadratic programming to set the bounds.

This chapter presents the theoretical background to the solution procedure and the specific manner in which it is applied to the commuter airline routing and scheduling problem. Topics include a brief introduction to branch and bound methods, the method for applying a branch and bound procedure to model (P), the theoretical development of Beale's algorithm and a method of reducing the number of variables and constraints in model (P).

BRANCH AND BOUND METHODS

General Description

Branch and bound methods are useful in solving problems which have certain variables which may assume only certain discrete values. In order to develop a branch and bound procedure for solving a particular problem a technique for branching and a method of setting bounds on these

branches are needed. By branching, one means fixing the values of some of the variables. The bound is then determined by optimizing the problem with the rest of the variables unrestricted. (Unrestricted in the sense that the remaining variables are not restricted to certain discrete values.) A solution has been obtained when all of the restricted variables have been assigned specific values. For minimization problems the bound at any stage will be less than or equal to the solution obtained if the restricted variables currently fixed keep their present values while the remainder of the restricted variables are fixed. Additional restrictions to a problem cannot improve the optimal solution.

Once a solution has been reached, it may be used to "prune" branches. Any time the bound on a given branch is greater than this solution there is no need to explore that branch further because it cannot lead to a better solution. Only those branches indicating a more promising solution need be explored. Each time a better solution is found, that solution is used to evaluate branches until all branches have been explored to a final solution or have been "pruned." For a more detailed discussion of branch and bound procedures the reader is referred to Cooper and Steinberg [5].

Application to Models (P) and (P^1)

To solve the commuter airline routing and scheduling problem, the branching is performed by setting one of the

integer variables to zero. The problem is updated and solved as a continuous variable quadratic programming problem. To update the problem, the variable is simply dropped in this case. Then, the same variable is set to one and the problem updated and solved again. To update the problem in this case, the variable is dropped and the technological coefficients of these variables are subtracted from the resources corresponding to the same constraints.

If the "one" problem yields a solution less than the "zero" problem the problem is left as it is. If not, the problem is returned to its previous (zero) form. The next integer variable is then set first to zero and then to one with the problem being updated and then solved as a continuous variable quadratic programming problem in each case. Again, if the "one" problem has a lower solution (bound), that branch is pursued. If not the "zero" branch is pursued. This continues until all branches have been explored or "pruned."

THE QUADRATIC PROGRAMMING ALGORITHM

The Kuhn Tucker Conditions

Beale's algorithm, as discussed in the following section, is used to solve the continuous variable quadratic programming problems. Before discussing Beale's algorithm it will be helpful to review the Kuhn Tucker necessary conditions [22].

Optimality Conditions. Consider the problem

$$\min_{\bar{x}} z = f(\bar{x})$$

$$\text{subject to } g_i(\bar{x}) = 0 \quad i = 1 \dots m^1$$

$$g_i(\bar{x}) \geq 0 \quad i = m^1 + 1 \dots m^2$$

where all functions are differentiable. In addition, all constraints must satisfy a certain qualification. (Linear constraints satisfy this qualification.) The reader is referred to Zangwell [22] for a further discussion.

If x^* is an optimal solution to the problem three conditions hold:

(1) $\bar{x}^*$ is feasible

(2) there exist multipliers $\lambda_i \geq 0$, $i = 1, \dots, m^1$

and λ_i unrestricted, $i = m^1 + 1, \dots, m^2$ such that $\lambda_i g_i(\bar{x}^*) = 0$
 $i = 1, \dots, m^2$

$$(3) -\nabla f(\bar{x}^*) + \sum_{i=1}^{m^2} \lambda_i g_i(x^*) = 0.$$

Beale's Algorithm

The quadratic programming problem with slack and surplus variables added may be stated as:

$$\min_{\bar{x}} z = \bar{p}^T \bar{x} + 1/2 \bar{x}^T D \bar{x}$$

$$\text{subject to } A \bar{x} - \bar{b} = \bar{0}$$

$$\bar{x} \geq 0$$

where p , x are $n \times 1$, b is $m \times 1$, A is $m \times n$ and D is symmetric positive semidefinite $n \times n$.

If $\bar{x}^*$ is a local minimum then from (3) $-\bar{p} - D\bar{x} + A^T \bar{y} + \bar{\lambda}_2 = \bar{0}$

where, $\bar{y} = [\lambda_1 \dots \lambda_m]$ and $\bar{\lambda}_2 = [\lambda_{m+1} \dots \lambda_{m+n}]$.

Therefore, $\bar{\lambda}_2 = \bar{p} + D\bar{x} - A^T\bar{y}$.

Hence, (4) $\bar{x}^{*T} (\bar{p} + D\bar{x}^* - A^T\bar{y}) = \bar{0}$ from (2)

$$(5) \bar{y}^T (A\bar{x}^* - \bar{b}) = 0 \text{ from (1)}$$

and since $\bar{\lambda}_2 \geq \bar{0}$, then

$$(6) \bar{p} + D\bar{x}^* \geq A^T\bar{y}.$$

These are similar to the conditions stated in [12].

Now define $\bar{c}$ as $\nabla f(\bar{x}^*) = \bar{p}^T\bar{x}^* + D\bar{x}^*$.

(That is the gradient evaluated at $\bar{x}^*$).

Then $\bar{x}^*$ minimizes the linear function

$$\begin{aligned} \min_{\bar{x}} z &= \bar{c}^T \bar{x} \\ \text{subject to } A\bar{x} - \bar{b} &= \bar{0} \\ \bar{x} &\geq 0. \end{aligned}$$

Define B^* to be the $m \times m$ basis matrix at $\bar{x}^*$, $\bar{x}_B$ to be the $m \times 1$ basis vector and $\bar{c}_B$ to be those elements of $\bar{c}$ corresponding to $\bar{x}_B$. By linear programming optimality conditions [5], $x_j - c_j \leq 0$ for any nonbasic j . Since $z_j = \bar{c}_B^T B^{-1}a_j$, $\bar{c}_B^T B^{-1}a_j - c_j \leq 0$. If we define $\bar{y} = \bar{c}_B^T B^{-1}$; then $A^T\bar{y} - \bar{c} \leq 0$ are to optimality conditions. This is (6). Thus the linear programming problem is optimized if the quadratic programming problem is optimized.

Now it will be shown that if $\bar{x}^*$ minimizes the linear programming problem then conditions (4), (5) and (6) hold, thus the quadratic programming problem is optimized. From the LP optimality conditions [5]:

$\bar{c}_B^T B^{-1} a_j - c_j = 0$ if j is basic and $\bar{c}_B^T B^{-1} a_j - c_j \leq 0$ if j is non basic. Therefore, $A^T \bar{y} - \bar{c} \leq \bar{0}$ (6). If x_i is basic then $\bar{c}_B^T B^{-1} a_j - c_j = 0$. If x_i is nonbasic then, $x_i = 0$. Thus $\bar{x}^{*T} (\bar{c} - A^T \bar{y}) = \bar{0}$ which is (4). Finally the constraints force $A \bar{x}^* - \bar{b} = \bar{0}$, so that condition (5) also holds.

Convexity

As shown in Chapter 5, the objective function for the commuter airline routing and scheduling problem is a positive semidefinite matrix. Thus, the objective function is convex. Therefore, a local minimum is also a global minimum [22].

Artificial Constraints

The algorithm moves from basis to basis using the usual simplex procedure. At each basic solution the function $\bar{c} = \bar{p} + D\bar{x}$ is computed. It then checks the optimality of $\bar{c}^T \bar{x}$. If this test fails, the usual simplex rules are used to determine the direction of movement. However, in some cases the QP objective function may first decrease and then increase before the next basic solution is reached. To prevent this, artificial constraints are introduced to stop the movement before the objective function begins to increase. The method of determining the artificial constraints is from Land and Morton [11]. Figure 6.1 illustrates an artificial constraint.

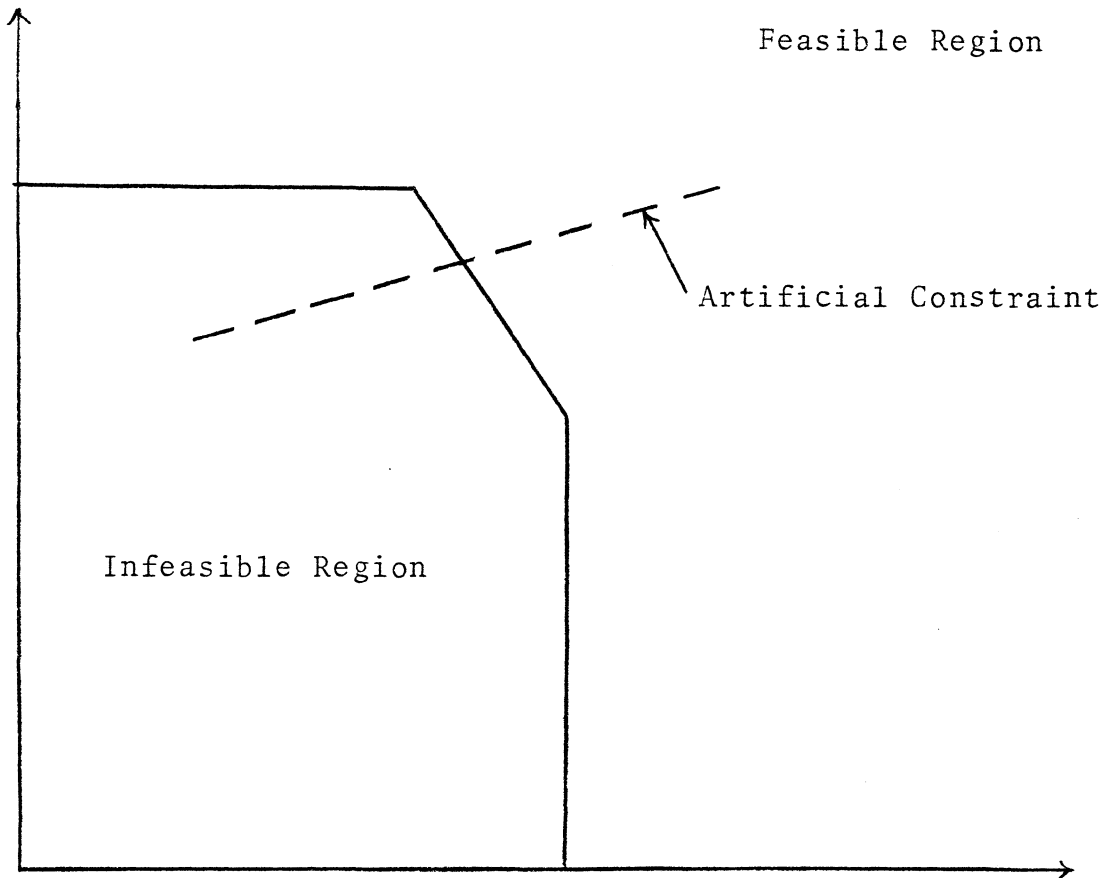


FIGURE 6.1
EXAMPLE OF AN ARTIFICIAL CONSTRAINT

Corresponding to a basic solution define B as the basis matrix, Q the remainder of A , $\bar{x}_B$ the basis vector and $\bar{x}_Q$ the nonbasic component of $\bar{x}$. Then

$$(7) \quad B\bar{x}_B + Q\bar{x}_Q = \bar{b}, \quad \bar{x}_B \geq \bar{0}, \quad \bar{x}_Q = \bar{0}$$

$$(8) \quad \bar{x}_R = B^{-1}\bar{b} - B^{-1}Q\bar{x}_Q.$$

Partition Q and $\bar{x}_Q$ into $(a_k \ R)$ and $(x_k \ \bar{x}_R)$ where a_k is any column of Q and x_k is the single element of $\bar{x}_Q$ corresponding to a_k .

Then from (8): (9) $\bar{x}_B = B^{-1}a_k x_k - B^{-1}R\bar{x}_R$. Now,

$$\frac{df(x)}{dx_k} = \sum_{x_i \in x_B} \frac{\partial f}{\partial x_i} \cdot \frac{dx_i}{dx_k} + \frac{\partial f}{\partial x_k},$$

$$(10) \quad \frac{df(x)}{dx_k} = \sum_{x_i \in x_B} c_i \frac{dx_i}{dx_k} + c_k \quad \text{and} \quad (11) \quad \frac{d\bar{x}_B}{dx_k} = B^{-1}a_k \quad \text{from (9)}.$$

Therefore (10) may be written as $-\bar{c}_B^T B^{-1}a_k + c_k$.

For a basic solution to be optimal $\bar{c}_B^T B^{-1}a_k - c_k \leq 0$. Thus, to prevent an increase in $f(x)$; constraints of the form $\bar{c}_B^T B^{-1}a_k - c_k \geq 0$ are used. These constraints are referred to as artificial constraints. The reader should remember that artificial constraints are not true constraints limiting the feasible solution space but rather stopping points when moving in a given direction. If at any iteration the slack variable associated with an artificial constraint becomes positive the artificial constraint and the associated slack may be eliminated from the problem. If the associated slack becomes positive the constraint is not effective and since it does not really limit the feasible region it serves no purpose.

PROBLEM SIMPLIFICATIONS

For a moderate size commuter system there may easily be 50 binary (0,1) variables. This leads to $2^{50} = 1.126 \times 10^{15}$ possible problems that may need to be solved. To solve this many problems at one second per problem would take 35,700,000 years. Although feasibility criteria in this problem will automatically prevent exploration of many branches, this may not be a sufficient reduction. Fortunately, due to the nature of models (P) and (P¹) it is not necessary to treat all of the binary variables explicitly as integer variables. In addition, some of the integer variables and two sets of constraints may be eliminated from the problem. The following sections explain the rationale of these changes.

The Passenger Choice Variables

The integer variables in the commuter airline routing and scheduling problem are the x, y and z type variables. Fortunately, due to the nature of the objective function and the first set of constraints, the passenger choice (x) variables may be treated as continuous. The portion of the objective function $\sum_j x_{ipj} w_{ip} (e_{ipj} + c_{1ipj})$ and the constraint $\sum_i x_{ipj} = 1$, force one x_{ipj} to be one and the rest zero. The x_{ipj} to be one will correspond to the minimum $(e_{ipj} + c_{1ipj})$ for each combination of i and p. An exception will occur if two trips into the hub on the

same route connect equally well with a particular CAB flight. This presents no problem because the value of the objective function is unchanged for any convex combination of the two x variables. In other words, since the two earliness, lateness penalties are identical, any feasible method of weighting the penalties will yield the same result in the objective function as giving all the weight to one or the other penalty.

The Starting Point Variables

It is possible to reduce the number of integer variables by reducing the number of starting point (z) variables through the second set of constraints. The constraints state that each route will start at one and only one community. Hence, it is known that if a route starts at community A it does not start at any other city. Also, if the route does not start at any other community it will start at community A. Therefore one may eliminate z_{1i} for each route from the problem and determine the values of these variables from the other variables. The need for the second set of constraints is also eliminated. For each route the number of z variables is reduced by one.

For instance, consider the constraint $z_{11} + z_{12} + z_{13} = 1$. Obviously, $z_{11} = 1 - z_{12} - z_{13}$. The variable z_{11} may be eliminated from the problem by substituting

$1 - z_{12} - z_{13}$ wherever z_{11} occurs. It is easily determined that substituting $1 - z_{12} - z_{13}$ for z_{11} in the constraint yields $0 = 0$. Therefore, the need for the constraint is eliminated. The value of z_{11} may be determined from the values of z_{21} and z_{31} after the problem has been solved.

The Community-Connecting Variables

The fifth set of constraints may be used to reduce the number of community-connecting (y) variables. If it is known that a given route connects directly with the hub from community A then it does not directly connect with the hub from any other community. Also, if the route connects directly with the hub from some other community it does not connect directly with the hub from community A. If k is the closest city to the hub ($k=1$) and m is the hub ($m=0$), then $y_{kmi} = y_{10i}$ may be eliminated.

For example, consider the constraint $y_{401} + y_{301} + y_{201} + y_{101} = 1$. Here, $y_{101} = 1 - y_{401} - y_{301} - y_{201}$. The variable y_{101} may be eliminated from the problem by substituting $1 - y_{401} - y_{301} - y_{201}$ everywhere y_{101} appears. Note that substituting $1 - y_{401} - y_{301} - y_{201}$ into the constraint for y_{101} yields $0 = 0$. Therefore, the constraint may be eliminated from the problem provided the substitution for y_{101} is made.

SUMMARY

The commuter airline routing and scheduling problem model (P), may be solved with a branch and bound procedure with bounds set by continuous variable quadratic programming. Beale's algorithm is used for the quadratic programming problems.

This chapter explains the manner in which the general concept of a branch and bound procedure is applied to model (P). Next the theoretical development of Beale's algorithm is detailed. The gradient vector evaluated at the current basis solution is used to permit the use of LP optimization criteria in determining the direction of movement. Artificial constraints are introduced to prevent "overshooting."

Finally, a number of particular characteristics of model (P) are used to reduce the number of computations involved. The following steps summarize the solution procedure:

- A. Perform problem simplifications to reduce the number of starting point and community-connecting variables.
- B. Initialize the current optimal to 10^{13} and stage to the first integer variable.
- C. Set the variable corresponding to stage to zero and solve the remaining problem using continuous variable quadratic programming to obtain s^0 .

- D. Set the variable corresponding to stage to one and solve the remaining problem to obtain s^1 .
- E. If $\min(s^0, s^1) > \text{current optimal}$ move back to a stage with an unexplored branch with bound $< \text{current optimal}$, and go to step 2. If none exist stop.
- F. If $\min(s^0, s^1) \leq \text{current optimal}$
 - 1. and stage $< \text{last integer variable}$, pursue branch corresponding to $\min(s^0, s^1)$, set stage up by one, go to step 2.
 - 2. and stage = last integer variable, set current optimal to $\min(s^0, s^1)$. Determine if there is an unexplored branch with bound $\leq \text{current optimal}$. If such a branch exists update stage to correspond to this branch and go to step 2. If not stop.

CHAPTER 7

APPLICATION

In order to test the results of the solution procedure discussed in Chapter 6 a FORTRAN program has been developed. A quadratic programming core using Beale's algorithm is taken from [12]. One routine has been written to generate the objective function and the constraints from input data. Another routine has been written to perform the necessary branch and bound computations and transformations. Test data provided by Pilgrim Airlines is used to test both the model and the solution procedure.

PILGRIM AIRLINES

Pilgrim Airlines is a commuter airline based in New London, Connecticut. Its routes connect Boston, Bridgeport, Fishers Island, Hartford, Montreal, New Haven, New London, New York (Laguardia), and New York (JFK) as shown in Figure 7.1. The problem used to test the commuter routing and scheduling algorithm is a subset of the total Pilgrim routing and scheduling problem. Two planes are selected to serve JFK from Bridgeport, Hartford, New Haven and New London (see Figure 7.2). Hartford is preassigned to one route and New London to another. The time to fly between city pairs is given in table 7.2. When possible the flight times are taken directly from the "Pilgrim Airlines System Timetable." Where there is no nonstop flight between a city pair, the

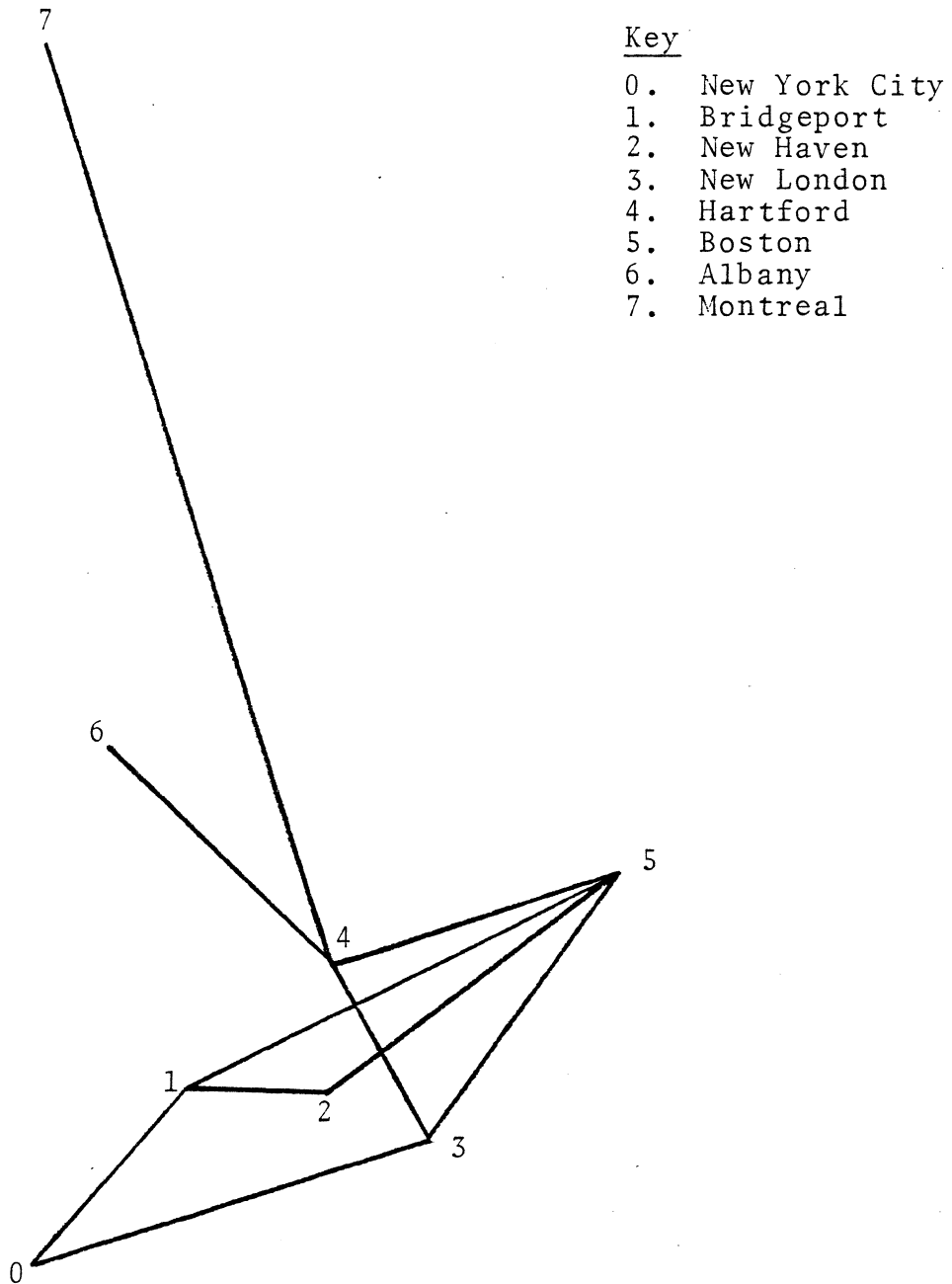


FIGURE 7.1
ROUTE STRUCTURE OF PILGRIM AIRLINES

Key

- 0. New York City
- 1. Bridgeport
- 2. New Haven
- 3. New London
- 4. Hartford

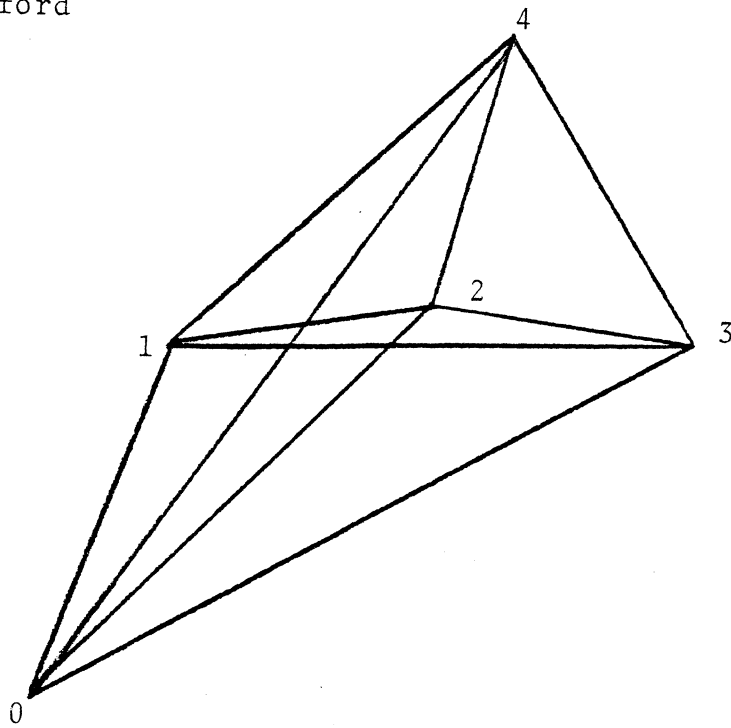


FIGURE 7.2
SUBSET OF PILGRIM AIRLINES
ROUTE STRUCTURE

TABLE 7.1
APPROXIMATE DISTANCE MATRIX (MILES)

From/To	<u>JFK</u>	<u>Bridgeport</u>	<u>New Haven</u>	<u>New London</u>	<u>Hartford</u>
JFK		50	62	96	106
Bridgeport	50		15	50	46
New Haven	62	15		42	33
New London	96	50	42		38
Hartford	106	46	33	38	

TABLE 7.2
FLIGHT TIMES (MIN.)

From/To	<u>JFK</u>	<u>Bridgeport</u>	<u>New Haven</u>	<u>New London</u>	<u>Hartford</u>
JFK		30	35	45	52
Bridgeport	30		21	33	30
New Haven	35	21		30	25
New London	45	33	30		25
Hartford	52	30	25	25	

the flight time is synthesized from the other flight times. Assuming 15 minutes [23] to takeoff, taxi, land, load and unload passengers and using Table 7.1 with the Pilgrim Airlines timetable, the approximate speed of the Pilgrim aircraft is found to be 2.94 miles/minute. Using Table 7.1, the nonstop flight times between city pairs not in the timetable are calculated. Each route will arrive at the hub twice and no flight may begin before 7:00 a.m.

The demand profile for each city preassigned to a route is shown in Table 7.3. The profiles do not reflect actual CAB certificated carrier flights but rather the relative importance of arriving at the hub at each half hour point. This is because it is difficult for Pilgrim Airlines officials to quantify the importance of connecting with individual CAB flights. Since the commuters are attempting to meet certain times rather than actual flights, the penalty for a unit of time late is equivalent to the penalty for a unit of time early.

These demand profiles may be estimated in several ways. The preferred method is to conduct a survey in those communities being served to determine demand as a function of time of day. A second method is to make the weights proportional to the number of CAB flights departing from the hub within a fixed interval (e.g., 15 minutes) of each half hour point. A third method is simply to rely on past experience and knowledge of the system to estimate demand.

TABLE 7.3

RELATIVE DEMAND AS A FUNCTION OF TIME OF DAY

ARIV-DEPT	Time	<u>Hartford</u>	<u>New London</u>
0	8:00 a.m.	6	6
0	8:30	6	6
0	9:00	4	4
0	9:30	3	3
0	10:00	3	3
0	10:30	4	4
0	11:00	6	6

Note: ARIV-DEPT = 0 Passenger connecting to CAB
 certificated carrier from commuter.
 ARIV-DEPT = 1 Passengers connecting to
 commuter from CAB certificated carrier.

RESULTS

Table 7.4 presents the principal output from the Pilgrim Airlines routing and scheduling problem. Hartford and New Haven are assigned to route one with arrival times of 8:30 a.m. and 10:30 a.m. at JFK International. New London and Bridgeport are assigned to route two with arrival times of 8:30 a.m. and 10:36 a.m.

It may be noted that specific times have not been assigned to each flight leg. This is to allow the individual airline more freedom in developing its timetable. As mentioned previously, a fifteen minute allowance for takeoff, landing, loading and unloading passengers has been incorporated into the travel time between city pairs. Thus, if the program schedules a plane to land at 8:30 a.m., as on route one, it actually lands slightly before 8:30 a.m. The Pilgrim timetable might state that the flight arrives at 8:25 a.m. because it is at this point that the passengers may start to debark. The timetable may then state that the aircraft departs from JFK at 8:30 a.m. Table 7.5 presents the approximate timetable that is developed from the program output.

This problem is executed with 650 bytes of storage and requires 52 minutes to reach a final solution on an IBM 370. There are 122 continuous variables, 24 binary variables, and 92 constraints. The objective function (D) matrix is 56x56. Forty-eight continuous variable quadratic programming

TABLE 7.4

SOLUTION TO THE PILGRIM AIRLINES
ROUTING AND SCHEDULING PROBLEM

BEGIN FROM 1 AT HARTFORD

FLY FROM HARTFORD TO NEW HAVN ON ROUTE 1

FLY FROM NEW HAVN TO JFK INTL ON ROUTE 1

BEGIN FIRST TRIP ON ROUTE 1 AT 7,500

ARRIVE AT JFK INTL ON TRIP 1 AT 8.500

LEAVE JFK INTL AT 8.500

ARRIVE AT JFK INTL ON TRIP 2 AT 10.500

LEAVE JFK INTL AT 10.500

BEGIN ROUTE 2 AT NEW LOND

FLY FROM NEW LOND TO BRIDGEPT ON ROUTE 2

FLY FROM BRIDGEPT TO JFK INTL ON ROUTE 2

BEGIN FIRST TRIP ON ROUTE 2 AT 7.450

ARRIVE AT JFK INTL ON TRIP 1 AT 8.500

LEAVE JFK INTL AT 8.500

ARRIVE AT JFK INTL ON TRIP 2 AT 10.600

LEAVE JFK INTL AT 10.600

TABLE 7.5
APPROXIMATE PILGRIM AIRLINES TIMETABLE

ROUTE	DEPART	TIME (a.m.)	ARRIVE	TIME (a.m.)
1	Hartford	7:30	New Havn	7:55
1	New Havn	7:55	JFK Intl	8:30
1	JFK Intl	8:30	New Havn	9:05
1	New Havn	9:05	Hartford	9:30
1	Hartford	9:30	New Havn	9:55
1	New Havn	9:55	JFK Intl	10:30
1	JFK Intl	10:30	New Havn	11:05
1	New Havn	11:05	Hartford	11:30
2	New Lond	7:27	Bridgept	8:00
2	Bridgept	8:00	JFK Intl	8:30
2	JFJ Intl	8:30	Bridgept	9:00
2	Bridgept	9:00	New Lond	9:33
2	New Lond	9:33	Bridgept	10:06
2	Bridgept	10:06	JFK Intl	10:36
2	JFK Intl	10:36	Bridgept	11:06
2	Bridgept	11:06	New Lond	11:39

problems are solved.

With this test application there is no slack time between trips one and two into JFK on route one. This is known because the plane is scheduled to leave the hub as soon as possible after landing. The same is true for route two. For the sake of argument, suppose the plane assigned to route one is to leave JFK at 9:00 a.m. In this case, there is a 30 minute slack period between trips. Since this problem involves a morning system, the aircraft is free to wait at JFK, New Haven or Hartford. Now, suppose an afternoon system is being optimized and there is a 30 minute slack period between scheduled trips to the hub. In this case, the aircraft is required to wait at the hub in order to allow passengers time to connect with the commuter from CAB certificated carriers.

COMPARISON OF SOLUTIONS

A comparison of the objective function values generated from the solution given in Table 7.5 and the solution currently being used by Pilgrim Airlines is difficult for two reasons:

- A. The routes are developed subject to certain ideal standards which, in reality, are sometimes relaxed.
- B. The number of trips per day into the hub on a given route is also an ideal standard which may

be relaxed.

Never the less, a comparison is presented here with the understanding that other factors, aside from the objective function value, also should be considered.

The objective function value from the program is 25.1. The objective function value with the present solution is 47.5. This indicates that the program reduced passenger inconvenience by 47 percent.

SENSITIVITY

This section briefly discusses the effect that changes in the input data will have on the solution. The demand function has a strong impact on the timing of trips into the hub. Small changes in the weights will probably not affect the routes but will have an impact on the schedule. The relative lateness penalty will also tend to affect the schedule before the routes are affected. As the lateness penalty increases the schedule will shift forward and attempt to reach the hub before the first demand point (i.e., 8:00 a.m. in this Pilgrim application).

The solution to this application will not be affected by changes in the limit on intermediate stops. With the solution generated by the program there is one intermediate stop on each route. Therefore, reducing the limit from two to one will not affect the solution. Also, increasing the limit will not affect the solution. Changes in the

communities to be served, the preassignment of cities to routes, the number of aircraft, or the number of daily trips on a route may affect either the routes or the schedules depending on the nature of the changes.

CONCLUSIONS

The computer program is capable of solving real world problems in a reasonable amount of time. The solution procedure yields an exact, globally optimal solution. The routes form connected tours and the aircraft are fully capable of adhering to the schedule. All constraints are satisfied and passenger inconvenience is minimized.

CHAPTER 8

SUMMARY AND EXTENSIONS

The objective of this paper is to present a management tool to be used by commuter airlines in developing routes and schedules. A mathematical model is developed to represent the decision process involved. A branch and bound solution procedure with bounds set by Beale's algorithm is developed and successfully applied to an operating commuter airline. The problem is summarized below:

A. The principal assumptions are:

1. The commuter provides a feeder service to a single hub.
2. During the morning the traffic flow is one way to the hub.
3. During the afternoon the flow is two way.
4. Once a route is determined the plane will not vary from that route.
5. Routes are constructed such that cities are visited in the order that will minimize the distance flown on the route.
6. A plane will retrace its steps from the hub to the starting point of the route after each trip to the hub.

B. The required input data are:

1. The demand as a function of time of day at

discrete points for each route

2. The number of daily trips into the hub for each route
 3. The maximum allowable number of intermediate stops
 4. The number of available aircraft
 5. The cities to be served
 6. The flight time between each city pair
 7. The penalty for a unit of time late relative to a unit of time early
 8. The earliest time that a route may begin
 9. The city preassigned to each route.
- C. The decision variables are:
1. The cities included on each route
 2. The time(s) that each city is served.
- D. The objective function minimizes passenger wait time at the hub.
- E. The constraints insure that:
1. The routes are constructed such that a plane must either start at city m or fly into city m before flying out of city m .
 2. All routes terminate at the hub.
 3. The number of intermediate stops on any route does not exceed the limit.
 4. All cities are served.
 5. All trips begin after a specified time.

6. A plane has sufficient time to complete its route between scheduled stops at the hub.

The solution procedure involves the following steps:

- A. Perform problem simplifications to reduce the number of starting point and community-connecting variables.
- B. Initialize the current optimal to 10^{13} and stage to the first integer variable.
- C. Set the variable corresponding to stage to zero and solve the remaining problem using continuous variable quadratic programming to obtain s^0 .
- D. Set the variable corresponding to stage to one and solve the remaining problem to obtain s^1 .
- E. If $\min(s^0, s^1) > \text{current optimal}$ move back to a stage with an unexplored branch with bound $< \text{current optimal}$, and go to step 2. If none exist stop.
- F. If $\min(s^0, s^1) \leq \text{current optimal}$
 1. and $\text{stage} < \text{last integer variable}$, pursue branch corresponding to $\min(s^0, s^1)$, set stage up by one, go to step 2.
 2. and $\text{stage} = \text{last integer variable}$, set current optimal to $\min(s^0, s^1)$. Determine if there is an unexplored branch with bound $\leq \text{current optimal}$. If such a branch exists update stage to correspond to this branch and go to step 2. If not stop.

EXTENSIONS

There are two ways in which the model developed in this paper may be expanded to provide a more general representation of the commuter system. The first is to model a system with multiple hubs. This will permit global optimization of some systems which must, with the present model, be separated into subsystems and suboptimized. The second method of providing a more general representation of a commuter system involves modifying the objective function. Currently, w_{ip} is a constant. However, if w_{mp} is defined as the weight corresponding to CAB flight p from city m then:

$$w_{ip} = \sum_m w_{mp} z_{mi} + \sum_k \sum_m w_{mp} y_{kmi}$$

and the objective function becomes:

$$\min q = \sum_i \sum_p \sum_j (\sum_m w_{mp} z_{mi} + \sum_k \sum_m w_{mp} y_{kmi}) x_{ipj} (e_{ipj} + c_{ipj} - 1).$$

Developing a solution procedure with this objective function will provide a challenging problem.

CITED REFERENCES AND BIBLIOGRAPHY

1. Barcum, S. D., et. al., "Airline Seat Share: A Study in False Optimization," Management Science, (October 1974) 148-156.
2. Barnett, A., and Kleitman, D. J., "Optimal Scheduling Policies for Some Simple Transportation Systems," Transportation Science, (February 1973) 85-99.
3. Braitsch, R. J., Jr., "A Computer Comparison of Four Quadratic Programming Algorithms," Management Science, (July 1972) 632-643.
4. Commuter Airlines, (Report Number 3), Commuter Airline Association, 1975.
5. Cooper, L., and Steinberg, D., Methods and Applications of Linear Programming, W. B. Saunders Company, 1974.
6. Ferguson, A. R., and Dantzig, G. B., "The Problem of Routing Aircraft," Management Science, (April 1955) 51-55.
7. Ferguson, A. R., and Dantzig, G. B., "The Allocation of Aircraft to Routes-An Example of Linear Programming Under Uncertain Demand," Management Science, (October 1956)..
8. Gagnon, Gilles, "A Model for Flowing Passengers Over Airline Networks," working paper, Air Canada.
9. Hadley, G., Nonlinear and Dynamic Programming, Addison-Wesley Publishing Company Inc. 1964.
10. Hyman, G. and Gordon, L., "Commercial Airline Scheduling Technique," Transportation Research (March 1968) 23-29.
11. Land, A. H., and Morton G., "An Inverse-Basis Method for Beale's Quadratic Programming Algorithm," Management Science, (January 1973) 510-516.
12. Land, A. H., and Powell, S., FORTTRAN Codes for Mathematical Programming, John Wiley and Sons Ltd., 1973.
13. Miller, J. C., "A Time-of-Day Model for Aircraft Scheduling," Transportation Science, (August 1972) 221-242.
14. Nemhauser, G. L., "Scheduling Local and Express Service," Transportation Science, (May 1969) 164-175.

15. Richardson R., "An Optimization Approach to Routing Aircraft," Transportation Science, (March 1976) 52-71.
16. Salzbom, F. J., "Timetables for a Suburban Rail Transit System," Transportation Science, (November 1969) 297-316.
17. Salzbom, F. J., "Optimum Bus Scheduling," Transportation Science, (May 1972) 137-148.
18. Scheiner, J. I., "The Effect of Interstate Systems on Short Haul Air Passenger Demand," Transportation Science, (November 1967) 286-294.
19. Shaw, G. C., "The Concept of the Best Schedule," working paper, Faculty of Administrative Studies, York University.
20. Francis, D. R., and White, J. A., Facility Layout and Location, Prentice-Hall, Inc., 1974.
21. Young, D. R., "Scheduling a Fixed-Schedule Common Carrier Passenger Transportation System," Transportation Science, (August 1970) 243-269.
22. Zangwell, W. I., Non-Linear Programming A Unified Approach, Prentice-Hall, Inc., 1964.
23. Information provided by John W. Presburg, Director of Passenger Services, Pilgrim Airlines, 1976.
24. Results of a survey of commuter airlines, 1976 (see Appendix A).
25. Information provided by Janet St. Mark, V.P. Administration, National Air Transportation Associations, 1975.

APPENDIX A
SAMPLE QUESTIONNAIRE
AND
SUMMARY OF RESPONSES

QUESTIONNAIRE

1. Approximately what percentage of your passengers connect with CAB certificated carriers?

60 percent

2. Is slotting (i.e., airport officials allowing commuters to land and takeoff only during certain time slots) a problem?

No

3. Approximately what percentage of the day's traffic is carried during the following periods?

	Monday-Friday	Saturday	Sunday
4 a.m.-6 a.m.	0	0	0
6 a.m.-8 a.m.	3	3	3
8 a.m.-10 a.m.	20	0	0
10 a.m.-12 noon	15	15	10
12 noon-2 p.m.	20	25	25
2 p.m.-4 p.m.	9	10	9
4 p.m.-6 p.m.	9	10	11
6 p.m.-8 p.m.	9	20	25
8 p.m.-10 p.m.	<u>15</u>	<u>20</u>	<u>20</u>
Sum	100	100	100

4. What are the planning and operating decisions you have to make in scheduling commuter service (e.g., determination of departure times, determination of routes, assignment of crews and planes to routes, etc.)? Please list in order of decreasing importance if possible.

System Needs
 Competition
 Departure Times
 Block Hours
 Aircraft Flow
 Crew Flow

5. How important is it that a commuter schedule minimize the time that a passenger must wait at a hub airport when connecting between a commuter airline and a CAB certificated carrier?

Except for those cities where there is no competition it is the only thing that commuters have to promote.

6. What do you feel is a reasonable maximum number of allowable intermediate stops during a commuter flight with passengers?

Two stops

7. How important is the timing of a schedule in increasing the market for commuter airlines?

Very important. Commuters, especially in Florida cannot compete with the CAB carriers.

8. Please rate the following measures of effectiveness on a scale of 1 to 10 (10 being very important) as to their importance in evaluating a schedule.

- | | |
|---|----|
| a. low direct operating costs | 8 |
| b. high level of service (i.e., frequent stops) | 8 |
| c. high percentage of non-stop flights | 8 |
| d. small number of planes and crews required | 8 |
| e. timing of the schedule | 10 |

9. Are there any other objectives or resource limitations important in designing a schedule for commuter airlines which have not been discussed? If so, what are they?

Responses are from Florida Air Lines

SUMMARY OF SURVEY RESPONSES

In answer to question five concerning the importance of minimizing connect lag time, the responses are as follows:

1. Unimportant	3
2. Fairly important	2
3. Important	2
4. Very important	<u>5</u>
Total responses	12

In answer to question six concerning the maximum number of intermediate stops the responses are as follows:

1. Zero	1
2. One	7
3. Two	4
4. Three	2
5. Four	<u>1</u>
Total responses	15

In answer to question seven concerning the importance of the timing of a schedule in increasing the market for commuters the responses are as follows:

1. Unimportant	0
2. Fairly important	1
3. Important	2
4. Very important	<u>8</u>
Total responses	11

The results of question eight are presented in Table

A.1.

TABLE A.1
RELATIVE IMPORTANCE OF MEASURES
OF EFFECTIVENESS

Measure of Effectiveness	Score									
	1	2	3	4	5	6	7	8	9	0
a	0	0	1	0	1	0	1	4	0	4
b	1	0	0	0	0	4	0	4	0	2
c	1	0	0	0	1	0	1	2	1	5
d	0	0	0	0	2	0	0	5	2	2
e	0	0	0	0	0	0	1	3	0	7

APPENDIX B

PILGRIM AIRLINES

SYSTEM TIMETABLE

PILGRIM AIRLINES/SYSTEM TIMETABLE

FROM BOSTON

Flt.	Leave	Arrive	Freq.	Stops
To Albany				
464	8:45 PM	10:10 PM	ExSaSu	1

To Bridgeport

155	9:45 AM	11:00 AM	ExSaSu	1
159	2:45 PM	4:00 PM	ExSaSu	1
169	6:10 PM	7:30 PM	ExSaSu	1
167	6:05 PM	7:40 PM	SaSu	2

To Hartford/Springfield

155	9:45 AM	10:25 AM	ExSaSu	0
159	2:45 PM	3:25 PM	ExSaSu	0
164	8:45 PM	9:25 PM	ExSaSu	0

To New Haven

169	6:10 PM	7:40 PM	ExSaSu	0
167	6:05 PM	7:25 PM	SaSu	1

To New London/Groton

167	6:05 PM	6:55 PM	SaSu	0
-----	---------	---------	------	---

To New York J = JFK Int'l L = LaGuardia

169	6:10 PM	8:00 PM L	ExSaSu	2
167	6:05 PM	8:15 PM J	SaSu	3

FROM BRIDGEPORT

Flt.	Leave	Arrive	Freq.	Stops
To Boston				
152	8:25 AM	9:25 AM	ExSaSu	0
156	12:30 PM	2:00 PM	ExSaSu	2
216	3:00 PM	4:30 PM	SaSu	2
162	4:30 PM	5:45 PM	ExSaSu	1

To Hartford/Springfield

156	12:30 PM	1:10 PM	ExSaSu	1
-----	----------	---------	--------	---

To Montreal

575	12:30 PM	2:45 PM	ExSaSu	2
-----	----------	---------	--------	---

To New York J = JFK Int'l L = LaGuardia

169	7:40 PM	8:00 PM L	ExSaSu	0
167	7:45 PM	8:15 PM J	SaSu	0

FROM FISHER ISLAND

Flt.	Leave	Arrive	Freq.	Stops
To New York J = JFK Int'l L = LaGuardia				
607	7:00 AM	7:50 AM L Mon. only	0	
619	7:20 PM	8:10 PM L Sun. only	0	

FROM HARTFORD/SPRINGFIELD

Flt.	Leave	Arrive	Freq.	Stops
To Albany				
464	9:40 PM	10:10 PM	ExSaSu	0

To Boston

156	1:20 PM	2:00 PM	ExSaSu	0
-----	---------	---------	--------	---

To Bridgeport

155	10:10 AM	11:00 AM	ExSaSu	0
159	1:30 PM	4:00 PM	ExSaSu	0

To Burlington

71	8:10 AM	9:25 AM	Sa	0
----	---------	---------	----	---

To Montreal

71	8:10 AM	9:25 AM	ExSaSu	0
71	8:10 AM	10:00 AM	Sa	1

From Hartford/Springfield (Continued)

To Montreal (Cont.)

75	1:30 PM	2:45 PM	ExSaSu	0
77	4:20 PM	6:00 PM	Sa	0
87	4:50 PM	6:05 PM	ExSaSu	0

To New London/Groton

72	11:30 AM	11:55 AM	ExSaSu	0
88	8:10 PM	8:55 PM	ExSa	0

FROM MONTREAL

Flt.	Leave	Arrive	Freq.	Stops
To Hartford/Springfield				
72	10:00 AM	11:15 AM	ExSaSu	0
71	2:00 PM	3:40 PM	Sa	0
76	3:10 PM	4:25 PM	ExSaSu	0
88	6:25 PM	7:40 PM	ExSa	0

To New Haven

72/215	10:00 AM	1:30 PM	ExSaSu	2
--------	----------	---------	--------	---

To New London/Groton

72	10:00 AM	11:55 AM	ExSaSu	1
88	6:25 PM	8:55 PM	ExSa	1

FROM NEW HAVEN

Flt.	Leave	Arrive	Freq.	Stops
To Albany				
464	6:40 PM	10:10 PM	ExSaSu	3

To Boston

152	8:05 AM	9:25 AM	ExSaSu	1
156	12:45 PM	2:00 PM	ExSaSu	1
216	3:15 PM	4:30 PM	SaSu	1
464	6:40 PM	7:55 PM	ExSaSu	1

To Hartford/Springfield

156	12:45 PM	1:10 PM	ExSaSu	0
-----	----------	---------	--------	---

To Montreal

575	12:45 PM	2:45 PM	ExSaSu	1
-----	----------	---------	--------	---

To New York J = JFK Int'l L = LaGuardia

451	7:05 AM	7:15 AM L	ExSaSu	0
211	10:40 AM	11:00 AM L	Daily	0
455	10:45 AM	11:15 AM L	Daily	0
215	1:10 PM	2:00 PM L	Daily	0
459	1:15 PM	4:00 PM L	Daily	0
464	1:45 PM	5:30 PM L	ExSa	0
364	5:20 PM	5:55 PM L	Daily	0
169	7:20 PM	8:00 PM L	ExSaSu	1
167	7:30 PM	8:15 PM J	SaSu	1

FROM NEW LONDON/GROTON

Flt.	Leave	Arrive	Freq.	Stops
To Albany				
464	7:15 PM	10:10 PM	ExSaSu	2

To Boston

152	7:15 AM	9:25 AM	ExSaSu	2
216	3:50 PM	4:30 PM	SaSu	0
162	5:05 PM	5:45 PM	ExSaSu	0
464	7:15 PM	7:55 PM	ExSaSu	0

To Burlington

71	7:10 AM	9:25 AM	Sa	1
----	---------	---------	----	---

From New London/Groton (Continued)

To Hartford/Springfield

71	7:30 AM	7:55 AM	ExSa	0
75	1:00 PM	1:25 PM	ExSaSu	0
464	7:15 PM	9:25 PM	ExSaSu	1

To Montreal

71	7:30 AM	9:25 AM	ExSaSu	1
71	7:30 AM	10:00 AM	Sa	2
75	1:00 PM	2:45 PM	ExSaSu	1

To New York J = JFK Int'l L = LaGuardia

451	6:15 AM	7:45 AM L	ExSaSu	1
201	7:15 AM	8:00 AM L	Daily	0
211	10:00 AM	11:00 AM L	Daily	1
455	10:15 AM	11:15 AM L	Daily	1
215	1:05 PM	2:00 PM L	Daily	1
459	2:40 PM	4:00 PM L	Daily	1
221	4:05 PM	4:50 PM L	Daily	0
464	4:15 PM	5:30 PM L	ExSa	1
227	7:15 PM	8:00 PM J	ExSa	0
167	7:00 PM	8:15 PM J	Sa	2

FROM NEW YORK

Flt.	Leave	Arrive	Freq.	Stops
To Boston				
216	2:30 PM J	4:30 PM	SaSu	3
464	5:55 PM L	7:55 PM	ExSaSu	2

To Bridgeport

216	2:30 PM J	2:55 PM	SaSu	0
-----	-----------	---------	------	---

To Fishers Island

618	5:55 PM L	6:50 PM	Fri. only	0
-----	-----------	---------	-----------	---

To New Haven

452	8:20 AM L	9:00 AM	ExSaSu	0
202	8:30 AM L	9:00 AM	Daily	0
212	11:30 AM J	12:00 N	Daily	0
454	1:10 PM L	1:40 PM	Daily	0
216	2:30 PM J	3:00 PM	ExSaSu	0
216	2:30 PM J	3:10 PM	SaSu	1
462	4:15 PM L	4:50 PM	Daily	0
464	5:55 PM L	6:35 PM	ExSa	0
366	6:05 PM J	6:15 PM	Daily	0
472	8:30 PM L	9:10 PM	ExSa	0
228	8:45 PM J	9:15 PM	Sa	0
192	9:00 PM J	9:35 PM	ExSaSu	0
466	11:00 PM J	11:45 PM	Daily	1
466	11:15 PM L	11:45 PM	Daily	0

To New London/Groton

452	8:20 AM L	9:20 AM	ExSaSu	1
202	8:30 AM J	9:30 AM	Daily	1
212	11:30 AM J	12:30 PM	Daily	1
456	1:10 PM L	2:45 PM	Daily	1
216	2:30 PM J	3:30 PM	ExSaSu	1
216	2:30 PM J	3:40 PM	SaSu	2
464	5:55 PM L	7:05 PM	ExSa	1
221	6:00 PM J	6:45 PM	Daily	0
472	8:40 PM L	9:40 PM	ExSa	1
228	8:45 PM J	9:30 PM	ExSa	0
228	8:45 PM J	9:45 PM	Sa	1
466	11:00 PM J	12:15 AM	Daily	2
466	11:15 PM L	12:15 AM	Daily	1

APPENDIX C

ROUTING AND SCHEDULING
PROGRAM

```

C      MAIN   PROGRAM
C      ROUTINE START
COMMON /TWO/ XNAME(20),A(50),HUB
REAL*8 XNAME,HUB
COMMON /Q/ D(100,100),ISQ(900),HOLDQ(900),KEYTQ(180),MAXQ,NUMQ
COMMON /AREF/ AA(5000),IRQ(275),JCOL(5000),MAXA
DIMENSION NSTAR(10)
REAL INV
COMMON /ONE/ NSTOR3(5000,2),STOR3(5000)
COMMON /THREE/ NSTOR1(1000,2),NSTOR2(5000,2),STOR1(1000),STOR2(500
C0),NS(4),LMAX,LQMAX
COMMON /FOUR/ NLEGS,J(4),P(4),C(7),MM1,I
COMMON /FIVE/ W(500),LSW(500)
COMMON /SEVEN/ REM(5000),ST2(5000),NST2(5000,2),NSET(500),LVAR
COMMON /NINE/ NSTAGE
COMMON /TEN/ RI(500),BND(500)
COMMON /TWELVE/ M1,M2,NFES,SOL,IVAR
DIMENSION WW(4,20),F(4,20),LE(4,20)
INTEGER R
INTEGER P
READ(5,9000) AIR,LINE
9000 FORMAT(2A4)
WRITE(6,9001) AIR,LINE
9001 FORMAT(1H0, 2A4,' AIRLINES')
READ (5,8000) I,R,LIM
8000 FORMAT(3I2)
WRITE(6,8010) I,R,LIM
8010 FORMAT(1H0,'NUMBER OF ROUTES',I2,',', 'NUMBER OF CITIES',I3,',', 'MAX NU
NUMBER OF INTERMEDIATE STOPS',I2)
READ (5,8005) C1,TSTAR
8005 FORMAT(2F8.3)
WRITE(6,8020) TSTAR

```

```

8020 FORMAT(1H0,'EARLIEST STARTING TIME IS',F9.3)
      WRITE(6,8030) C1
8030 FORMAT(1H0,'RELATIVE LATENESS PENALTY IS',F9.3)
      DO 77 K=1,I
        77 READ(5,8006) NSTAR(K)
8006 FORMAT(I2)
      DO 40 K=1,I
        READ (5,8001) J(K)-P(K)
8001 FORMAT(2I2)
        40 WRITE(6,8050) J(K),K,P(K)
8050 FORMAT(1H0,'THERE ARE',I2,' TRIPS ON ROUTE',I2,' WITH',I3,' CONNEC
      UTING POINTS')
      C(1)=0.
      C(2)=0.
      C(3)=0.
      C(4)=0.
      C(5)=0.
      NLEGS=0
      MM1=R-1
      DO 100 K=1,MM1
100  NLEGS=NLEGS+MM1+2-K
      NLAGS=NLEGS+1
      DO 41 K=1,NLAGS
        41 READ(5,8002) A(K)
8002 FORMAT(F6.3)
      DO 500 K=1,I
        NX=P(K)
        DO 500 K1=1,NX
          C(1)=C(1)+J(K)
        500 C(2)=C(2)+J(K)
        DO 501 K=1,I
          C(3)=C(3)+J(K)*2.+1

```

```

      C(4)=C(4)+MM1
501  C(5)=C(5)+NLEGS
      C(6)=C(2)
      C(7)=C(2)
      WRITE(6,8080)
8080 FORMAT(1H0,'ROUTE   CONNECTING TIME   WEIGHT   ARIV-DEPT SWITCH')
      NCOL=C(1)
      DO 70 K=1,I
        NX=P(K)
        DO 70 K1=1,NX
          READ (5,8003) F(K,K1),WW(K,K1),LE(K,K1)
8003  FORMAT(2F10.3,I1)
          WRITE(6,8090) K,F(K,K1),WW(K,K1),LE(K,K1)
8090  FORMAT(' ',I3,9X,F9.3,4X,F10.3,6X,I1)
          NX2=J(K)
          DO 70 K2=1,NX2
            NCOL=NCOL+1
            LSW(NCOL)=LE(K,K1)
            70  W(NCOL)=WW(K,K1)
          READ(5,7004) HUB
7004  FORMAT(A8)
          DO 73 K=1,R
            READ(5,7004) XNAME(K)
            73  CONTINUE
          DO 94 K=1,I
            K1=NSTAR(K)
            WRITE(6,8040) K,XNAME(K1)
8040  FORMAT(1H0,'ROUTE',I2,' INCLUDES ',A8)
          94  CONTINUE
          K4=1
          K1=R
          DO 92 K3=1,R

```

```

      K2=K1-1
      DO 91 K=1,K1
        IF(K2.NE.0) GO TO 90
        WRITE(6,3060) XNAME(K1),HUB,A(K4)
        GO TO 91
      90 WRITE(6,3060) XNAME(K1),XNAME(K2),A(K4)
3060 FORMAT(1H0,'TRAVEL TIME FROM ',A3,' TO ',A8,' IS ',F9.3)
      K2=K2-1
      91 K4=K4+1
      K1=K1-1
      92 CONTINUE
      L=0
C      GENERATE BOUNDS ON VARIABLES
      N1=1
      N2=N1-1+MM1*I
      NVAR=C(1)+C(2)+C(3)+1
      DO 3 K=N1,N2
        L=L+1
        NSTOR2(L,1)=NVAR
        NSTOR2(L,2)=0
        STOR2(L)=1.
        NVAR=NVAR+1
      3 CONTINUE
      N1=N2+1
      N2=N1-1+NLEGS*I
      NVAR=C(1)+C(2)+C(3)+C(4)+1
      DO 4 K=N1,N2
        L=L+1
        NSTOR2(L,1)=NVAR
        NSTOR2(L,2)=0
        STOR2(L)=1.
        NVAR=NVAR+1

```

```

      4 CONTINUE
C      GENERATE RIGHT HAND SIDE
      NO=0
      N1=N2+1
      N2=N1-1+R*I+R+I
      DO 5 K=N1,N2
      NO=NO+1
      N3=N1+R*I-1
      IF(K.GT.N3) GO TO 6
      L=L+1
      NSTOR2(L,1)=NO
      NSTOR2(L,2)=2
      STOR2(L)=0.
      NX=0
      GO TO 5
    5 N3=N3+R
      IF(K.GT.N3) GO TO 7
      L=L+1
      NSTOR2(L,1)=NO
      NSTOR2(L,2)=2
      IF(NX.GT.0) GO TO 200
      STOR2(L)=-1.
      NX=NX+1
      GO TO 5
200 STOR2(L)=1.
      GO TO 5
    7 N3=N3+I
      IF(K.GT.N3) GO TO 5
      L=L+1
      NSTOR2(L,1)=NO
      NSTOR2(L,2)=1
      STOR2(L)=LIM

```

```

5  CONTINUE
   DO 8 K=1,I
     NJ=J(K)
     DO 8 K1=1,NJ
       NO=NO+1
       L=L+1
       N4=2
       IF(K1.EQ.1) N4=1
       NSTOR2(L,1)=NO
       NSTOR2(L,2)=0
8  STOR2(L)=A(NLAGS)*N4
     DO 50 K=1,I
       NX=J(K)
       DO 50 K1=1,NX
         NX1=P(K)
         DO 50 K2=1,NX1
           NO=NO+1
           L=L+1
           NSTOR2(L,1)=NO
           NSTOR2(L,2)=0
50  STOR2(L)=F(K,K2)
     DO 76 K=1,I
       L=L+1
       NO=NO+1
       NSTOR2(L,1)=NO
       NSTOR2(L,2)=0
       STOR2(L)=1.
       IF(NSTAR(K).EQ.1) STOR2(L)=0.
76  CONTINUE
     DO 1 K=1,I
       NP=P(K)
       DO 1 K1=1,NP

```

```

      L=L+1
      NO=NO+1
      NSTOR2(L,1)=NO
      NSTOR2(L,2)=0
      STOR2(L)=1.
1  CONTINUE
      DO 32 K=1,I
      NP=P(I)
      DO 82 K1=1,NP
      NJ=J(I)
      DO 82 K2=1,NJ
      L=L+1
      NO=NO+1
      NSTOR2(L,1)=NO
      NSTOR2(L,2)=0
32  STOR2(L)=0.
      DO 84 K=1,I
      L=L+1
      NO=NO+1
      NSTOR2(L,1)=NO
      NSTOR2(L,2)=2
94  STOR2(L)=TSTAR
C  GENERATE CONSTRAINTS
      DO 9 K=1,I
      NVAR=C(1)+C(2)+C(3)+(K-1)*MM1
      DO 10 K1=1,MM1
      NVAR=NVAR+1
      L=L+1
      NSTOR2(L,1)=K
      NSTOR2(L,2)=NVAR
      STOR2(L)=-1.
10  CONTINUE

```

```

NVAR=C(1)+C(2)+C(3)+C(4)+(K-1)*NLEGS+MM1
DO 11 K1=1,MM1
L=L+1
NSTOR2(L,1)=K
NSTOR2(L,2)=NVAR
STOR2(L)=1.
NVAR=NVAR+R-K1
11 CONTINUE
DO 9 K1=1,MM1
L=L+1
NSTOR2(L,1)=K
NSTOR2(L,2)=NVAR
STOR2(L)=1.
NVAR=NVAR-K1-1
9 CONTINUE
NO=I
DO 12 K=2,R
DO 12 K1=1,I
NO=NO+1
L=L+1
NVAR=C(1)+C(2)+C(3)+1+(K1-1)*MM1+(K-2)
NSTOR2(L,1)=NO
NSTOR2(L,2)=NVAR
STOR2(L)=1.
NVAR=C(1)+C(2)+C(3)+C(4)+R-K+NLEGS*(K1-1)
IF(K.GE.R) GO TO 14
NX=MM1
DO 13 KL=K,MM1
L=L+1
NSTOR2(L,1)=NO
NSTOR2(L,2)=NVAR
STOR2(L)=1.

```

```

      NVAR=NVAR+NX
13  NX=NX-1
14  CONTINUE
      IF(K.LT.R) GO TO 16
      NX=0
      GO TO 17
16  NX=0
      NX1=R
      NX2=R-K
      DO 15 K2=1,NX2
      NX=NX+NX1
15  NX1=NX1-1
17  NX=NX+1
      NVAR=C(1)+C(2)+C(3)+C(4)+NX-1+(K1-1)*NLEGS
      DO 12 K2=1,K
      NVAR=NVAR+1
      L=L+1
      NSTOR2(L,1)=NO
      NSTOR2(L,2)=NVAR
      STOR2(L)=-1.
12  CONTINUE
      NVAR=C(1)+C(2)+C(3)
      NC=NC+1
      N3=C(4)
      DO 18 K=1,N3
      L=L+1
      NVAR=NVAR+1
      NSTOR2(L,1)=NO
      NSTOR2(L,2)=NVAR
      STOR2(L)=-1.
18  CONTINUE
      DO 19 K=1,I

```

```

NVAR=C(1)+C(2)+C(3)+C(4)+(K-1)*NLEGS+R-1
DO 19 K1=1,MM1
L=L+1
NSTOR2(L,1)=NO
NSTOR2(L,2)=NVAR
STOR2(L)=1.
19 NVAR=NVAR+(R-K1)
DO 20 K=2,R
NO=NO+1
NVAR=C(1)+C(2)+C(3)+K-2+1
DO 21 K1=1,I
L=L+1
NSTOR2(L,1)=NO
NSTOR2(L,2)=NVAR
STOR2(L)=1
21 NVAR=NVAR+MM1
IF(K.EQ.R) GO TO 20
DO 30 K1=1,I
NVAR=C(1)+C(2)+C(3)+C(4)+(R-K)+NLEGS*(K1-1)
DO 30 K2=K,MM1
L=L+1
NSTOR2(L,1)=NO
NSTOR2(L,2)=NVAR
STOR2(L)=1
NVAR=NVAR+R-K2+1
30 CONTINUE
20 CONTINUE
DO 22 K=1,I
NO=NO+1
NVAR=C(1)+C(2)+C(3)+C(4)+(K-1)*NLEGS+1
DO 22 K1=2,R
NX=R-K1+1

```

```

DO 23 K2=1,NX
L=L+1
NSTOR2(L,1)=NO
NSTOR2(L,2)=NVAR
STOR2(L)=1.
23 NVAR=NVAR+1
22 NVAR=NVAR+1
NSX=J
DO 24 K=1,I
NX=J(K)
NSX=NSX + NX *2. + 1
DO 24 K1=1,NX
NO=NO+1
NVAR=C(1)+C(2)+NSX-(NX*2.+1)+(K1-1)*2.
DO 25 K2=1,3
IF (K1.EQ.1.AND.K2.EQ.1) GO TO 25
L=L+1
NSTOR2(L,1)=NO
NSTOR2(L,2)=NVAR
IF(K2.EQ.1) STOR2(L)=-1.
IF(K2.EQ.2) STOR2(L)=-1.
IF(K2.EQ.3) STOR2(L)=1.
25 NVAR=NVAR+1
MVAR=0
M1=R
NVAR=C(1)+C(2)+C(3)+C(4)+NLEGS*(K-1)
27 DO 26 K2=1,M1
L=L+1
NSTOR2(L,1)=NO
NVAR=NVAR+1
NSTOR2(L,2)=NVAR
MVAR=MVAR+1

```

```

N4=2
IF(K1.EQ.1) N4=1
STOR2(L)=-N4*A(MVAR)
IF(K2.EQ.N1) STOR2(L)=-N4*(A(MVAR)-A(NLAGS))
26 CONTINUE
M1=M1-1
IF(M1.GE.2) GO TO 27
24 CONTINUE
NS1=J
NJ=C(1)+C(2)
NSX=0
DO 23 K=1,I
  NSX=NSX+2.*J(K)+1
  NX=J(K)
  DO 23 K1=1,NX
    NX1=P(K)
    NVAR1=NSX-(2.*J(K)+1)+2.*K1 + NJ
    IF(K.NE.1) GO TO 29
    NVAR2=C(1)+C(2)+C(3)+C(4)+C(5)+K1
    NVAR3=NVAR2+C(6)
    NVAR4=C(1)+K1
    GO TO 31
29 KM1=K-1
  NS1=0
  DO 79 K4=1,KM1
79 NS1=NS1+P(K4)*J(K4)
    NVAR2=C(1)+C(2)+C(3)+C(4)+C(5)+K1+NS1
    NVAR3=NVAR2+C(6)
    NVAR4=C(1)+K1+NS1
31 DO 28 K2=1,NX1
  NO=NO+1
  L=L+1

```

```

NSTOR2(L,1)=NO
NSTOR2(L,2)=NVAR1
STOR2(L)=1.
L=L+1
NSTOR2(L,1)=NO
NSTOR2(L,2)=NVAR2
STOR2(L)=-1.
IF(LSW(NVAR4).EQ.1) STOR2(L)=1.
NVAR2=NVAR2+J(K)
L=L+1
NSTOR2(L,1)=NO
NSTOR2(L,2)=NVAR3
STOR2(L)=1.
IF(LSW(NVAR4).EQ.1) STOR2(L)=-1.
NVAR3=NVAR3+J(K)
IF(LSW(NVAR4).EQ.0) GO TO 128
L=L+1
NVAR5=NVAR1+1
NSTOR2(L,1)=NO
NSTOR2(L,2)=NVAR5
STOR2(L)=1.
128 NVAR4=NVAR4+J(K)
28 CONTINUE
DO 71 K=1,I
NO=NO+1
IF(NSTAR(K).EQ.1) GO TO 72
NVAR=C(1)+C(2)+C(3)+(K-1)*MM1+NSTAR(K)-1
L=L+1
NSTOR2(L,1)=NO
NSTOR2(L,2)=NVAR
STOR2(L)=1.
IF(NSTAR(K).EQ.R) GO TO 71

```

```

      NX=NSTAR(K)
      NVAR=C(1)+C(2)+C(3)+C(4)+(K-1)*NLEGS+R-NX
      DO 73 K1=NX,MM1
      L=L+1
      NSTOR2(L,1)=NO
      NSTOR2(L,2)=NVAR
      STOR2(L)=1.
73  NVAR=NVAR+MM1-K1+NX
      GO TO 71
72  NVAR=C(1)+C(2)+C(3)+(K-1)*MM1+1
      DO 74 K1=1,MM1
      L=L+1
      NSTOR2(L,1)=NO
      NSTOR2(L,2)=NVAR
      STOR2(L)=-1.
74  NVAR=NVAR+1
      NVAR=C(1)+C(2)+C(3)+C(4)+(K-1)*NLEGS+MM1
      DO 75 K1=1,MM1
      L=L+1
      NSTOR2(L,1)=NO
      NSTOR2(L,2)=NVAR
      STOR2(L)=1
75  NVAR=NVAR+MM1+1-K1
71  CONTINUE
      NVAR=1
      DO 190 K=1,I
      NP=P(K)
      DO 190 K1=1,NP
      NG=NG+1
      NJ=J(K)
      DO 190 K2=1,NJ
      L=L+1

```

```

      NSTOR2(L,1)=NO
      NSTOR2(L,2)=NVAR
      STOR2(L)=1.
190  NVAR=NVAR+1
      NVAR1=C(1)
      NVAR2=C(1)+C(2)+C(3)+C(4)+C(5)
      NVAR3=NVAR2+C(6)
      DO 80 K=1,I
      NP=P(I)
      DO 80 K1=1,NP
      NJ=J(I)
      DO 80 K2=1,NJ
      NVAR1=NVAR1+1
      NVAR2=NVAR2+1
      NVAR3=NVAR3+1
      NO=NO+1
      L=L+1
      NSTOR2(L,1)=NO
      NSTOR2(L,2)=NVAR1
      STOR2(L)=1.
      L=L+1
      NSTOR2(L,1)=NO
      NSTOR2(L,2)=NVAR2
      STOR2(L)=-C1
      L=L+1
      NSTOR2(L,1)=NO
      NSTOR2(L,2)=NVAR3
      STOR2(L)=-1.
80  CONTINUE
      NVAR=C(1)+C(2)+1
      DO 83 K=1,I
      L=L+1

```

```

      NO=NO+1
      NSTOR2(L,1)=NO
      NSTOR2(L,2)=NVAR
      STOR2(L)=1.
83  NVAR=NVAR+(J(K)*2) +1
      LMAX=L
      ISBND=C(4)+C(5)
      NSUM=0
      NS(1)=NSUM
      NS(2)=NSUM+MM1*I+NLEGS*I
      NS(3)=I*R+I+R
      DO 202 K=1,I
      NX=J(K)
      DO 202 K1=1,NX
202  NS(3)=NS(3)+1
      DO 203 K=1,I
203  NS(3)=NS(3)+P(K)*J(K)*2
      NS(3)=NS(3)+I
      DO 201 K=1,I
      NP=P(K)
      DO 201 K1=1,NP
201  NS(3)=NS(3)+1
      NS(3)=NS(3)+I
      NS(4)=LMAX-NS(1)-NS(2)-NS(3)
      N=C(1)+C(2)+C(3)+C(4)+C(5)+C(6)+C(7)
      M=K*I+R+I
      DO 800 K=1,I
      M=M+J(K)
      M=M+P(K)
800  M=M+J(K)*P(K)*2
      M=M+I
      M=M+I

```

```

NUMQ=C(1)+C(2)
IVAR=C(4)+C(5)
M1=M
M2=N
CALL QSTART
CALL BRANCH
STOP
END
FUNCTION A(I,J)
COMMON /AREF/ AA(5000), IROW(276), JCOL(5000), MAXA
ISTART = IROW(I)
LAST = IROW(I+1) - 1
A = 0.0
DO 1 LOOK = ISTART, LAST
  JHERE = JCOL(LOOK)
  IF(JHERE.LT.J) GO TO 1
  IF(JHERE.GT.J) RETURN
  A = AA(LOOK)
RETURN
1 CONTINUE
RETURN
END
SUBROUTINE ADDCON
INTEGER XBASIS, YBASIS, SIZE, SIZE1
COMMON B(275), BOUND(900), C(900), GR(275), INBASE(900), INV(165,165),
KISEFF(275), PIV(900), S(275), SLACK(275), TOL(5), X(900), XBASIS(275), XR
K(275), Y(275), YAC(900), YBASIS(275), YR(275), BIG, DRIVER, INPEV, IR, IRMA
KX, ISBIG, ISBND, ISDNE, ISTATE, ITR, ITRMAX, M, MARKI, MARKK, MAXM, MAXN, MNO
KW, MORE, MOREPR, MXSIZE, N, NEGINV, NEGROW, NEWX, NEWY, NUMSLK, OBJ, R, SIZE, S
KIZE1, SMALL, XKPOS, YAMINC, G(275), IEND
COMMON /AREF/ AA(5000), IROW(276), JCOL(5000), MAXA
REAL INV

```

```

      IF(SIZE1.GT.MXSIZE) GO TO 40
      I = NEWY - SIZE
      DO 10 L = 1,SIZE
      INV(L,SIZE1) = 0.0
10  INV(SIZE1,L) = 0.0
      ISTART = IROW(I)
      LAST = IROW(I+1) - 1
      DO 30 LOOK = ISTART, LAST
      J = JCOL(LOOK)
      IF(INBASE(J).LE.0) GO TO 30
      K = INBASE(J)
      AIJ = AA(LOOK)
      DO 20 L = 1,SIZE
20  INV(SIZE1,L) = INV(SIZE1,L) - AIJ * INV(K,L)
30  CONTINUE
      INV(SIZE1,SIZE1) = 1.0
      XR(SIZE1) = SLACK(I)
      ISEFF(I) = SIZE1
      XBASIS(SIZE1) = I + N
      YBASIS(SIZE1) = I
      YR(SIZE1) = 0.0
      SIZE = SIZE1
      SIZE1 = SIZE1 + 1
      IF(SIZE.GT.ISBIG) ISBIG = SIZE
      NUMSLK = NUMSLK + 1
      NEWY = SIZE
      GO TO 50
40  ISTATE = 4
50  RETURN
      END
      SUBROUTINE BABD
      COMMON /FOUR/ NLEGS,J(4),P(4),C(7),MML,I

```

```

COMMON /SEVEN/ REM(5000),ST2(5000),NST2(5000,2),NSET(500),LVAR
COMMON /TEN/ R1(500),BND(500)
NSTART = C(1)+C(2)+C(3)+1
NLAST=NSTART+C(4)+C(5)-1.
DO 1 K=NSTART,NLAST
WRITE (6,1000) K,NSET(K),BND(K),R1(K)
1000 FORMAT(1H0,2I10,E12.2,F10.2)
1 CONTINUE
RETURN
END
SUBROUTINE BRANCH
COMMON /THREE/ NSTOR1(1000,2),NSTOR2(5000,2),STOR1(1000),STOR2(50
CO),NS(4),LMAX,LQMAX
COMMON /FOUR/ NLEGS,J(4),P(4),C(7),MM1,I
COMMON /FIVE/ W(500),LSW(500)
COMMON /SEVEN/ REM(5000),ST2(5000),NST2(5000,2),NSET(500),LVAR
COMMON /NINE/ NSTAGE
COMMON /TEN/ P1(500),BND(500)
COMMON /TWELVE/ M1,M2,NFES,SOL,IVAR
INTEGER R
INTEGER P
COMMON /Q/ D(100,100),ISQ(900),HOLDC(900),KEYTOD(180),MAXQ,NUMQ
REAL INV
COPT=-10*#10
LVAR=C(1)+C(2)+C(3)+C(4)+C(5)
DO 77 K=1,LVAR
NSET(K)=2
R1(K)=0
77 BND(K)=-10*#10
NSTAGE=C(1)+C(2)+C(3)+1
NSET(NSTAGE)=0
40 CONTINUE

```

```

      NX=C(1)+C(2)+C(3)+C(4)
      IF(NSTAGE.LE.NX) GO TO 2
      GO TO 3
2   NS1=NS(1)+NS(2)+NS(3)+1
      DO 20 K=NS1,LMAX
      IF(NSTOR2(K,2).NE.NSTAGE) GO TO 20
      IF(INSET(NSTAGE).EQ.1) GO TO 21
      REM(K)=STOR2(K)
      STOR2(K)=0.
      GO TO 20
21  STOR2(K)=REM(K)
      NS3=NS(1)+NS(2)+1
      NS2=NS3-1+NS(3)
      DO 22 K1=NS3,NS2
      IF(NSTOR2(K1,1).NE.NSTOR2(K,1)) GO TO 22
      STOR2(K1)=STOR2(K1)-STOR2(K)
      STOR2(K)=0.
22  CONTINUE
20  CONTINUE
      IF(BND(NSTAGE).NE.-10**10) GO TO 29
      IF(INSET(NSTAGE).EQ.1) GO TO 26
25  CALL SOP
      CALL QP
      ISTATE=NFES
      OBJ=SOL
      IF(ISTATE.EQ.2) OBJ=-1.0E12
      AO=OBJ
      WRITE(6,1000) ISTATE,OBJ
1000 FORMAT(1H0,110,E12.2)
      NSET(NSTAGE)=1
      GO TO 2
26  NVAR=C(1)+C(2)+C(3)+1

```

```

A1=0.
DO 30 K=1,I
NSUM=0
N1=NVAR+(K-1)*(MM1+1)
N2=N1+MM1
IF(NSTAGE.LT.N1.OR.NSTAGE.GT.N2) GO TO 30
DO 33 K1=N1,N2
IF(NSET(K1).EQ.1) NSUM=NSUM+1
IF(NSUM.GT.1) A1=-1.0E12
IF(A1.NE.0) GO TO 52
68 CONTINUE
30 CONTINUE
CALL QP
ISTATE=NFES
OBJ=SOL
A1=OBJ
IF(ISTATE.EQ.2) OBJ=-1.0E12
WRITE(6,1000) ISTATE,OBJ
IF(A1.GT.A0) GO TO 28
52 CALL BRBC
NSET(NSTAGE)=0
GO TO 91
28 NSET(NSTAGE)=1
91 CONTINUE
TEST=AMAX1(A1,A0)
IF(TEST.LT.COPT) GO TO 41
X=AMIN1(A1,A0)
BND(NSTAGE)=X
29 CONTINUE
NSTAGE=NSTAGE+1
NSET(NSTAGE)=0
GO TO 40

```

```

3  NS1=NS(1)+NS(2)+NS(3)+1
   DO 32 K=NS1,LMAX
   IF(NSTOR2(K,2).NE.NSTAGE) GO TO 32
   IF(NSET(NSTAGE).EQ.1) GO TO 33
   REM(K)=STOR2(K)
   STOR2(K)=0.
   GO TO 32
33  STOR2(K)=REM(K)
   NS3=NS(1)+NS(2)+1
   NS2=NS3-1+NS(3)
   DO 31 K1=NS3,NS2
   IF(NSTOR2(K1,1).NE.NSTOR2(K,1)) GO TO 31
   STOR2(K1)=STOR2(K1)-STOR2(K)
   STOR2(K)=0.
31  CONTINUE
32  CONTINUE
   IF(BND(NSTAGE).NE.-10**10) GO TO 51
   IF(NSET(NSTAGE).EQ.1) GO TO 34
   CALL SOP
   CALL QP
   ISTATE=NFES
   OBJ=SQL
   IF(ISTATE.EQ.2) OBJ=-1.0E12
   IF(NSTAGE.NE.LVAR) GO TO 35
   IF(OBJ.LT.COPT) GO TO 35
   CALL DATR
   COPT=OBJ
35  AO=OBJ
   IF(NSTAGE.EQ.LVAR) CALL SABD
   WRITE(6,1000) ISTATE,OBJ
   NSET(NSTAGE)=1
   GO TO 3

```

```

34 A1=0.
   NVAR=C(1)+C(2)+C(3)+C(4)+1
   DO 36 K=1,I
   NSUM=0
   N1=NVAR+(K-1)*NLEGS
   N2=N1-1+NLEGS
   IF(NSTAGE.LT.N1.OR.NSTAGE.GT.N2) GO TO 36
   NVAR=N1+MM1
   DO 37 K1=1,MM1
   IF(NSET(NVAR).EQ.1) NSUM=NSUM+1
   IF(NSUM.GT.1) A1=-1.0E12
   IF(A1.NE.0.) GO TO 39
37 NVAR=NVAR+MM1+1-K1
36 CONTINUE
38 CONTINUE
   CALL QP
   ISTATE=NFES
   OBJ=SQL
   IF(ISTATE.EQ.2) OBJ=-1.0E12
   IF(NSTAGE.NE.LVAR) GO TO 27
   IF(OBJ.LT.COPT) GO TO 27
   CALL DATR
   COPT=OBJ
27 A1=OBJ
   IF(NSTAGE.EQ.LVAR) CALL BABD
   WRITE(6,1000) ISTATE,OBJ
   IF(A1.GT.A0) GO TO 42
39 CALL BR8C
   NSET(NSTAGE)=0
   GO TO 90
42 NSET(NSTAGE)=1
90 CONTINUE

```

```

      TEST=AMAX1(A1,A0)
      IF(TEST.LT.COPT) GO TO 41
      X=AMIN1(A1,A0)
      BND(NSTAGE)=X
51  CONTINUE
      NSTAGE=NSTAGE+1
      NSET(NSTAGE)=0
      IF(NSTAGE.LE.LVAR) GO TO 40
41  CONTINUE
      IF(NSTAGE.GT.LVAR) NSTAGE=LVAR
      NBT=0
72  NX=C(1)+C(2)+C(3)+C(4)
      IF(NSTAGE.GT.NX) GO TO 54
      NX=C(1)+C(2)+C(3)
      IF(NSTAGE.GT.NX) GO TO 43
      NCOUNT=0
54  NS1=NS(1)+NS(2)+NS(3)+1
      DO 60 K=NS1,LMAX
      IF(NSTOR2(K,2).NE.NSTAGE) GO TO 60
      STOR2(K)=REM(K)
      IF(NSET(NSTAGE).EQ.0) GO TO 60
      NS3=NS(1)+NS(2)+1
      NS2=NS3-1+NS(3)
      DO 61 K1=NS3,NS2
      IF(NSTOR2(K1,1).NE.NSTOR2(K,1)) GO TO 61
      STOR2(K)=REM(K)
      STOR2(K1)=STOR2(K1)+STOR2(K)
61  CONTINUE
60  CONTINUE
43  NS1=NS(1)+NS(2)+NS(3)+1
      DO 70 K=NS1,LMAX
      IF(NSTOR2(K,2).NE.NSTAGE) GO TO 70

```

```

      STOR2(K)=REM(K)
      IF(NSET(NSTAGE).EQ.0) GO TO 70
      NS3=NS(1)+NS(2)+1
      NS2=NS3-1+NS(3)
      DO 71 K1=NS3,NS2
      IF(NSTOR2(K1,1).NE.NSTOR2(K,1)) GO TO 71
      STOR2(K1)=STOR2(K1)+STOR2(K)
71  CONTINUE
70  CONTINUE
53  CONTINUE
      IF(NBT.GT.0) GO TO 73
      NX=C(1)+C(2)+C(3)+1
      NSTAGE=NSTAGE-1
      COMP=C(1)+C(2)+C(3)
      IF(NSTAGE.EQ.COMP) RETURN
      IF(R1(NSTAGE).EQ.1) GO TO 72
      IF(BND(NSTAGE).LT.CDPT) GO TO 72
      NBT=NBT+1
      GO TO 72
73  CONTINUE
      IF(NSET(NSTAGE).EQ.0) GO TO 80
      NSET(NSTAGE)=0
      GO TO 81
80  NSET(NSTAGE)=1
81  CONTINUE
      R1(NSTAGE)=1
      NP1=NSTAGE+1
      DO 76 K=NP1,LVAR
      BND(K)=-10+*10
      NSET(K)=2
76  R1(K)=0.
      GO TO 40

```

```

END
SUBROUTINE BRBC
COMMON /THREE/ NSTOR1(1000,2),NSTOR2(5000,2),STOR1(1000),STOR2(500
CO),NS(4),LMAX,LQMAX
COMMON /SEVEN/ REM(5000),ST2(5000),NST2(5000,2),NSET(500),LVAR
DO 1 K=1,LMAX
  NSTOR2(K,1)=NST2(K,1)
  NSTOR2(K,2)=NST2(K,2)
1 STOR2(K)=ST2(K)
RETURN
END
SUBROUTINE CHACC
COMMON /AREF/ AA(5000),IROW(276),JCOL(5000),MAXA
INTEGER XBASIS,YBASIS,SIZE,SIZE1
COMMON B(275),BOUND(900),C(900),GR(275),INBASE(900),INV(165,165),
KISEFF(275),PIV(900),S(275),SLACK(275),TOL(8),X(900),XBASIS(275),XR
K(275),Y(275),YAC(900),YBASIS(275),YR(275),BIG,DRIVER,INREV,IR,IRMA
KX,ISBIG,ISBND,ISDONE,ISTATE,ITR,ITRMAX,H,MARKI,MARKK,MAXH,MAXN,MNO
KW,MORE,MOREPR,MXSIZE,N,NEGINV,NEGROW,NEWX,NEWY,NUMSLK,OBJ,R,SIZE,S
KIZE1,SMALL,XKPOS,YAMINC,G(275),IEND
REAL INV
9000 FORMAT (1H0,'UNACCEPTABLE ERROR OF ',F16.3,' FOUND IN B-SLACK-AX 0
1F CONSTRAINT',I6)
9004 FORMAT (1H0,'UNACCEPTABLE RELATIVE ERROR OF ',F16.3,' FOUND IN B-S
1LACK-AX OF CONSTRAINT',I6/1H ,'THE ABSOLUTE ERROR IS ',F16.8,' AND
2 B(I) IS ',F16.3)
9003 FORMAT (1H0,'UNACCEPTABLE ERROR OF ',F16.3,' FOUND IN YA-C OF BASI
1C VARIABLE ',I6)
9012 FORMAT (1H0,'UNACCEPTABLE RELATIVE ERROR OF ',F16.8,' FOUND IN YA-
1C OF BASIC VARIABLE, ',I6/1H ,'THE ABSOLUTE ERROR IS ',F16.8,' AND
2 C(J) IS ',F16.3)
IF(NUMSLK.EQ.0) GO TO 10

```

```

      DO 5 K = 1, SIZE
      IF (XBASIS(K).LE.N) GO TO 5
      I = XBASIS(K) - N
      SLACK(I) = XR(K)
5  CONTINUE
10 DO 20 J = 1, N
      IF (INBASE(J).LE.0) GO TO 20
      YAC(J) = -C(J)
20 CONTINUE
      TOL2 = TOL(2)
      TOL6 = TOL(6)
      DO 40 I = 1, ANOW
      ISEFFI = ISEFF(I)
      YI = Y(I)
      BAXSL = B(I) - SLACK(I)
      ISTART = IROW(I)
      LAST = IROW(I+1) - 1
      DO 30 LOOK = ISTART, LAST
      J = JCOL(LOOK)
      INJ = INBASE(J)
      IF (INJ.EQ.0) GO TO 30
      AIJ = AA(LOOK)
      BAXSL = BAXSL - X(J) * AIJ
      IF (INJ.GT.0.AND.ISEFFI.NE.0) YAC(J) = YAC(J) + YI * AIJ
30 CONTINUE
      ERR = ABS(BAXSL)
      IF (ERR.GT.TOL2) GO TO 60
      ABSB = ABS(B(I))
      IF (ABSB.LT.1.0) ABSB = 1.0
      IF (ERR / ABSB .GT. TOL6) GO TO 65
40 CONTINUE
      TOL7 = TOL(7)

```

```

TOL4 = TOL(4)
DO 50 J = 1,N
IF (IMBASE(J) .LE. 0) GO TO 50
ERR = ABS(YAC(J))
IF (ERR .GT. TOL4) GO TO 70
ABSC = ABS(C(J))
IF (ABSC.LT.1.0) ABSC = 1.0
IF (ERR / ABSC .GT. TOL7) GO TO 75
50 CONTINUE
GO TO 90
60 WRITE (6,9000) ERR,I
GO TO 80
65 RELERR = ERR / ABSB
WRITE (6,9004) RELERR,I,ERR,ABSB
GO TO 80
70 WRITE (6,9008) ERR,J
GO TO 80
75 RELERR = ERR / ABSC
WRITE (6,9012) RELERR,J,ERR,ABSC
80 ISTATE = 7
90 RETURN
END
SUBROUTINE CHBSIS
INTEGER XBASIS,YBASIS,SIZE,SIZE1
COMMON B(275),ROUND(900),C(900),GR(275),INBASE(900),INV(165,165),
KISEFF(275),PIV(900),S(275),SLACK(275),TOL(8),X(900),XBASIS(275),XR
K(275),Y(275),YAC(900),YBASIS(275),YR(275),BIG,DRIVER,INREV,IP,IRMA
KX,ISSIG,ISBND,ISDONE,ISTATE,ITR,ITRMAX,M,MARKI,MARKK,MAXM,MAXN,MNO
KW,MORE,MOREPR,MXSIZE,N,NEGINV,NEGROW,NEWX,NEWY,NUMSLK,OBJ,R,SIZE,S
KIZE1,SMALL,XKPOS,YAMINC,G(275),IFND
REAL INV
ITR = ITR + 1

```

```

MOSNEG = 0
HOWNEG = 0.0
XOFNEG = 0.0
DRITEM = 0.0
TOL1 = TOL(1)
IF(INREV.EQ.1) GO TO 90
IF(R.EQ.0.0) GO TO 40
DO 30 K = 1,SIZE
XR(K) = XR(K) - R * GR(K) * XKPGS
XXX = XR(K)
IF(ABS(XXX).LE.TOL1) XR(K) = 0.0
J = XBASIS(K)
IF(J.LE.N) GO TO 10
XXX = XR(K)
I = J - N
SI = S(I)
IF(SI.EQ.0.0.AND.XXX.GT.0.0.OR.SI.EQ.-1.0) XXX = -XXX
GO TO 20
10 BOUNDJ = BOUND(J)
IF(ABS(BOUNDJ-XXX).LE.TOL1) XR(K) = BOUNDJ
XXX = XR(K)
IF(XXX.LE.BOUNDJ.OR.BOUNDJ.EQ.-1.0) GO TO 20
XXX = BOUNDJ - XXX
20 IF(K.EQ.NEGINV) XOFNEG = XXX
IF(XXX.GE.HOWNEG.OR.K.EQ.NEGINV) GO TO 30
MOSNEG = K
DRITEM = 1.0
IF(XR(K).GE.0.0) DRITEM = -1.0
HOWNEG = XXX
30 CONTINUE
IF(NEWY.NE.-1) GO TO 40
IT = INBASE(NEWX)

```

```

INBASE(NEWX) = -1
IF(IT.EQ.-1) INBASE(NEWX) = 0
IXOUT = NEWX
OBJ = OBJ - R * YAMINC
GO TO 120
40 IXOUT = XBASIS(NEWY)
IF(IXOUT.GT.N) GO TO 50
INBASE(IXOUT) = 0
IF(GR(NEWY) * XKPOS.LT.0.0.AND.NEWY.NE.NEGINV) INBASE(IXOUT) = -1
IF(NEWY.EQ.NEGINV.AND.XR(NEWY).GT.0.0) INBASE(IXOUT) = -1
50 IF(NEWX.GT.N) GO TO 60
IHOLD = INBASE(NEWX)
INBASE(NEWX) = NEWY
60 XBASIS(NEWY) = NEWX
IF(NEWX.GT.N) NUMSLK = NUMSLK + 1
IF(IXOUT.GT.N) NUMSLK = NUMSLK - 1
XR(NEWY) = R
IF(NEWX.LE.N) GO TO 70
I = NEWX - N
IF(S(I).EQ.-1.0) XR(NEWY) = -R
GO TO 80
70 IF(IHOLD.EQ.-1) XR(NEWY) = SOUND(NEWX) - R
80 OBJ = OBJ - R * YAMINC
90 RR = 1.0/ GR(NEWY)
DO 110 L = 1,SIZE
IF(ABS(INV(NEWY,L)).LT.SMALL) GO TO 110
RL = INV(NEWY,L) * RR
DO 100 K = 1,SIZE
INV(K,L) = INV(K,L) - RL * GR(K)
100 CONTINUE
INV(NEWY,L) = RL
IF(INREV.NE.1) YR(L) = YR(L) - RL * YAMINC * XKPOS

```

```

110 CONTINUE
120 IF (R.EQ.0.0 .OR. XOFNEG.LT.0.0 .AND. NEWY.NE.NEGINV) GO TO 130
    NEGINV = MODNEG
    DRIVER = DRITEM
130 RETURN
    END
SUBROUTINE COPY
COMMON /AREF/ AA(5000), IROW(276), JCOL(5000), MAXA
INTEGER XBASIS, YBASIS, SIZE, SIZE1
COMMON B(275), BOUND(900), C(900), GR(275), INBASE(900), INV(165,165)
KISEFF(275), PIV(900), S(275), SLACK(275), TOL(3), X(900), XBASIS(275),
K(275), Y(275), YAC(900), YBASIS(275), YR(275), BIG, DRIVER, INREV, IR, IRI
KX, ISBIG, ISBND, ISDONE, ISTATE, ITR, IT9MAX, M, MARKI, MARKK, MAXM, MAXN, MP
KW, MORE, MOREPR, MXSIZE, N, NEGINV, NEGROW, NEWX, NEWY, NUMSLK, OBJ, R, SIZE,
KIZEL, SMALL, XKPOS, YAMINC, G(275), IEND
REAL INV
IF (MNOW .LE. MAXM) GO TO 5
CALL IEXIT(6)
GO TO 30
5 ISTRIV = 1
IT = -1
IRM = IROW(MNOW)
DO 10 J = 1, N
    AIJ = PIV(J)
    IF(AIJ.EQ.0.0) GO TO 10
    ISTRIV = 0
    IT = IT + 1
    LOOK = IRM + IT
    IF(LOOK.GT.MAXA) CALL IEXIT(6)
    IF(ISDONE.EQ.1) GO TO 30
    AA(LOOK) = AIJ
    JCOL(LOOK) = J

```

```

10 CONTINUE
   IF(ISTRIV.EQ.0) GO TO 20
   LOOK = IRM
   IF(LOOK.GT.MAXA) CALL IEXIT(6)
   IF(ISDONE.EQ.1) GO TO 30
   AA(LOOK) = 0
   JCOL(LOOK) = 1
20 NEXTM = MNOW + 1
   IROW(NEXTM) = LOOK + 1
   NEXTM = MNOW + 2
   IROW(NEXTM) = LOOK + 1
30 RETURN
END
SUBROUTINE CHSLCK
COMMON /AREF/ AA(5000), IROW(276), JCOL(5000), MAXA
INTEGER XBASIS, YBASIS, SIZE, SIZE1
COMMON B(275), BOUND(900), C(900), GR(275), INBASE(900), INV(165,165),
KISEFF(275), PIV(900), S(275), SLACK(275), TOL(8), X(900), XBASIS(275), XR
K(275), Y(275), YAC(900), YBASIS(275), YR(275), BIG, DRIVER, INREV, IR, IRMA
KX, ISBIG, ISBND, ISDONE, ISTATE, ITR, ITRMAX, M, MARKI, MARKK, MAXM, MAXN, MNO
KW, MORE, MOREPR, MXSIZE, N, NEGINV, NEGROW, NEWX, NEWY, NUMSLK, OBJ, P, SIZE, S
KIZE1, SMALL, XKPOS, YAMINC, G(275), IEND
REAL INV
IF (R .NE. 0.0) NEGROW = 0
HOWNEG = 0.0
DO 10 J = 1, N
YACJ = 0.0
K = INBASE(J)
IF(K.LE.0) YACJ = -C(J)
YAC(J) = YACJ
XJ = 0.0
IF(K.EQ.-1) XJ = BOUND(J)

```

```

      IF(K.GT.0) XJ = XR(K)
10  X(J) = XJ
      TOL2 = TOL(2)
      DO 70 I = 1,MNCW
      L = ISEFF(I)
      Y(I) = 0.0
      IF(L.EQ.0) GO TO 30
      YI = YR(L)
      Y(I) = YI
      SLACK(I) = 0.0
      LAST = IROW(I+1) -1
      ISTART = IROW(I)
      DO 20 LOOK = ISTART, LAST
      J = JCOL(LOOK)
      IF(INBASE(J).GT.0) GO TO 20
      AIJ = AA(LOOK)
      YAC(J) = YAC(J) + YI * AIJ
20  CONTINUE
      GO TO 70
30  IF(INREV.NE.1) GO TO 50
      SLKI = B(I)
      DO 40 J = 1,N
      IF(INBASE(J).EQ.0) GO TO 40
      SLKI = SLKI - A(I,J) * X(J)
40  CONTINUE
      GO TO 60
50  IF(R.EQ.0.0) GO TO 70
      SLKI = SLACK(I) - R * G(I) * XKPOS
60  IF(ABS(SLKI).LE.TOL2) SLKI = 0.0
      SLACK(I) = SLKI
      IF(NEGINV.NE.0) GO TO 70
      SI = S(I)

```

```

      ABSLKI = ABS(SLKI)
      IF(SI .NE. 0.0 .AND. SI*SLKI .GE. HOWNEG .OR. SI .EQ. 0.0 .AND.
1    -ABSLKI .GE. HOWNEG) GO TO 70
      HOWNEG = -ABSLKI
      NEGROW = I
70    CONTINUE
      INREV = 0
      IF (MARKI .NE. 0) SLACK(MARKI) = XR(MARKK)
      RETURN
      END
      SUBROUTINE DATA
      COMMON /AREF/ AA(5000), IROW(276), JCOL(5000), MAX4
      INTEGER XBASIS, YBASIS, SIZE, SIZE1
      COMMON B(275), BOUND(900), C(900), GR(275), INBASE(900), INV(165,165),
      KISEFF(275), PIV(900), S(275), SLACK(275), TOL(8), X(900), XBASIS(275), XR
      K(275), Y(275), YAC(900), YBASIS(275), YR(275), BIG, DRIVER, INREV, IP, IRMA
      KX, ISBIG, ISBND, ISDONE, ISTATE, ITR, ITRMAX, M, MARKI, MARKK, MAXM, MAXN, MNO
      KW, MORE, MOREPR, MXSIZE, N, NEGINV, NEGROW, NEWX, NEWY, NUMSLK, OBJ, R, SIZE, S
      KIZE1, SMALL, XKPOS, YAMINC, G(275), IEND
      REAL INV
      COMMON /ONE/ NSTOR3(5000,2), STOR3(5000)
      REAL K3
      COMMON /THREE/ NSTOR1(1000,2), NSTOR2(5000,2), STOR1(1000), STOR2(500
      C), NS(4), LMAX, LQMAX
      COMMON /NINE/ NSTAGE
      DIMENSION K1(3), K2(3), K3(3)
9020 FORMAT(1H0, 'YOU ARE TRYING TO SOLVE A PROBLEM WHICH IS TOO BIG FOR
      1 THE PROGRAM.'/1X, 'THE LARGEST PROBLEM HAS', I5, ' CONSTRAINTS AND',
      2I5, ' VARIABLES.')
9024 FORMAT(1H0, 'IT IS NOT POSSIBLE TO HAVE A NEGATIVE NUMBER OF ROWS
      1 OR COLUMNS.')
9030 FORMAT(1H0, 'INPUT CARDS FOR THE C ELEMENTS . . . .')

```

```

9034 FORMAT (1H0,'INPUT CARDS FOR THE UPPER BOUNDS ON SINGLE VARIABLES
1. . . .')
9038 FORMAT (1H0,'INPUT CARDS FOR THE B VECTOR, THE RIGHT-HAND SIDES OF
1 THE CONSTRAINTS . . . .')
9042 FORMAT (1H0,'INPUT CARDS FOR THE ROWS OF THE A MATRIX . . . .')
9064 FORMAT (1H , 'CARD ',I4)
9068 FORMAT(1H ,4('(',I3,'/',I1,'/',F12.5,')')/1H ,4('(',I3,'/',I1,'/',
1F12.5,')'))
9070 FORMAT(1H0,'YOU HAVE SPECIFIED AN ELEMENT FOR VARIABLE NO. ',I5,
1' WHICH IS GREATER THAN N, ',I5,'.'/' IT HAS BEEN IGNORED.')
9074 FORMAT (1H0,'YOU HAVE SPECIFIED AN ELEMENT FOR ROW NO. ',I5,', WHI
1CH IS GREATER THAN M, ',I5,'.'/' IT HAS BEEN IGNORED.')
9078 FORMAT (1H0,'THE ELEMENT IN THE 4TH COLUMN OF EACH FIELD OF 10 CAN
1 ONLY BE 1 FOR A LESS-THAN-OR-EQUAL CONSTRAINT, OR 0 '/' FOR AN EQ
2UALITY, OR 2 FOR GREATER-THAN-OR EQUAL. YOU HAVE A ',I5,'.'/'1X,
3'IT HAS BEEN IGNORED')
9082 FORMAT (1H0,'YOU ARE TRYING TO READ ROW NO.',I5,' WHICH IS MORE TH
1AN YOUR SPECIFIED NO. OF ROWS,',I5,','')
9084 FORMAT (1H0,'YOUR PREVIOUS ROW WAS ',I5,', AND YOU ARE NOW TRYING
1TO READ ROW NO.',I6,'.'/' IT IS ESSENTIAL THAT THE ROWS OF THE A M
1ATRIX ARE ENTERED IN THE RIGHT ORDER.')
9092 FORMAT(1H0,'**** WARNING **** YOU HAVE NOT SPECIFIED A RIGHT HAND
1SIDE FOR ALL THE CONSTRAINTS.'/' ANY NOT SPECIFIED HAVE BEEN ASSU
2MED TO BE A LARGE NUMBER')
9096 FORMAT(1H0,'**** YOU HAVE SPECIFIED THAT THERE ARE',I5,' ROWS AND
1YOU HAVE TRIED TO READ IN ELEMENTS OF A FOR',I5,' ROWS')
9098 FORMAT (1H0,'***** THERE IS A FATAL ERROR IN YOUR DATA INPUT ****
1**')
BIG = 1.0E11
SMALL = 1.0E-9
MXSIZE=275
MAXA=5000

```

```

MAXM=275
MAXN=900
CALL STFRM
ISDONE = 0
ICARD = 1
KC = 0
KB = 0
MOREPR=3
ITRMAX=200
IRMAX=3
IF(ITRMAX.LE.0) GO TO 10
GO TO 30
10 IJK = ISBND
   IF (IJK.EQ.-1) IJK = N
   ITRMAX = 3 * (M + N + IJK)
30 CONTINUE
   IF(M.LE.MAXM.AND.N.LE.MAXN.AND.ISBND.LE.MAXN) GO TO 35
   WRITE(6,9020) MAXM,MAXN
   ISDONE = 1
   GO TO 610
35 IF(M.GE.0.AND.N.GE.0 .AND. ISBND .GE. -1) GO TO 40
   WRITE(6,9024)
   ISDONE = 1
   GO TO 610
40 ANDW = 1
   IROW(1) = 1
   ISTEP = 1
   BNDJ = -1.0
   IF(ISBND.EQ.-1) BNDJ = 1.0
   DO 50 J = 1,N
   C(J) = 0.0
   PIV(J) = 0.0

```

```

50 BOUND(J) = BNDJ
   DO 60 I = 1,M
     B(I) = BIG
60  S(I) = 1.0
   IF(MOREPR.LE.1) WRITE(6,9030)
   I1=1
   NSUM=NS(I)
   L=0
   NSWIT=0
   GO TO 3
100 IF(NSWIT.EQ.1) GO TO 3
   DO 1 N1=1,8
     L=L+1
     IF (L.GT.NSUM) GO TO 2
     IF(I1.LE.3) GO TO 5
     IF(N1.NE.1) GO TO 6
     K3(I)=NSTOR3(L,1)
     L=L-1
     K1(I)=0
     K2(I)=0
     GO TO 8
6   IF(NSTOR3(L,1).NE.K3(I))GO TO 2
     K1(N1)=NSTOR3(L,2)
     K2(N1)=0
     K3(N1)=STOR3(L)
     GO TO 8
5   K1(N1)=NSTOR3(L,1)
     K2(N1)=NSTOR3(L,2)
     K3(N1)=STOR3(L)
     GO TO 8
2   K1(N1)=0
     L=L-1

```

```

      K2(N1)=0
      K3(N1)=0
      NSWIT=1
8    CONTINUE
      IF(I1.EQ.4.AND.L.LT.LMAX) NSWIT=0
1    CONTINUE
      GO TO 4
3    K1(1)=999
      I1=I1+1
      IF(I1.LE.4) NSUM=NSUM+NS(I1)
      NSWIT=0
4    CONTINUE
      NQMAX=L
      ICARD = ICARD + 1
      IF(MOREPR.GE.2) GO TO 120
      WRITE(6,9064) ICARD
      WRITE(6,9068) (K1(JK),K2(JK),K3(JK),JK=1,8)
120  IF(K1(1) .NE. 999) GO TO 130
      ISTEP = ISTEP + 1
      IF(ISTEP.EQ.2.AND.ISBND.LE.0) ISTEP = 3
      IF(ISTEP.EQ.5) GO TO 600
      IF(MOREPR.GE.2) GO TO 100
      IF (ISTEP .EQ. 2) WRITE(6,9034)
      IF(ISTEP .EQ. 3) WRITE(6,9038)
      IF (ISTEP .EQ. 4) WRITE (6,9042)
      GO TO 100
130  IF(ISCONE.EQ.1) GO TO 100
      GO TO (200,300,400,500),ISTEP
200  DO 210 JK = 1,8
      J = K1(JK)
      IF (J .LE. 0) GO TO 210
      IF (J .LE. N) GO TO 205

```

```

        WRITE (6,9070) J,N
        GO TO 210
205 C(J) = K3(JK)
        KC = KC + 1
210 CONTINUE
        GO TO 100
300 DO 320 JK = 1,8
        J = K1(JK)
        IF (J .LE. 0) GO TO 320
        IF (J .LE. N) GO TO 315
        WRITE (6,9070) J,N
        GO TO 320
315 BOUND(J) = K3(JK)
320 CONTINUE
        GO TO 100
400 DO 420 JK = 1,8
        I = K1(JK)
        IF (I .LE. 0) GO TO 420
        IF (I .LE. M) GO TO 405
        WRITE (6,9074) I,M
        GO TO 420
405 SI = K2(JK)
        IF (SI .EQ. 1.0 .OR. SI .EQ. 2.0 .OR. SI .EQ. 0.0) GO TO 410
        WRITE (6,9079) K2(JK)
        GO TO 420
410 IF (SI .EQ. 2.0) SI = -1.0
        S(I) = SI
        B(I) = K3(JK)
        KB = KB + 1
420 CONTINUE
        GO TO 100
500 I = K3(1)

```

```

510 IF(I.EQ.MNOW) GO TO 540
    IF (I .LE. M) GO TO 515
    WRITE (6,9032) I,M
    MOREPR = 0
    ISDONE = 1
    GO TO 100
515 IF (I .GT. MNOW) GO TO 520
    WRITE (6,9034) MNOW,I
    MOREPR = 0
    ISDONE = 1
    GO TO 100
520 CALL COPY
    IF(ISDONE.NE.1) GO TO 525
    MOREPR = 0
    GO TO 100
525 DO 530 J = 1,N
530 PIV(J) = 0.0
    MNOW = MNOW + 1
    GO TO 510
540 DO 550 JK = 2,3
    J = K1(JK)
    IF (J .LE. 0) GO TO 550
    IF (J .LE. N) GO TO 545
    WRITE (6,9070) J,N
    GO TO 550
545 PIV(J) = K3(JK)
550 CONTINUE
    GO TO 100
600 IF(ISDONE.EQ.0) CALL COPY
    IF(ISDONE.EQ.1) GO TO 610
    INREV = 0
    IR = 0

```

```

      ITR = 0
      NEGINV = 0
      NEGROW = 0
      NEWX = 0
      NEWY = 0
      R = 0.0
      SIZE = 0
      ISBIG = 1
      TOL(1)=.01
      TOL(2)=.01
      TOL(3)=.01
      TOL(4)=.01
      TOL(5)=.001
      TOL(6)=.01
      TOL(7)=.01
      TOL(3) = TOL(5) = 10.0
      Y4INC = 0.0
      MORE=0
      IF(KB.LT.M) WRITE(6,9092)
      IF(MNOW.EQ.M) GO TO 610
      WRITE(6,9096) M,MNOW
      ISDONE = 1
610  IF (ISDONE .EQ. 1) WRITE (6,9098)
      RETURN
      END
      SUBROUTINE DATR
      COMMON /FOUR/ NLEGS,J(4),P(4),C(7),MM1,I
      COMMON /THIR/ J1(4),C1(7),MM11,I1,NLEG
      NLEG=NLEGS
      I1=I
      MM11=MM1
      DO 1 K=1,I

```

```

      J1(K)=J(K)
1  CONTINUE
      DO 2 K=1,7
      C1(K)=C(K)
2  CONTINUE
      CALL SSO
      RETURN
      END
      SUBROUTINE DOAQP
      INTEGER XBASIS,YBASIS,SIZE,SIZE1
      COMMON B(275),BOUND(900),C(900),GR(275),INBASE(900),INV(165,165),
      KISEFF(275),PIV(900),S(275),SLACK(275),TOL(3),X(900),XBASIS(275),XR
      K(275),Y(275),YAC(900),YBASIS(275),YR(275),BIG,DRIVER,INREV,IR,IRMA
      KX,ISSIG,ISBND,ISDCNE,ISTATE,ITR,ITRMAX,M,MARKI,MARKK,MAXM,MAXN,MNO
      KW,MORE,MOREPR,MXSIZE,N,NEGINV,NEGROW,NEWX,NEWY,NUMSLK,OBJ,R,SIZE,S
      KIZE1,SMALL,XKPOS,YAMINC,G(275),IEND
      REAL INV
      PREOBJ = -BIG
      IF (ISTATE .EQ. 11) GO TO 20
      CALL FIRST3
10  CALL CHSLCK
      IF (ITR .LE. ITRMAX) GO TO 20
      ISTATE = 5
      GO TO 100
20  IF (NEGROW .EQ. 0 .AND. NEGINV .EQ. 0) GO TO 40
      IF (NEGINV .NE. 0) GO TO 30
      NEWY = NEGROW + SIZE
      DRIVER = 1.0
      IF (SLACK(NEGROW) .GT. 0.0) DRIVER = -1.0
      NEGINV = SIZE1
      CALL ADDCON
      IF (ISTATE.EQ.4) GO TO 100

```

```

30 CALL SEEKX
   IF (NEWX .NE. 0) GO TO 70
   ISTATE = 2
   GO TO 100
40 CALL PURGE
   CALL PRICE
   IF (OBJ .GT. PREOBJ .OR. R .EQ. 0.0 .OR. MNOW .GT. M) GO TO 60
   NEWX = IXOUT
   IF (NEWX .GT. N) GO TO 50
   YAMINC = YAC(NEWX)
   IF (INBASE(NEWX) .EQ. -1) YAMINC = -YAMINC
   GO TO 70
50 I = NEWX - N
   YAMINC = Y(I) * S(I)
   GO TO 70
60 CALL ISOPT
   IF (NEWX .NE. 0) GO TO 70
   ISTATE = 9
   GO TO 100
70 CALL NEWVEC
   IF (NEGROW .EQ. 0 .AND. NEGINV .EQ. 0 .AND. OBJ .LE. PREOBJ .OR.
1    MNOW .GT. M) CALL QCON
   IF (ISDONE .EQ. 1) GO TO 100
   IF (NEGROW .EQ. 0 .AND. NEGINV .EQ. 0) PREOBJ = OBJ
   CALL SEEKY
   IF (NEWY .NE. 0) GO TO 80
   ISTATE = 3
   GO TO 100
80 IF (NEWY .LE. SIZE) GO TO 90
   CALL ADDCON
   IF (ISTATE.EQ.4) GO TO 100
90 IXOUT = NEWX

```

```

      IF (NEWY .NE. -1) IXOUT = XBASIS(NEWY)
      CALL CHBSIS
      CALL REDUCE
      GO TO 10
100 RETURN
      END
      SUBROUTINE FIRST0
      INTEGER XBASIS,YBASIS,SIZE,SIZE1
      COMMON B(275),BOUND(900),C(900),GR(275),INBASE(900),INV(165,165),
      KISEFF(275),PIV(900),S(275),SLACK(275),TOL(8),X(900),XBASIS(275),XR
      K(275),Y(275),YAC(900),YBASIS(275),YR(275),BIG,DRIVER,INREV,IR,IRMA
      KX,ISBIG,ISAND,ISDONE,ISTATE,ITR,ITRMAX,M,MARKI,MARKK,MAXM,MAXN,MNO
      KW,MORE,MOREPR,MXSIZE,N,NEGINV,NEGROW,NEWX,NEWY,NUMSLK,OBJ,R,SIZE,S
      KIZE1,SMALL,XXPOS,YAMINC,G(275),IEND
      REAL INV
      DO 10 J = 1,N
10  INBASE(J) = 0
      DO 20 I = 2,MNOW
20  ISEFF(I) = 0
      DRIVER = 0.0
      NEGINV = 0
      SS = S(1)
      BB = B(1)
      IF(SS .EQ. 1.0 .AND. BB .GE. 0.0 .OR. SS .EQ. -1.0 .AND. BB .LE.
10.0 .OR. SS .EQ. 0.0 .AND. BB .EQ. 0.0) GO TO 30
      NEGINV = 1
      DRIVER = 1.0
      IF(BB.GT.0.0) DRIVER = -1.0
30  SIZE = 1
      SIZE1 = 2
      NEWX = N + 1
      XBASIS(1) = N + 1

```

```

INV(1,1) = 1.0
XR(1) = BB
OBJ = 0.0
YR(1) = 0.0
YBASIS(1) = 1
ISEFF(1) = 1
NUMSLK = 1
MARKI = 1
MARKK = 1
ITR = ITR + 1
INREV = 1
RETURN
END
SUBROUTINE IEXIT(JK)
COMMON /AREF/ AA(5000), IROW(275), JCOL(5000), MAXA
INTEGER XBASIS, YBASIS, SIZE, SIZE1
COMMON B(275), BOUND(900), C(900), GR(275), INBASE(900), INV(165,165),
XISEFF(275), PIV(900), S(275), SLACK(275), TOL(3), X(900), XPBASIS(275), XR
K(275), Y(275), YAC(900), YBASIS(275), YR(275), BIG, DRIVER, INREV, IR, IRMA
KX, ISBIG, ISBND, ISDONE, ISTATE, ITR, ITRMAX, M, MARKI, MARKK, MAXM, MAXN, MNO
KW, MORE, MOREPR, MXSIZE, N, NEGINV, NEGROW, NEWX, NEWY, NUMSLK, OBJ, R, SIZE, S
KIZE1, SMALL, XKPOS, YAMINC, G(275), IEND
REAL INV
9001 FORMAT (1H0,'OPTIMUM')
9002 FORMAT (1H0,'INFEASIBLE.')
9003 FORMAT (1H0,'UNBOUNDED.')
9004 FORMAT (1H0,'THE MAXIMUM SIZE OF THE INVERSE HAS BEEN EXCEEDED')
9005 FORMAT (1H0,'THE MAXIMUM NUMBER OF ITERATIONS HAS BEEN REACHED')
9006 FORMAT (1H0,'EITHER THE AA VECTOR IS FULL WITH',I6,' ELEMENTS, OR
1THE NUMBER OF CONSTRAINTS IS ABOUT TO EXCEED',I6,'.'/1X,'THE PROGR
2AM IS TRYING TO COPY ROW',I6,' WHICH IS AS FOLLOWS...')
9007 FORMAT (1H0,'STILL INACCURATE AFTER ',I5,' REINVERSIONS.')
```

```

9008 FORMAT (1H0,'INTEGER PROGRAM OPTIMUM.')
9009 FORMAT (1H0,'QUADRATIC PROGRAM OPTIMUM.')
9013 FORMAT (1H0,'THE VECTOR GR IS AS FOLLOWS...')
9014 FORMAT (1H0,'THE VECTOR G IS AS FOLLOWS...')
9015 FORMAT (//////)
      IF(JK.EQ.2.OR.JK.EQ.3) MOREPR = 1
      WRITE (6,9015)
      ISDONE = 1
      GO TO (1,2,3,4,5,6,7,8,9),JK
1  WRITE(6,9001)
   GO TO 10
2  WRITE (6,9002)
   GO TO 10
3  WRITE (6,9003)
   WRITE (6,9013)
   WRITE (6,9014)
   GO TO 10
4  WRITE (6,9004)
   GO TO 10
5  WRITE (6,9005)
   GO TO 10
6  WRITE (6,9006) MAXA,MAXM,MNOW
   GO TO 10
7  WRITE (6,9007) IR
   GO TO 10
8  WRITE (6,9008)
   GO TO 10
9  WRITE (6,9009)
10 RETURN
   END
   SUBROUTINE ISOPT
   INTEGER XBASIS,YBASIS,SIZE,SIZE1

```

```

COMMON B(275),BCOND(900),C(900),GR(275),INBASE(900),INV(165,165),
KISEFF(275),PIV(900),S(275),SLACK(275),TOL(3),X(900),XBASIS(275),X
K(275),Y(275),YAC(900),YBASIS(275),YR(275),BIG,DRIVER,INREV,IR,IRM
KX,ISBIG,ISBND,ISDONE,ISTATE,ITR,ITRMAX,M,MARKI,MARKK,MAXM,MAXN,MN
KW,MORE,MOREPR,MXSIZE,N,NEGINV,NEGROW,NEWX,NEWY,NUMSLK,OBJ,R,SIZE,
KIZEL,SMALL,XKPOS,YAMINC,G(275),IEND
REAL INV
YAMINC = - TOL(3)
NEWX = 0
DO 10 L = 1,SIZE
I = YBASIS(L)
SI = S(I)
IF(SI.EQ.0.0) GO TO 10
YRL = YR(L) * SI
IF(YRL.GE.YAMINC) GO TO 10
YAMINC = YRL
NEWX = I + N
10 CONTINUE
TOL4 = TOL(4)
DO 20 J = 1,N
INBJ = INBASE(J)
IF(INBJ.GT.0.OR.BCOND(J).EQ.0.0) GO TO 20
T = YAC(J)
IF(ABS(T).LE.TOL4) T = 0.0
YAC(J) = T
IF(INBJ.EQ.-1) T = -T
IF(T.GE.YAMINC) GO TO 20
YAMINC = T
NEWX = J
20 CONTINUE
RETURN
END

```

```

SUBROUTINE LOUT
COMMON /THREE/ NSTOR1(1000,2),NSTOR2(5000,2),STOR1(1000),STOR2(500
CO),NS(4),LMAX,LQMAX
DO 635 L=1,LMAX
WRITE (6,736) NSTOR2(L,1),NSTOR2(L,2),STOR2(L)
635 CONTINUE
DO 637 L=1,LQMAX
WRITE (6,736) NSTOR1(L,1),NSTOR1(L,2),STOR1(L)
637 CONTINUE
736 FORMAT(1H0,2I3,F8.2)
RETURN
END
SUBROUTINE NEWVEC
INTEGER XBASIS,YBASIS,SIZE,SIZE1
COMMON B(275),BGUND(900),C(900),GR(275),INBASE(900),INV(165,165),
KBASEFF(275),PIV(900),S(275),SLACK(275),TOL(3),X(900),XBASIS(275),XR
K(275),Y(275),YAC(900),YBASIS(275),YR(275),BIG,DRIVER,INREV,IR,IRMA
KX,ISBIG,ISBND,ISDONE,ISTATE,ITR,ITRMAX,M,MARKI,MARKK,MAXM,MAXN,MNO
KW,YCRE,MOREPR,MXSIZE,N,NEGINV,NEGROW,NEWX,NEWY,NUMSLK,CBJ,R,SIZE,S
KIZE1,SMALL,XKPOS,YAMINC,G(275),IEND
REAL INV
XKPOS = 1.0
IF(NEWX.GT.N) GO TO 40
DO 10 K = 1,SIZE
10 GR(K) = 0.0
DO 30 L = 1,SIZE
I = YBASIS(L)
AIJ = A(I,NEWX)
IF(AIJ.EQ.0.0) GO TO 30
DO 20 K = 1,SIZE
20 GR(K) = GR(K) + AIJ * INV(K,L)
30 CONTINUE

```

```

      IF(INBASE(NEWX).EQ.-1) XKPCS = -1.0
      GO TO 60
40  I = NEWX - N
      L = ISEFF(I)
      DO 50 K = 1,SIZE
50  GR(K) = INV(K,L)
      IF(S(I).EQ.-1.0) XKPCS = -1.0
60  RETURN
      END
      SUBROUTINE PRICE
      COMMON /AREF/ AA(5000), IROW(276), JCOL(5000), MAXA
      COMMON /Q/ D(100,100), ISQ(900), HOLDC(900), KEYTOD(180), MAXQ, NUMQ
      COMMON /RETAIN/ KEEP(275)
      INTEGER XBASIS, YBASIS, SIZE, SIZE1
      COMMON B(275), BOUND(900), C(900), GR(275), INBASE(900), INV(165,165)
      KISEFF(275), PIV(900), S(275), SLACK(275), TOL(8), X(900), XBASIS(275),
      K(275), Y(275), YAC(900), YBASIS(275), YR(275), BIG, DRIVER, INREV, IR, IR,
      KX, ISBIG, ISBND, ISDONE, ISTATE, ITR, ITRMAX, M, MARKI, MARKK, MAXM, MAXN, M
      KW, MORE, MOREPR, MXSIZE, N, NEGINV, NEGROW, NEWX, NEWY, NUMSLK, CSJ, R, SIZE
      KIZEL, SMALL, XKPCS, YAMING, G(275), IEND
      REAL INV
      DO 5 J = 1, N
      YAC(J) = 0.0
      IF (INBASE(J) .LE. 0) YAC(J) = -C(J)
5  CONTINUE
      DO 20 ID = 1, NUMQ
      J = KEYTOD(ID)
      CJ = HOLDC(J)
      DO 10 JD = 1, NUMQ
      JJ = KEYTOD(JD)
      XJ = X(JJ)
      IF (XJ .EQ. 0.0) GO TO 10

```

```

      CJ = CJ + XJ * D(ID,J0)
10  CONTINUE
      IF (ABS(CJ) .LE. SMALL) CJ = 0.0
      YAC(J) = 0.0
      IF (INBASE(J) .LE. 0) YAC(J) = -CJ
20  C(J) = CJ
      DO 30 I = 1,MNOW
      IF (SLACK(I) .NE. 0.0) KEEP(I) = 0
30  Y(I) = 0.0
      DO 70 L = 1,SIZE
      YRL = 0.0
      DO 40 K = 1,SIZE
      J = XBASIS(K)
      IF (J .GT. N) GO TO 40
      YRL = YRL + C(J) * INV(K,L)
40  CONTINUE
      IF (ABS(YRL) .LE. SMALL) YRL = 0.0
      I = YBASIS(L)
      IF (I .LE. M) GO TO 50
      IF (S(I) * YRL .GT. 0.0) S(I) = -S(I)
50  YR(L) = YRL
      Y(I) = YRL
      LAST = IROW(I+1) - 1
      ISTART = IROW(I)
      DO 60 LOOK = ISTART, LAST
      JJ = JCOL(LOOK)
      IF (INBASE(JJ) .GT. 0) GO TO 60
      AIJ = AA(LOOK)
      YAC(JJ) = YAC(JJ) + YRL * AIJ
60  CONTINUE
70  CONTINUE
      OBJ = 0.0

```

```

      DO 80 J = 1,N
      XJ = X(J)
      IF (XJ.EQ. 0.0) GO TO 80
      OBJ = OBJ + (C(J) + HOLDC(J)) * XJ
80  CONTINUE
      OBJ = OBJ * 0.5
      RETURN
      END
      SUBROUTINE PURGE
      COMMON /AREF/ AA(5000), IROW(276), JCOL(5000), MAXA
      COMMON /RETAIN/ KEEP(275)
      INTEGER XBASIS, YBASIS, SIZE, SIZE1
      COMMON B(275), BOUND(900), C(900), GR(275), INBASE(900), INV(165,165),
      K(ISEFF(275),PIV(900),S(275),SLACK(275),TOL(3),X(900),XBASIS(275),)
      K(275),Y(275),YAC(900),YBASIS(275),YR(275),BIG,DRIVER,INPEV,IP,IR
      KX,ISBIG,ISBND,ISDONE,ISTATE,ITR,ITRMAX,H,MARKI,MARKK,MAXM,MAXN,MA
      KW,MORE,MOREPR,MAXSIZE,N,NEGINV,NEGROW,NEWX,NEWY,NUMSLK,CRJ,R,SIZE,
      KIZEL,SMALL,XKPOS,YAMINC,G(275),IEND
      REAL INV
      MOVE = 0
      MSTART = MNOW
      M1 = M + 1
      IF (M1.GT.MSTART) GO TO 50
      DO 40 I = M1,MSTART
      I1 = I
      IF (ISEFF(I).NE.0 .OR. KEEP(I).NE.0) GO TO 40
      ILEFT = I - 1
10  MNOW = MNOW - 1
20  I1 = I1 + 1
      IF (I1.GT.MSTART) GO TO 50
      IF (ISEFF(I1).EQ.0 .AND. KEEP(I1).EQ.0) GO TO 10
      ILEFT = ILEFT + 1

```

```

NEXT = IPDW(I1)
MOVE = NEXT - IROW(ILEFT)
LASTIN = IROW(I1+1) - 1
DO 30 IJ = NEXT, LASTIN
  IJNOW = IJ - MOVE
  AA(IJNOW) = AA(IJ)
30 JCOL(IJNOW) = JCOL(IJ)
  LENGTH = LASTIN - NEXT + 1
  INEXT = ILEFT + 1
  IROW(INEXT) = IROW(ILEFT) + LENGTH
  B(ILEFT) = B(I1)
  Y(ILEFT) = Y(I1)
  SLACK(ILEFT) = SLACK(I1)
  S(ILEFT) = S(I1)
  KEEP(ILEFT) = KEEP(I1)
  ISEFF(ILEFT) = ISEFF(I1)
  L = ISEFF(ILEFT)
  IF(L.NE.0) YBASIS(L) = ILEFT
  GO TO 20
40 CONTINUE
50 RETURN
END
SUBROUTINE QCON
COMMON /RETAIN/ KEEP(275)
COMMON /Q/ D(100,100),ISQ(900),HOLDQ(900),KEYTCD(180),MAXQ,NUMQ
INTEGER XBASIS,YBASIS,SIZE,SIZE1
COMMON B(275),BCUND(900),C(900),GR(275),INBAS(900),INV(165,165),
KISEFF(275),PIV(900),S(275),SLACK(275),TOL(3),X(900),XBASIS(275),XR
K(275),Y(275),YAC(900),YBASIS(275),YR(275),BIG,DRIVER,INFEV,IR,IRMA
KX,ISBIG,ISBND,ISDCNE,ISTATE,ITR,ITPMAX,M,MARKI,MARKK,MAXM,MAXN,MNO
KW,MORE,MOREPR,MXSIZE,N,NEGINV,NEGROW,NEWX,NEWY,NUMSLK,OBJ,R,SIZE,S
KIZE1,SMALL,XKPCS,YAMINC,G(275),IEND

```

```

PEAL INV
MNOW = MNOW + 1
DO 10 J = 1,N
10 PIV(J) = 0.0
BMNOW = 0.0
IF (NEWX .GT. N) GO TO 30
BMNOW = HOLDC(NEWX) * XKPCS
IDNEWX = ISQ(NEWX)
IF (IDNEWX .EQ. 0) GO TO 30
DO 20 JD = 1,NUMQ
J = KEYTOD(JD)
20 PIV(J) = -D(IDNEWX,JD) * XKPOS
30 DO 50 K = 1,SIZE
J = XBASIS(K)
IF (J .GT. N) GO TO 50
BMNOW = BMNOW - GR(K) * HOLDC(J) * XKPOS
ID = ISQ(J)
IF (ID .EQ. 0) GO TO 50
DO 40 JD = 1,NUMQ
J = KEYTOD(JD)
40 PIV(J) = PIV(J) + GR(K) * D(ID,JD) * XKPOS
50 CONTINUE
DO 60 J = 1,N
IF (ABS(PIV(J)) .LE. SMALL) PIV(J) = 0.0
60 CONTINUE
IF (ABS(BMNOW) .LE. SMALL) BMNOW = 0.0
H(MNOW) = BMNOW
Y(MNOW) = 0.0
SLACK(MNOW) = - YAMINC
S(MNOW) = 1.0
ISEFF(MNOW) = 0
KEEP(MNOW) = 1

```

```

CALL COPY
RETURN
END
SUBROUTINE QP
  INTEGER XBASIS,YBASIS,SIZE,SIZE1
  COMMON B(275),BOUND(900),C(900),GR(275),INBASE(900),INV(165,165),
  KISEFF(275),PIV(900),S(275),SLACK(275),TOL(8),X(900),XBASIS(275),
  K(275),Y(275),YAC(900),YBASIS(275),YR(275),BIG,DRIVER,INREV,IR,IRP,
  KX,ISBIG,ISBND,ISDONE,ISTATE,ITR,ITRMAX,M,MARKI,MARKK,MAXM,MAXN,MN,
  KW,MORE,MOREPR,MXSIZE,N,NEGINV,NEGROW,NEWX,NEWY,NUMSLK,OBJ,R,SIZE,
  KIZE1,SMALL,XKPOS,YAMINC,G(275),IEND
  COMMON /TWELVE/ M1,M2,NFES,SOL,IVAR
  REAL INV
  ISBND=IVAR
  M=M1
  N=M2
10 CALL QPDATA
  IF (ISDONE.EQ.1) GO TO 60
  ISTATE = 0
20 CALL QDAQP
  IF (ISDONE .EQ. 1) GO TO 60
  IF (ISTATE .GT. 3 .AND. ISTATE .NE. 9) GO TO 40
  CALL CHACC
  IF (ISTATE.NE.7) GO TO 40
  IF (IR.GE.IRMAX) GO TO 40
  CALL REVERT
  ISTATE = 11
  GO TO 20
40 CALL IEXIT(ISTATE)
50 IF (MORE.NE.0) GO TO 10
60 SOL=OBJ
  NFES=ISTATE

```

```

RETURN
END
SUBROUTINE QPDATA
COMMON /RETAIN/ KEEP(275)
COMMON /Q/ D(100,100),ISQ(900),HOLDQ(900),KEYTQ(180),MAXQ,NUMQ
INTEGER XBASIS,YBASIS,SIZE,SIZE1
COMMON B(275),BOUND(900),C(900),GR(275),INBASE(900),INV(165,165),
KISEFF(275),PIV(900),S(275),SLACK(275),TOL(3),X(900),XBASIS(275),X
K(275),Y(275),YAC(900),YBASIS(275),YR(275),BIG,DRIVER,INREV,IR,IRM
KX,ISBIG,ISBND,ISDONE,ISTATE,ITR,ITRMAX,M,MARKI,MARKK,MAXM,MAXN,MNI
KH,MORE,MOPEPR,MXSIZE,N,NEGINV,NEGROW,NEWX,NEWY,NUMSLK,OBJ,R,SIZE,
KIZE1,SMALL,XKPGS,YAMINC,G(275),IEND
REAL INV
DIMENSION K1(4),K2(4),F(4)
COMMON /THREE/ NSTOR1(1000,2),NSTOR2(5000,2),STOR1(1000),STOR2(500
CO),NS(4),LMAX,LQMAX
9100 FORMAT (1H0,'THE NUMBER OF VARIABLES WITH QUADRATIC COEFFICIENTS M
JUST BE BETWEEN 1 AND',I5,' YOU HAVE SPECIFIED ',I5,')
9104 FORMAT (1H0,'YOU HAVE SPECIFIED A QUADRATIC COEFFICIENT FOR A VARI
TABLE OUTSIDE THE RANGE 1 TO N, NAMELY ',I5,') IT HAS BEEN IGNOR
ED.')
9108 FORMAT (1H0,'YOU HAVE SPECIFIED THAT THERE ARE ',I2,' VARIABLES WIT
H NON-LINEAR COEFFICIENTS, BUT HAVE NOW LISTED ',I5,')
9112 FORMAT (1H0,'YOU HAVE SPECIFIED TWO DIFFERENT VALUES FOR M OR FOR
N.')
MAXQ=180
ISDONE = 0
C INSERT IN QPDATA
IF(NUMQ .GE. 1 .AND. NUMQ .LE. MAXQ) GO TO 20
WRITE (6,9100) MAXQ,NUMQ
ISDONE = 1
GO TO 200

```

```

20 DO 30 J = 1,N
30 ISQ(J) = 0
   DO 40 I = 1,NUMQ
     KEYTOD(I) = 0
     DO 40 J = 1,NUMQ
40 D(I,J) = 0.0
   NEXT = 0
   NCHK=J
   L=0
50 IF(NCHK.EQ.1) GO TO 3
   DO 1 K=1,4
     L=L+1
     IF(L.GT.LQMAX) GO TO 2
     K1(K)=NSTOR1(L,1)
     K2(K)=NSTOR1(L,2)
     F(K)=STOR1(L)
     GO TO 1
2  NCHK=1
   K1(K)=0
   K2(K)=0
   F(K)=0.
1  CONTINUE
   GO TO 5
3  K1(1)=99999
5  CONTINUE
   IF (K1(1) .EQ. 99999) GO TO 130
   DO 120 K = 1,4
     ID = K1(K)
     IF (ID .GE. 1 .AND. ID .LE. N) GO TO 60
     IF (ID .LE. 0) GO TO 50
     WRITE (6,9104) ID
   GO TO 120

```

```

60 KEYI = ISQ(ID)
   IF (KEYI .NE. 0) GO TO 80
   NEXT = NEXT + 1
   IF (NEXT .LE. NUMQ) GO TO 70
   WRITE (6,9108) NUMQ ,NEXT
   ISDONE = 1
   GO TO 200
70 ISQ(ID) = NEXT
   KEYTOO(NEXT) = ID
   KEYI = NEXT
80 JD = K2(K)
   IF (JD .GE. 1 .AND. JD .LE. N) GO TO 90
   WRITE (6,9104) JD
   GO TO 120
90 KEYJ = ISQ(JD)
   IF (KEYJ .NE. 0) GO TO 110
   NEXT = NEXT + 1
   IF (NEXT .LE. NUMQ) GO TO 100
   WRITE (6,9108) NUMQ,NEXT
   ISDONE = 1
   GO TO 200
100 ISQ(JD) = NEXT
   KEYTOO(NEXT) = JD
   KEYJ = NEXT
110 COEFF = F(K)
   D(KEYI,KEYJ) = COEFF
   D(KEYJ,KEYI) = COEFF
120 CONTINUE
   GO TO 50
130 MHOLD = M
   NHOLD = N
   CALL DATA

```

```

      IF (ISDONE .EQ. 1) GO TO 200
      IF (M .EQ. MHOLD .AND. N .EQ. NHOLD) GO TO 140
      WRITE (6,9112)
      ISDONE = 1
      GO TO 200
140  CONTINUE
      ISTART = 1
      IEND = ISTART + 11
145  IF (N .LE. IEND) IEND = N
      IF (IEND .GE. N) GO TO 150
      ISTART = IEND + 1
      IEND = IEND + 12
      GO TO 145
150  IEND = 9
      ISTART = 1
155  IF (NUMQ .LE. IEND) IEND = NUMQ
      IF (IEND .GE. NUMQ) GO TO 170
      ISTART = IEND + 1
      IEND = ISTART + 8
      GO TO 155
170  DO 175 I = 1,M
175  KEEP(I) = 0
      DO 180 J = 1,N
      IF (BOUND(J) .EQ. -1.0) BOUND(J) = BIG
180  HOLD(J) = C(J)
      IF (ISBND.EQ.J) ISBND = 1
200  RETURN
      END
      SUBROUTINE QSTART
C      INSERT FOR QP SOLUTION
      COMMON /THREE/ NSTOR1(1000,2),NSTOR2(5000,2),STOR1(1000),STOR2(500
      C),NS(*),LMAX,LQMAX

```

```

COMMON /G/ D(100,100),ISQ(900),HOLDG(900),KEYTCD(180),MAXQ,NUMQ
COMMON /FOUR/ NLEGS,J(4),P(4),C(7),MM1,I
COMMON /FIVE/ W(500),LSW(500)
INTEGER P
L=1
NCOL=C(1)+1
NROW=1
DO 104 K=1,I
  NX=P(K)
  DO 104 K1=1,NX
    K2=J(K)
    DO 104 K3=1,K2
      NSTOR1(L,1)=NCOL
      NSTOR1(L,2)=NROW
      STOR1(L)=-W(NCOL)
      L=L+1
      NROW=NROW+1
      NCOL=NCOL+1
104 CONTINUE
  LQMAX=L-1
  RETURN
END
SUBROUTINE REDUCE
  INTEGER XBASIS,YBASIS,SIZE,SIZE1
  COMMON B(275),BOUND(900),C(900),GR(275),INBASE(900),INV(165,165),
  KISEFF(275),PIV(900),S(275),SLACK(275),TOL(8),X(900),XBASIS(275),XR
  K(275),Y(275),YAC(900),YBASIS(275),YR(275),BIG,DRIVER,INREV,IR,IRMA
  KX,ISBIG,ISBND,ISDONE,ISTATE,ITR,ITRMAX,M,MARKI,MARKK,MAXM,MAXN,MNO
  KW,MORE,MOREPR,MXSIZE,N,NEGINV,NEGROW,NEWX,NEWY,NUMSLK,OBJ,R,SIZE,S
  KIZE1,SMALL,XKPOS,YAMINC,G(275),IEND
  REAL INV
  MARKI = 0

```

```

MARKK = 0
IF(NUMSLK.EQ.0) GO TO 80
IT = SIZE
DO 60 K = 1,IT
10 IF(SIZE.LE.1) GO TO 70
J = XBASIS(K)
IF(J.LE.N) GO TO 60
I = J - N
SI = S(I)
IF(SI * XR(K).LT.0.0.OR.SI.EQ.0.0.AND.XR(K).NE.0.0) GO TO 60
IF(K.EQ.SIZE) GO TO 30
DO 20 L = 1,SIZE
20 INV(K,L) = INV(SIZE,L)
J = XBASIS(SIZE)
XBASIS(K) = J
IF(J.LE.N) INBASE(J) = K
30 SLACK(I) = XR(K)
XR(K) = XR(SIZE)
IF(NEGINV.EQ.SIZE) NEGINV = K
L = ISEFF(I)
ISEFF(I) = 0
IF(L.EQ.SIZE) GO TO 50
DO 40 KK = 1,SIZE
40 INV(KK,L) = INV(KK,SIZE)
YR(L) = YR(SIZE)
YBASIS(L) = YBASIS(SIZE)
I = YBASIS(SIZE)
ISEFF(I) = L
50 XBASIS(SIZE) = 0
SIZE = SIZE - 1
SIZE1 = SIZE1 - 1
NUMSLK = NUMSLK - 1

```

```

      GO TO 10
60 CONTINUE
70 IF(SIZE.LT.2.AND.XBASIS(1).GT.N) MARKK = 1
   IF(NEGINV.EQ.0.AND.MARKK.EQ.0) GO TO 80
   J = 0
   IF(NEGINV.NE.0) J = XBASIS(NEGINV)
   IF(J.GT.N) MARKK = NEGINV
   IF(MARKK.EQ.0) GO TO 80
   MARKI = XBASIS(MARKK) - N
80 RETURN
END
SUBROUTINE REVERT
  INTEGER XBASIS,YBASIS,SIZE,SIZE1
  COMMON B(275),BOUND(900),C(900),GR(275),INBASE(900),INV(165,165),
  KISEFF(275),PIV(900),S(275),SLACK(275),TOL(3),X(900),XBASIS(275),XR
  K(275),Y(275),YAC(900),YBASIS(275),YR(275),BIG,DRIVER,INREV,IR,IRMA
  XX,ISBIG,ISBND,ISDONE,ISTATE,ITR,ITRMAX,M,MARKI,MARKK,MAXM,MAXN,MNO
  KW,MORE,MOREPR,MXSIZE,N,NEGINV,NEGROW,NEWX,NEWY,NUMSLK,OBJ,R,SIZE,S
  KIZEL,SMALL,XKPCS,YA*INC,G(275),IEND
  REAL INV
9000 FORMAT (1H,20X,'REINVERTED AT ITERATION ',I6)
  IR = IR + 1
  ITHOLD = ITR
  INREV = 1
  HOLD = SMALL
  SMALL = 0.0
  TOL3 = TOL(3)
  DO 20 K = 1,SIZE
10 IF (XBASIS(K) .LE. N) GO TO 20
   I = XBASIS(K) - N
   L = ISEFF(I)
   IF (K .EQ. L) GO TO 20

```

```

XBASIS(K) = XBASIS(L)
XBASIS(L) = I + N
J = XBASIS(K)
IF (J .GT. N) GO TO 10
20 CONTINUE
DO 40 K = 1,SIZE
  IF (XBASIS(K) .LE. N) XBASIS(K) = -XBASIS(K)
DO 30 L = 1,SIZE
30 INV(K,L) = 0.0
40 INV(K,K) = 1.0
DO 50 J = 1,N
  IF (INBASE(J) .NE. -1) INBASE(J) = 0
50 CONTINUE
DO 90 K = 1,SIZE
  NEWX = -XBASIS(K)
  IF (NEWX .LT. 0 .OR. NEWX .GT. N) GO TO 90
60 CALL NEWVEC
  NEWY = 0
  DO 80 KK = 1,SIZE
    IF (XBASIS(KK) .GT. 0) GO TO 80
    ABDIF = ABS(GR(KK))
    IF (ABDIF .LT. TOL3) GO TO 80
    IF (NEWY .NE. 0) GO TO 70
    BEST = ABS(1.0 - ABDIF)
    NEWY = KK
  GO TO 80
70 ABDIF = ABS(1.0 - ABDIF)
  IF (ABDIF .GE. BEST) GO TO 80
  BEST = ABDIF
  NEWY = KK
80 CONTINUE
  IF (NEWY .EQ. 0) GO TO 90

```

```

      IHOLD = -XBASIS(NEWY)
      IF (IHOLD .EQ. NEWX) IHOLD = 0
      CALL CHBASIS
      XBASIS(NEWY) = NEWX
      IF (IHOLD .EQ. 0) GO TO 90
      NEWX = IHOLD
      GO TO 60
90  CONTINUE
      NUMSLK = 0
      DO 110 K = 1, SIZE
        J = XBASIS(K)
        IF (J .GT. 0) GO TO 100
        I = YBASIS(K)
        XBASIS(K) = N + I
        NUMSLK = NUMSLK + 1
      GO TO 110
100 IF (J .LE. N) INBASE(J) = K
      IF (J .GT. N) NUMSLK = NUMSLK + 1
110 CONTINUE
      SMALL = HOLD
      DO 120 K = 1, SIZE
        XR(K) = 0.0
120  YR(K) = 0.0
      DO 140 K = 1, SIZE
        I = YBASIS(K)
        J = XBASIS(K)
        TC = 0.0
        IF (J .LE. N) TC = C(J)
        TB = B(I)
        DO 130 JJ = 1, N
          IF (INBASE(JJ) .NE. -1) GO TO 130
          TB = TB - BOUND(JJ) * A(I, JJ)

```

```

130 CONTINUE
DO 140 L = 1, SIZE
  XR(L) = XR(L) + TB * INV(L,K)
  YR(L) = YR(L) + TC * INV(K,L)
  IF (ABS(YR(L)).LE. SMALL) YR(L) = 0.0
  IF (ABS(XR(L)) .LE. SMALL) XR(L) = 0.0
140 CONTINUE
  NEGINV = 0
  T = 0.0
DO 150 K = 1, SIZE
  XRK = XR(K)
  J = XBASIS(K)
  IF (J .GT. N) GO TO 160
  IF (ISBND .EQ. 0) GO TO 150
  IF (BCUND(J) .EQ. -1.0) GO TO 150
  IF (XRK .GT. BCUND(J)) XRK = BCUND(J) - XRK
150 IF (XRK .GE. T) GO TO 180
  GO TO 170
160 I = J - N
  IF (S(I) .NE. 0.0 .AND. XRK * S(I) .GE. T .OR. S(I) .EQ. 0.0 .AND.
1  ABS(XRK) * (-1.0) .GE. T) GO TO 180
170 T = -1.0 * ABS(XRK)
  NEGINV = K
  DRIVER = 1.0
  IF (XRK .GT. 0.0) DRIVER = -1.0
180 CONTINUE
  IF (NUMSLK.GE.1) CALL REDUCE
  CALL CHSLCK
  CALL ISOPT
  ITR = ITHOLD
  INREV = 0
  OBJ = 0.0

```

```

      DO 190 J = 1,N
      IF (INBASE(J) .EQ. 0) GO TO 190
      OBJ = OBJ + X(J) * C(J)
190  CONTINUE
      WRITE (6,9000) ITR
      RETURN
      END
      SUBROUTINE SEEKX
      INTEGER XBASIS,YBASIS,SIZE,SIZE1
      COMMON S(275),BOUND(900),C(900),GR(275),INBASE(900),INV(165,165),
      KISEFF(275),PIV(900),S(275),SLACK(275),TOL(3),X(900),XBASIS(275),X
      K(275),Y(275),YAC(900),YBASIS(275),YR(275),BIG,DRIVER,INREV,IR,IRM
      KX,ISBIG,ISOND,ISDOME,ISTATE,ITR,ITRMAX,M,MARKI,MARKK,MAXM,MAXN,MN
      XW,MORE,MOREPR,MXSIZE,N,NEGINV,NEGROW,NEWX,NEWY,NUMSLK,CBJ,R,SIZE,
      KIZE1,SMALL,XKPOS,YAMINC,G(275),IEND
      COMMON /AREF/ AA(5000),IRCW(276),JCOL(5000),MAXA
      REAL INV
      NEWX = 0
      R = -BIG
      YACP=0.0
      PIVMAX = 0.0
      JMAXP = 0
      BESPIV = 0.0
      TOL3 = TOL(3)
      TOL4 = TOL(4)
      TOL5 = TOL(5)
      DO 10 J = 1,N
10  PIV(J) = 0.0
      DO 40 L = 1,SIZE
      I = YBASIS(L)
      SI = S(I)
      YI = YR(I) * SI

```

```

      IF (ABS(YI).LT.TOL3) YI = 0.0
      RINV = INV(NEGINV,L)
      ISTART = IROW(I)
      LAST = IROW(I+1) - 1
      DO 20 LOOK = ISTART, LAST
      J = JCCL(LOOK)
      IF (INBASE(J) .GE. 1 .OR. BOUND(J) .EQ. 0.0) GO TO 20
      AIJ = AA(LOOK)
      PIV(J) = PIV(J) + AIJ * RINV
20  CONTINUE
      IF (SI .EQ. 0.0) GO TO 40
      PIVOT = RINV * SI * DRIVER
      IF (PIVOT .GE. -TOL5 .OR. PIVOT .GE. -0.5 .AND. NEWX .NE. 0
1    .AND. YI .LT. 0.0) GO TO 40
      IF (PIVOT .GE. -0.5 .AND. YI .LT. 0.0) GO TO 30
      RATIO = YI / PIVOT
      IF (RATIO .LT. R .AND. NEWX .NE. 0) GO TO 40
      IF (RATIO .EQ. 0.0 .AND. PIVOT .GE. BESPIV) GO TO 40
      IF (RATIO .EQ. 0.0) BESPIV = PIVOT
      R = RATIO
      YAMINC = YI
      NEWX = N + I
      GO TO 40
30  IF (PIVOT .GE. PIVMAX) GO TO 40
      YACP = YI
      JMAXP = N + I
      PIVMAX = PIVOT
40  CONTINUE
      DO 60 J = 1, N
      INJ = INBASE(J)
      IF (INJ .GE. 1 .OR. BOUND(J) .EQ. 0.0) GO TO 60
      SJ = 1.0

```

```

      IF (INJ .EQ. -1) SJ = -1.0
      FUNC = YAC(J) * SJ
      IF (ABS(FUNC) .LT. TOL4) FUNC = 0.0
      PIVOT = PIV(J) * SJ * DRIVER
      IF (PIVOT .GE. -TOL5 .OR. PIVOT .GE. -0.5 .AND. NEWX .NE. 0
1      .AND. FUNC .LT. 0.0) GO TO 60
      IF (PIVOT .GE. -0.5 .AND. FUNC .LT. 0.0) GO TO 50
      RATIO = FUNC / PIVOT
      IF (RATIO .LT. R .AND. NEWX .NE. 0) GO TO 60
      IF (RATIO .EQ. 0.0 .AND. PIVOT .GE. BESPIV) GO TO 60
      IF (RATIO .EQ. 0.0) BESPIV = PIVOT
      R = RATIO
      YAMINC = FUNC
      NEWX = J
      GO TO 60
50 IF (PIVOT .GE. PIVMAX) GO TO 60
      PIVMAX = PIVOT
      YACP = FUNC
      JMAXP = J
60 CONTINUE
      IF (NEWX .NE. 0) GO TO 70
      NEWX = JMAXP
      YAMINC=YACP
      IF(NEWX.NE.0) R=YAMINC/PIVMAX
70 RETURN
      END
      SUBROUTINE SEEKY
      INTEGER XBASIS,YBASIS,SIZE,SIZE1
      COMMON B(275),BOUND(900),C(900),GR(275),INBASE(900),INV(165,165),
      KISEFF(275),PIV(900),S(275),SLACK(275),TOL(3),X(900),XBASIS(275),Xf
      K(275),Y(275),YAC(900),YBASIS(275),YR(275),BIG,DRIVER,INREV,IR,IRMA
      KX,ISBIG,ISBND,ISDONE,ISTATE,ITR,ITRMAX,M,MARKI,MARKK,MAXM,MAXN,MNC

```

```

KW, MORE, MOREPR, MXSIZE, N, NEGINV, NEGROW, NEWX, NEWY, NUMSLK, CBJ, P, S
KIZE1, SMALL, XKPOS, YAMINC, G(275), IEND
COMMON /AREF/ AA(5000), IROW(276), JCOL(5000), MAXA
REAL INV
SI = 1.0
BOUNDJ = -1.0
R = BIG
NEWY = 0
TOL5 = TOL(5)
IF(ISBND.EQ.0.OR.NEWX.GT.N) GO TO 10
IF(BOUND(NEWX).EQ.-1.0) GO TO 10
R = BOUND(NEWX)
NEWY = -1
10 IF(NEGINV.EQ.0) GO TO 30
XRNEG = XR(NEGINV)
J = XBASIS(NEGINV)
IF(J.GT.N) GO TO 20
BOUNDJ = BOUND(J)
IF(BOUNDJ.GE.XRNEG.OR.BOUNDJ.EQ.-1.0) GO TO 20
XRNEG = XRNEG - BOUNDJ
20 RTRY = XRNEG / (XKPOS * GR(NEGINV))
IF(RTRY.GT.R) GO TO 30
R = RTRY
IF(R.LE.SMALL) R = 0.0
NEWY = NEGINV
IF(R.EQ.0.0) GO TO 140
30 DO 90 K = 1, SIZE
IF(K.EQ.NEGINV) GO TO 90
GK = GR(K) * XKPOS
IF(ABS(GK).LE.TOL5) GO TO 90
J = XBASIS(K)
IF(J.GT.N) SI = S(J-N)

```

```

      IF(J.LE.N) BOUNDJ = BOUND(J)
      XX = XR(K)
      IF(GK.LE.0.0) GO TO 70
      IF(XX.LT.0.0) GO TO 90
      IF(J.LE.N.AND.BOUNDJ.EQ.-1.0.OR.J.LE.N.AND.XX.LE.BOUNDJ) GO TO 4
      IF(J.GT.N.AND.SI.EQ.1.0) GO TO 40
      GO TO 90
40  IF(XX.GE.GK * R) GO TO 90
50  R = XX / GK
60  IF(R.LE.SMALL) R = 0.0
      NEWY = K
      IF(R.EQ.0.0) GO TO 140
      GO TO 90
70  IF(J.GT.N) GO TO 80
      IF(BOUNDJ.EQ.-1.0.OR.XX.LT.0.0.OR.XX.GT.BOUNDJ) GO TO 90
      IF((XX - GK * R) .LE. BOUNDJ) GO TO 90
      R = (BOUNDJ - XX) / (-1.0 * GK)
      GO TO 60
80  IF(XX.GE.0.0.OR.S(J-N).GE.0.0) GO TO 90
      IF((XX - GK * R) .LE. 0.0) GO TO 90
      GO TO 50
90  CONTINUE
      DO 130 I = 1,MNOW
      IF(ISEFF(I).EQ.0) GO TO 100
      G(I) = 0.0
      GO TO 130
100  SLACKI = SLACK(I)
      SI = S(I)
      GI = 0.0
      ISTART = IROW(I)
      LAST = IROW(I+1) - 1
      DO 120 LOOK = ISTART, LAST

```

```

      NJ=J(K)
      DO 8 K1=1,NJ
      IF(K1.NE.1) GO TO 9
      TIMEB=X(NVAR2)
      WRITE(6,1005) K,TIMEB
1005  FORMAT(1H0,'BEGIN FIRST TRIP ON ROUTE',I2,' AT ',F9.3)
      9  NVAR3=NVAR2+1
      WRITE(6,1002) HUB,K1,X(NVAR3)
1002  FORMAT(1H0,'ARRIVE AT ',A8,' ON TRIP',I2,' AT ',F9.3)
      TIMEL=X(NVAR3)+X(NVAR3 + 1)
      WRITE (6,1007) HUB,TIMEL
1007  FORMAT(1H0,'LEAVE ',A8,' AT ',F9.3)
      NVAR2=NVAR2+2
      8  CONTINUE
      NVAR2=NVAR2+1
      1  CONTINUE
      WRITE(6,2004)
      RETURN
      END
      SUBROUTINE STFRM
      COMMON /ONE/ NSTOR3(5000,2),STOR3(5000)
      COMMON /THREE/ NSTOR1(1000,2),NSTOR2(5000,2),STOR1(1000),STOR2(5
CO),NS(4),LMAX,LQMAX
      COMMON /SEVEN/ REM(5000),ST2(5000),NST2(5000,2),NSET(500),LVAR
      N2 = NS(1) + NS(2) + NS(3)
      N1 = NS(1) + NS(2) + 1
      DO 10 K=1,LMAX
      NSTOR1(K,1)=NSTOR2(K,1)
      NSTOR3(K,2)=NSTOR2(K,2)
10  STOR3(K)=STOR2(K)
      L=0
      DO 1 K=N1,N2

```

```

      DO 1 K=1,I
      DO 2 K1=1,MM1
      NVAR=NVAR+1
      IF(NSET(NVAR).EQ.0) GO TO 2
      NCITY=K1+1
      WRITE(6,1000) K,XNAME(NCITY)
1000  FORMAT(1H0,'BEGIN ROUTE',I2,' AT ',A8)
      GO TO 3
2  CONTINUE
      NCITY=1
      WRITE(6,1000) K,XNAME(NCITY)
3  NCIT1=MM1+1
      NCIT2=NCIT1
      NHUB=0
      DO 4 K1=1,NLEGS
      NVAR1=NVAR1+1
      NCIT2=NCIT2-1
      IF(NCIT2.GE.0) GO TO 5
      NCIT1=NCIT1-1
      NCIT2=NCIT1-1
5  IF(NSET(NVAR1).EQ.0) GO TO 4
      IF(NCIT2.NE.0) GO TO 7
      NHUB=1
      WRITE(6,1001)XNAME(NCIT1),HUB,K
      GO TO 4
7  WRITE(6,1001) XNAME(NCIT1),XNAME(NCIT2),K
1001  FORMAT(1H0,'FLY FROM ',A8,' TO ',A8,' ON ROUTE',I2)
4  CONTINUE
      IF(NHUB.NE.0) GO TO 6
      NCIT1=1
      WRITE(6,1001) XNAME(NCIT1),HUB,K
6  CONTINUE

```

```

      J = JCOL(LOOK)
      INJ = INRASE(J)
      IF(INJ.LE.0) GO TO 110
      GI = GI - AA(LOOK) * GR(INJ)
      GO TO 120
110 IF(J.EQ.NEWS) GI = GI + AA(LOOK)
120 CONTINUE
      G(I) = GI
      IF(ABS(GI).LE.TOL5) GO TO 130
      IF(SI.EQ.0.0.AND.SLACKI.NE.0.0) GO TO 130
      IF(SI * SLACKI .LT. 0.0) GO TO 130
      GI = GI * XKPOS
      T = SLACKI - GI * R
      IF(T.GE.0.0 .AND. SI.EQ.1.0 .OR. T.LE.0.0 .AND.SI.EQ.-1.0)GO TO130
      R = SLACKI / GI
      IF(R.LE.SMALL) R = 0.0
      NEWY = SIZE + I
      GR(SIZEI) = GI * XKPOS
      IF(R.EQ.0.0) GO TO 140
130 CONTINUE
140 RETURN
      END
      SUBROUTINE SOP
      COMMON /THREE/ NSTOR1(1000,2),NSTOR2(5000,2),STOR1(1000),STOR2(500
      CO),NS(4),LMAX,LQMAX
      COMMON /SEVEN/ REM(5000),ST2(5000),NST2(5000,2),NSET(500),LVAR
      DO 1 K=1,LMAX
      NST2(K,1)=NSTOR2(K,1)
      NST2(K,2)=NSTOR2(K,2)
1  ST2(K)=STOR2(K)
      RETURN
      END

```

```

SUBROUTINE SSO
COMMON B(275),BOUND(900),C(900),GR(275),INBASE(900),INV(165,165),
KISFFF(275),PIV(900),S(275),SLACK(275),TOL(3),X(900),XBASIS(275),
K(275),Y(275),YAC(900),YBASIS(275),YR(275),BIG,DRIVER,INREV,IR,II,
KX,ISBIG,ISBND,ISDONE,ISTATE,ITR,ITRMAX,M,MARKI,MARKK,MAXN,MAXN,
KW,MORE,MOREPR,MXSIZE,N,NEGINV,NEGROW,NEWX,NEWY,NUMSLK,OBJ,R,SIZE,
KIZE1,SMALL,XKPOS,YAMINC,G(275),IEND
COMMON /TWO/ XNAME(20),A(50),HUB
REAL=3 XNAME,HUB
COMMON /THREE/ NSTOR1(1000,2),NSTOR2(5000,2),STOR1(1000),STOR2(5000),
NS(4),LMAX,LQMAX
COMMON /SEVEN/ REM(5000),ST2(5000),NST2(5000,2),NSET(500),LVAR
COMMON /THIR/ J1(4),C1(7),MM11,I1,NLEG
INTEGER XBASIS,YBASIS,SIZE,SIZE1
COMMON /Q/ D(100,100),ISQ(900),HULOC(900),KEYTOD(180),MAXQ,NUMQ
REAL INV
DIMENSION J(4)
I=11
NLEGS=NLEG
MM1=MM11
DO 11 K=1,I
J(K)=J1(K)
11 CONTINUE
WRITE(6,2001)
2001 FORMAT('1','THE CURRENT OPTIMAL IS')
WRITE(6,2002) OBJ
2002 FORMAT(1H0,F10.2)
NVAR=C1(1)+C1(2)+C1(3)
NVAR2=C1(1)+C1(2)+1.
NVAR1=NVAR+C1(4)
WRITE(6,2004)
2004 FORMAT('1',' ')

```

```
L=L+1
IF (STOR3(K).GE.0.) GO TO 1
STOR3(K) = -STOR3(K)
IF (NSTOR3(K,2).EQ.0) GO TO 3
IF (NSTOR3(K,2).EQ.1) GO TO 2
NSTOR3(K,2) = 1
GO TO 3
2 NSTOR3(K,2) = 2
3 N3=N2 + 1
DO 4 K1 = N3,LMAX
  IF (NSTOR3(K1,1).EQ.L) STOR3(K1) = -STOR3(K1)
4 CONTINUE
1 CONTINUE
RETURN
END
```

**The vita has been removed from
the scanned document**

AN OPTIMIZATION APPROACH TO ROUTING AND SCHEDULING COMMUTER AIRLINES

BY

Thomas A. White

(ABSTRACT)

This paper presents a method for simultaneously optimizing the routes and schedules of certain commuter airlines. This method is specifically designed for those commuter airlines which primarily provide a feeder service from small communities into a single large community.

The problem is that, given a set of small communities, a single large community, a set of aircraft and the demand as a function of time of day, determine which communities each aircraft will serve, the order in which these communities will be served and the time(s) that the aircraft will stop at each community in order to minimize passenger inconvenience. These assignments are subject to service and economic constraints.

The problem is formulated as a mixed integer quadratic programming problem. A computer program is developed to solve this problem. An example problem with an operating commuter airline is included to demonstrate the application of the solution procedure.