INFORMATION TO USERS

This material was produced from a microfilm copy of the original document. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the original submitted.

The following explanation of techniques is provided to help you understand markings or patterns which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting thru an image and duplicating adjacent pages to insure you complete continuity.

2. When an image on the film is obliterated with a large round black mark, it is an indication that the photographer suspected that the copy may have moved during exposure and thus cause a blurred image. You will find a good image of the page in the adjacent frame.

3. When a map, drawing or chart, etc., was part of the material being photographed the photographer followed a definite method in "sectioning" the material. It is customary to begin photoing at the upper left hand corner of a large sheet and to continue photoing from left to right in equal sections with a small overlap. If necessary, sectioning is continued again — beginning below the first row and continuing on until complete.

4. The majority of users indicate that the textual content is of greatest value, however, a somewhat higher quality reproduction could be made from "photographs" if essential to the understanding of the dissertation. Silver prints of "photographs" may be ordered at additional charge by writing the Order Department, giving the catalog number, title, author and specific pages you wish reproduced.

5. PLEASE NOTE: Some pages may have indistinct print. Filmed as received.

Xerox University Microfilms
300 North Zeeb Road
Ann Arbor, Michigan 48106
THE OPTIMUM MANAGEMENT OF AIR TRAFFIC
BETWEEN MAJOR AIR TERMINALS

DISSERTATION

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the Graduate
School of The Ohio State University

By
Philip Edward Taylor, B.S., M.S.

* * * * *

The Ohio State University
1975

Reading Committee:  Approved By
A. B. Bishop  Walter G. Huff
W. C. Giffin  Adviser
J. B. Neuhardt  Department of Industrial
R. W. Swain  and Systems Engineering
PREFACE

The problem of controlling the flow of air traffic takeoffs and landings to minimize delay costs was introduced to the author by Dr. Walter C. Giffin. The author is indebted to Dr. Giffin for the introduction, his continual guidance and encouragement, and his displays of confidence when most needed. This work is one of several dissertations at The Ohio State University under the direction of Dr. Giffin in the field of air traffic control. It is hoped that others will follow.

The author is also fortunate for the guidance and careful editing provided by Dr. Albert B. Bishop in the final stages of the dissertation preparation. In addition, suggestions for improvements in the text and content from Dr. J. B. Neuhardt and Dr. R. W. Swain were greatly appreciated.
ACKNOWLEDGMENTS

The encouragement and assistance of Dean George W. Hardbeck and Dr. Paul N. Loveday, both of the University of Nevada, Las Vegas, was of primary importance in completing the research. The data and information assistance provided by Mr. Roy E. Nelson of the Federal Aviation Administration provided the interface between the theory and practice of the techniques developed in the research. Mr. Nelson's cooperative efforts were invaluable.

The excellent typing assistance provided by Mrs. Jan Banker, Mrs. Grayce Woodall, Mrs. Audrey Welch and Mrs. Joyce Standish was greatly appreciated.
VITA

October 20, 1944 .......... Born - Akron, Ohio

1968 ........................ B.S.I.E., North Carolina State University, Raleigh, North Carolina

1970 ........................ M.S.I.E., North Carolina State University, Raleigh, North Carolina


1971-1973 .................... Operations Research Associate, Management Decisions Develop­ment Corporation, Cincinnati, Ohio

1973-1975 .................... Assistant Professor, University of Nevada, Las Vegas, Nevada

FIELDS OF STUDY

Major Field: Industrial and System Engineering

Studies in Inventory and Queuing Theory. Professor Walter C. Giffin

Studies in Mathematical Programming. Professor Albert B. Bishop

Studies in Experimental Design. Professor John B. Neuhardt
# TABLE OF CONTENTS

PREFACE ..................................................... ii
ACKNOWLEDGMENTS .......................................... iii
VITA .......................................................... iv
LIST OF TABLES ............................................. ix
LIST OF FIGURES .......................................... xii
LIST OF TERMS ............................................. xiii

1. INTRODUCTION ............................................. 1
   1.1 Advanced Flow Control. ............................. 4
       1.1.1 Capacity Predictions. .......................... 7
       1.1.2 Demand Predictions. ........................... 9
   1.2 Central Flow Control Procedures ................. 10
   1.3 Problems Faced Using Present Central Flow Control Procedures ................. 13

2. REVIEW OF THE LITERATURE ............................. 19
   2.1 Heuristic Approach .................................. 19
   2.2 Optimization Approaches ............................ 21
   2.3 Other Research in Air Transportation Not Directly Related to the CF^2 Problem ................. 25
   2.4 Specific Areas of the Flow Control Literature Requiring Further Research ................. 27

3. PROBLEM FORMULATION ................................... 29
   3.1 Eyster's Network Diagram ............................ 29
       3.1.1 The Three States of a Terminal-Period ....... 32
4.9.1.1 Overload reduction for flight cancelations .......... 98
4.9.1.2 Lower bound calculation for flight cancelations .... 100
4.9.1.3 Cancelation cost decrease for each flight delay ..... 100
4.9.1.4 Controlling search with cancelation permitted .... 100
4.9.2 Multiple Leg Extension ......................... 102
4.9.2.1 Computing overload reduction for multiple leg journeys ........ 104
4.9.2.2 Costing multiple leg delays to cancelations .......... 109
4.9.2.3 Controlling search for multiple leg journeys .... 110
4.9.3 Notes on Extensions ......................... 111
4.10 An Alternative Solution Procedure: Zero-One Programming ..... 111
4.11 Computing Improved Lower Bound Using Zero-One Programming .... 121
5. COMPUTATIONAL ASPECTS: COMPUTER PROGRAMS, COMPUTATION TIMES, IMPLEMENTATION AND UTILIZATION CONSIDERATIONS .......... 124
5.1 The Branch-and-Bound Computer Program ................. 124
5.1.1 Program Description ................................ 124
5.1.2 Computational Experience ......................... 127
5.1.3 An Example Computer Run ......................... 131
5.1.4 Implementation and Application Considerations .......... 142
5.1.4.1 Using man-machine interaction ..................... 143
5.2 Zero-One Computer Program .......................... 144
5.2.1 Program Description ............................ 144
5.2.2 An Example Computer Run ......................... 145
6. SUMMARY AND CONCLUSION ........................... 149
6.1 Summary of This Research .......................... 149
6.2 Recommendations for Further Research ................. 151
6.3 Conclusion ........................................ 153
7. REFERENCES .............................................. 154
8. APPENDICES .............................................. 163
  8.1 Alternative Notation. ................................. 164
  8.2 Computer Program. ............................. 165
LIST OF TABLES

1.1 An Unchanged Overload Situation Using One Possible Ground Delay ........................................ 15
1.2 An Overload Reduction of Two Operations Using One Ground Delay ............................................. 15
1.3 FAA Estimated System Improvement Costs in Thousands of Dollars for Central Flow Control Facility ........................................ 17
3.1 Four Cases Resulting at (i,t+1) From A Shift In One Unit of Demand From Period t To Period t+1 at Terminal i .................................... 36
3.2 Overload Reductions Due To A Ground Delay, As A Function of the Terminal-Period States........ 38
3.3 Overload Reductions Due To An Air Delay, As A Function of The Terminal-Period States .......... 40
3.4 State Computation Using Section 3.1 Rules ......... 43
3.5 Projected State Computation Using Section 3.1.2.3 Rules .................................................. 45
3.6 Initial Data For A Three-Flight Problem .......... 56
3.7 Modified Flight Data Corresponding To Figure 3.7 .............................................................. 57
4.1 Projected Overload Computation Table For Example Problem .................................................... 70
4.2 Air and Ground Delay Costs For Each Flight In Figure 4.1 ....................................................... 72
4.3 Overload Reductions For Ground Delays As A Function of the Projected States (O, E, or U) of the Relevant Terminal-Periods ........ 76
4.4 Overload Reductions For Air Delays As A Function of the Projected States (O, E, or U) of the Relevant Terminal-Period ........ 76
4.5 Overload Reduction and Ratio Calculations for Initial Air and Ground Delays of Each Flight in the Example Problem .................. 77

4.6 Transforming Previous Projected Terminal-Period State Incorporating a Ground Delay of Flight (1,1)+(1,2) ......................... 83

4.7 New Id, Kd, and Fd Calculations for Delayed Flight ........................................ 83

4.8 Modifications of Id and Fd Due to Condition Change at (1,3) From Underloaded to Equilibrium. . 84

4.9 Modifications of Id and Fd Due to Condition Change at (1,3) From Underloaded to Equilibrium. . 85

4.10 An Updated Composite of Tables 4.5, 4.7, 4.8 and 4.9. ................................. 86

4.11 New Projected Terminal-Period States Incorporating a Ground Delay of (1,1) (1,3). . .. 91

4.12 Overload Reduction and Ratio Calculations For Example Problem Following a Ground Delay of (1,1)+(1,3) . ................................. 92

4.13 Overload Reduction For a Cancelation Of a Flight (i,t)+(i',t+k) As a Function of the Relevant Terminal Periods. .......................... 99

4.14 General Table of Overload Changes As a Function Of the State of Affected Terminal-Periods. . . . 105

4.15 General Table of Overload Reductions For Cancelations As a Function of Affected Terminal-Periods . ............................... 107

4.16 Overload Changes For All Possible Delays and Cancelations of the Two Leg Example Shown in Figure 4.9. ................................. 108

4.17 Flight Data For Zero-One Example Problem ........................................... 115

4.18 Terminal-Period Simple Overload States For Zero-One Example Problem . .......................... 115

4.19 Tableau For Zero-One Formulation of Example Problem of Figure 4.10 and Tables 4.17 and 4.18 .............................. 118
4.20 Lower Bound Calculation Using Zero-One Programming .................................................. 123
5.1 Computer Run Times For Five Sample Problems ............................................................ 128
5.2 Eyster's Computation Times As A Function Of Three Parameters .................................. 130
5.3 Capacity, Demand, and State Data .................................................................................... 133
5.4 Flight Data ......................................................................................................................... 136
5.5 Optimal Delay Plan For Problem Defined by Tables 5.3 and 5.4 .................................... 140
5.6 Tableau For Zero-One Example Problem Of Figure 4.1 and Tables 4.1 and 4.2 .......... 146
5.7 Results of Zero-One Problem of Table 5.6 ...................................................................... 147
LIST OF FIGURES

3.1 Network Representation of the Interactions Between Terminals and Flights ................. 30
3.2 Shifts in Demands at the Origin and the Destination Terminals Due to a Ground Delay .... 37
3.3 Shifts in Demand Due to an Air Delay of a Flight ................................................. 39
3.4 Example Showing How Overloads In One Period Cause Overloads In Later Periods .......... 44
3.5 A Three-Flight Problem ................................................................. 54
3.6 Tree Diagram for a Three-Flight Problem ............ 55
3.7 A Feasible Solution to the Three-Flight Problem of Figure 3.5 ............................... 56
4.1 Requested Flight Patterns for Example Problem . . 70
4.2 Flow Diagram for Lower Bound Calculation ................. 79
4.3 Update of Figure 4.1 Showing Flight (1,1) *(1,2) Ground Delayed One Period .......... 81
4.4 Optimal Solution to the Example Problem ................ 93
4.5 Tree Considered in Arriving at Optimality for the Example Problem ........................ 93
4.6 Flow Diagram of General Algorithm to Problems in the Class Defined in Section 3.1 95
4.7 A Comparison of Tree Structure for a One-Flight Problem With and Without Cancelation 98
4.8 A Three-Flight Problem Illustrating Controlled Search With the Presence of Cancelation 101
4.9 A Multiple Leg Example ................................................................. 103
4.10 Zero-One Example Problem ............................................................... 114
LIST OF TERMS

Air Delay: The delay in a holding pattern of a flight arrival.

Backtracking: A procedure which generates nonredundant solutions, once a feasible but not necessarily optimal partial solution has been found.

Branch-and-Bound: A enumerative algorithm which utilizes calculated lower bounds to reflect dominance relationships between branches and thereby avoid total explicit enumeration.

Capacity: The number of takeoffs and landings which a terminal can accommodate during a given period.

Combination: The mixture of flight delays defined by a partial solution.

Combinatorial Programming: Refer to branch-and-bound.

Construction: The process of appending delays to a partial solution.

Demand: The number of takeoffs and landings requested at a terminal during a given period.

Feasible Solution: A solution in which all overloads have been reduced to zero. That is, demands have been adjusted using delays to meet all capacity requirements.

Flight: A set of two-tuples representing the origin terminal-period, intermediate terminal-periods (if any), and the destination terminal-period.

Flight Schedule Index: A sequence of integers arbitrarily assigned to flights as reference numbers.

Ground Delay: The delay of a flight departure.

Lower Bound: A finite number which reflects the relative value of searching a particular branch of the tree.
Multiple Delay of Same Flight: The flexibility of delaying a flight more than once or one period.

Multiple-Leg Flight: A set of three or more two-tuples (see flight).

Node in a Network: Indicates the existence of a terminal-period. A node will contain the capacity of a terminal-period.

Node in a Tree: Indicates a delay or cancelation of the flight index shown in the node.

Optimal Solution: The least costly feasible solution.

Overload Reduction: The number of overloads relieved by a particular flight delay (see projected and simple overload).

Partial Solution: Any solution which contains zero or more flight delays.

Primal Method: A procedure which successively generates better feasible solutions until the optimal solution is obtained.

Projected Overload: Overloads including simple overloads which can be determined a priori to exist due to the projected state of prior terminal-periods.

Projected State: The state of a terminal-period with respect to the state or condition of prior terminal-periods.

Requested Flight Patterns: The initial schedule which the airlines wish to fly. The initial schedule is often infeasible.

Simple Overload: The isolated condition of a terminal-period when demand exceeds capacity.

Simple State: The isolated state of a terminal-period; overloaded, equilibrium, or underloaded.

Terminal-Period: A two-tuple in which the first element indicates a terminal identifier index and the second element indicates the period identifier index.

Tree: A graphic representation of possible combinations for a combinatorial problem.
1. INTRODUCTION

The control of air traffic into and out of the major airports in the United States has become a complex and economically important decision-making task. The Federal Aviation Administration (FAA) is responsible for the safe performance of this task and, as would be expected, the airlines hold the FAA responsible for maintaining an economical flow of aircraft.

With the rapid growth in air traffic over the last 25 years has come an even more rapid growth in the number of manual operations, communications and monitoring activities required of air traffic controllers. Concurrently during this period of increasing demand, the terminals have expanded their facilities, controllers have become better trained, and equipment has been developed and installed which has improved the safe operation of the system. However, the capacity of the terminals and the controllers to handle aircraft landings and takeoffs has not increased at a sufficient rate to meet the growth in demand. The result of this unfavorable demand-to-capacity relationship is that several high demand or high density terminals occasionally find themselves in situations
where more aircraft are demanding takeoffs and landings than can possibly be accommodated. This in turn has caused excessive waiting times for both aircraft wishing to depart and, more importantly from a safety standpoint, aircraft wishing to land. With the advent of fuel shortages the already prohibitive costs of air delays have increased markedly (14).

In the short term, the terminals experiencing the excessive demand now resort to requesting that inbound aircraft due to take off from other terminals be held on the ground at their points of departure if possible until the air space surrounding the saturated terminal can be partially cleared via landings. Once the air space is cleared to a safe number of aircraft, the release of the delayed aircraft from their points of departure is granted.

The above procedures were formally instituted and defined as Advanced Flow Control Procedures (AFCP's) by the FAA in 1968. The FAA determines when AFCP's are to be utilized or implemented. The criterion for implementation of AFCP's is based on a threshold time (e.g., one hour). If the expected waiting time of one or more aircraft waiting to land at a terminal exceeds the threshold time, the procedures are implemented for that terminal to reduce the air space congestion and the air waiting time (12), (17).
Prior to the development of AFCP's the controllers utilized stop-and-go flow techniques which held airborne aircraft destined for the delayed terminal over fixes enroute. The fixes were usually outside the control zone of the busy terminal which resulted in other control zones being burdened with excessive traffic. In addition, the stop-and-go technique caused unnecessary air waiting time expenses for the aircraft held over the fixes, as well as creating unsafe conditions for aircraft flying through the control zones.

Contrasting the stop-and-go methods to AFCP's results in the observation that stop-and-go methods are short-term corrective measures at best. At worst, they can cause excess aircraft operating expenses and clutter enroute air space causing unsafe flying conditions in the cluttered zones. The AFCP's correct these shortcomings but require more advanced information. A forecast of hourly demands and a forecast of all factors influencing terminal capacity such as weather, runway conditions, and number of controllers on duty are required. Once this knowledge is accumulated a controller can project when a terminal will experience excessive demand and suggest appropriate ground delays.

At present, the methods of forecasting demands and capacities are based on rough estimates which rely more on
a controller's experience than on current data. Additional reliability and consistency in the development of forecasts are required to insure more efficient application of AFCP's (31). To correct these shortcomings, the Federal Government in 1972 requested proposals for the automation of aircraft tracking, determination of terminal capacities and demands, computer assistance in making advanced flow control decisions and several related areas (26). The FAA estimates expenditures for the total project to be $11.6 million (1973). A contract has not been awarded to date due to further planning and costs will likely exceed this figure.

1.1 Advanced Flow Control

The major concern of this research is the computer modeling and optimal solution of the advanced flow control (AFC) problem defined below:

Determine the set of least costly air and ground holds required to prevent demands for landings and takeoffs from exceeding capacities during discrete time periods (e.g., one hour time periods) at the terminals having the largest volume of traffic.

This definition encompasses the ground holding features of AFCP's and the air holding aspects of stop-and-go flow methods. The objective is to minimize total delay costs in arriving at feasible terminal loads on an hour-by-hour basis.

The methodology to be employed in this paper is that
of combinatorial programming, otherwise known as branch-and-bound procedures. This topic is considered in depth in chapters 3, 4, and 5.

An examination of the present situation and solution procedure provides further insight into the use of advanced flow control procedures (AFCP's). At present AFCP's are established for sixteen high density terminals. These sixteen terminals are:

1) Atlanta 9) John F. Kennedy (New York)
2) Boston 10) Laguardia (New York)
3) Cleveland 11) Miami
4) Dallas-Fort Worth 12) O'Hare (Chicago)
5) Denver 13) Philadelphia
6) Washington National 14) Pittsburgh
7) Detroit 15) St. Louis
8) Newark 16) Los Angeles

The density of the traffic at a terminal is based on the number of aircraft landings and takeoffs. This class of traffic covers all commercial aircraft and any private or military aircraft using instrument flight rules (IFR). Traffic using visual flight rules (VFR) can also utilize the high density terminal facilities, but VFR traffic does receive lower priority in terms of takeoff and landing service. VFR traffic departing for high density terminals are therefore risking long air waiting time or not being able to land at their desired destinations due to a heavy concentration of IFR traffic. The VFR flights in this latter case are forced to land at alternate terminal facilities. To overcome this problem, an advanced
reservation system and minimum landing fees have been instituted at some but not all high density terminals. This allows VFR and IFR traffic to reserve landing slots, if any are available, in advance of takeoff and hence, assure landing at their desired destination without extensive air waiting times.

At present, only IFR aircraft are subject to AFCP's. A large number of IFR flights take off from one high density terminal and land at other high density terminals. This emphasizes the importance of coordinating the application of AFCP's at all high density terminals. For example, if AFCP's are applied for one terminal without regard to other high density terminals, the result can be a trade-off of the relief of excessive demands at one terminal for the fostering of excessive demands at several other terminals. The simultaneous relief of all excessive demands at all high density terminals should therefore be considered by the controllers in their application of AFCP's.

The responsibility for the coordinated use of AFCP's is borne by the air traffic control system command center (ATCSCC) which is composed of the following subunits (31):

i) The Central Flow Control Facility (CF²)

ii) The Central Altitude Reservation Facility (CARF) and

iii) The Airport Reservation Office (ARO).
The Central Flow Control Facility (CF²), established in 1971 and housed in Washington, D.C., is the subunit bearing the responsibility for the timely use of AFCP's. The precise responsibilities of the CF² are described in the National Aviation System Plan (31) as follows:

This facility will exercise operational management over all airspace use, traffic distribution, flow control and rerouting on a system basis. It will monitor all key elements of the Navigational Aids (NAS) system (weather, delays, demand and capacity), and will keep the industry informed of system status.

At present, the CF² has at its disposal sixteen open lines (hot lines) of communication with each of the sixteen terminals. These lines of communication are utilized to collect information on demands and capacities for each terminal for any particular time period in the day. In addition, the CF² controllers issue ground and air delay decisions to the terminal controllers via these open lines. Note that the CF² controllers are concerned with the coordination of all high density terminal activities while the terminal controllers are concerned only with the situation at their respective terminals.

1.1.1 Capacity Predications

The CF² controllers have access to large amounts of data on which to base their air and ground delay decisions. The following information is available to CF² controllers in their efforts to arrive at hour-by-hour predictions
of each terminal's capacity for handling landings and takeoffs (26), (33), (59), (62).

i) The most important in this regard is an estimate of capacity provided by the terminal controllers. For example, a terminal controller may estimate that his or her terminal can handle 73 landings and takeoffs between 10:01 a.m. and 11:00 a.m. but can handle only 35 landings and takeoffs between 11:01 a.m. and 12:00 noon. The CF\textsuperscript{2} controllers must verify the terminal controller's estimate by examining the following data.

ii) CF\textsuperscript{2} employs meteorologists around the clock in the CF\textsuperscript{2} complex. Their sole duty is to provide current information on frontal systems and the systems' movements which may effect the hour-by-hour ability of any one of the sixteen terminals to land or depart aircraft. Snow, rain, high wind, and heavy cloud cover are several examples of weather conditions which are likely to reduce a terminal's ability to safely land or depart aircraft. (One interesting case occurs when weather conditions allow for safe departures but make landings unsafe. This case exists because departures require less runway visibility than landings.)

iii) The layout condition and length of the runways at a terminal are significant factors in determining the capacity for handling the air traffic. Some terminals have parallel runways, others have crossing runways in many different fashions, and some have both types (19), (35), (57). The condition of the pavement, the lighting, the terminal's equipment and other related functions are considered in verifying a terminal's estimate of capacity.

iv) Finally the CF\textsuperscript{2} must consider the mixture of aircraft types; that is, the variety of performance characteristics, in particular the aircraft's range of landing speed. For example, additional spacing during landing must be provided between the landing of a slow aircraft followed by a high speed aircraft. Otherwise, the high speed aircraft
will overrun its slower predecessor (59). This spacing implies less landings per hour and, therefore, a reduction in capacity (62).

1.1.2 Demand Predictions

The consumers of capacity are landings and takeoffs. It is customary to assume that a landing or takeoff consumes the same amount of terminal time. Landings or takeoffs are often referred to as operations in the remainder of this paper. The CF^2 controllers must arrive at an estimate of the demands for operations on the same hour-by-hour basis as was accomplished for terminal capacities. This prediction is determined in a manner similar to the capacity prediction, but is more difficult due to the lack of reliability of the demand data (26), (33), (59), (62).

i) The major source for demand estimation is the terminal controllers. These controllers have gained, through experience and through the recording of historical data, an intuitive "guesstimate" of the number of landings and the number of takeoffs they will handle hour-by-hour on the average day. In addition, the terminal controllers receive constant updates on flight status including cancellations and new reservations from the advanced reservation office (ARO). The CF^2 controllers verify these demand estimates by examining the following information.

ii) The CF^2 controller's receive the same information on flight status including cancellations and new reservations. The source of this data is the advanced reservation office (ARO) which handles requests for landings and takeoffs (at critical terminals) for air carriers, air taxis, and other groups. It should be noted that
unlike many other service organizations, aircraft demand cannot be disregarded by controllers. A commercial pilot cannot, in most instances, cancel a landing due to flight impatience. The private pilot on the other hand has this option and the occurrence of a cancellation is provided through the ARO. Hence, the CF2 and terminal controllers have a rough estimate of this type of demand variation.

iii) Another major source of demand data available to the CF2 is the airline schedule guide (67). This class of aircraft makes up the major portion of the total IFR traffic. However, the airline guide is in a format which is inappropriate for CF2 personnel (the guide's major function is to provide passenger information and is indexed for that purpose). Therefore, to estimate airline demands, CF2 personnel refer to recorded history and rely upon their own experience (e.g., Friday often has high demand characteristics). A significant portion of the $11.6 million FAA project will be directed toward the development of an airline schedule data bank which should provide the CF2 controllers with more accuracy in determining hourly airline demand. In addition, the aircraft tracking system (also a part of the FAA project) will certainly improve the prediction since the status and approximate location of each flight will be known at any point in time.

1.2 Central Flow Control Procedures

Based on the capacity and demand estimates previously discussed, the CF2 controllers are able to project where and when the hourly demand will significantly exceed the hourly capacity. The measure of significance is based on the controller's confidence in the accuracy of the estimates and the projected waiting time for those aircraft being held. By regulation, a projected air holding time of
one hour or more for one or more aircraft will prompt the CF controllers to begin ground delays of flights at their point of origin. The estimation of capacity and demand is usually carried out between two and three hours in advance. This planning horizon has been expanded to five hours in advance under special conditions usually concerning weather (62).

The terminals demonstrating a surplus of demands during a particular period of time can be described as being critically overloaded or overloaded. The periods of time in which a terminal is overloaded can be included in the above definition of an overloaded terminal by using the term terminal-period and, therefore, refer to an overloaded terminal-period. In all following discussions, either overloaded terminal-period or simply overload will appear interchangeably. The term overload will refer to the number by which demand exceeds capacity.

Overloads are the symptoms which the CF must first recognize, then correct by ordering ground or enroute air delays. Ground delays can reduce the number of circling aircraft at a critical terminal-period and thereby reduce both the number of overloads and the air delay time of the waiting aircraft at the overloaded terminal-periods.

At present, the CF, upon recognizing the existence of overloads at several terminal-periods, must sort through some 1,500 to 3,000 flights in order to issue ground delay
commands to terminals which have aircraft planning to depart for the overloaded terminal-periods. The issuance of one to several hundred (under severe weather conditions) commands in the form:

Hold K (number) flights departing for terminal i and arriving during period t (59), (62), may be necessary. Then the terminal controllers are required to select on a random or first-come-first-serve basis k such flights and issue the direct order to the k pilots to delay on the ground. If the time periods are measured in hours, it can happen that a delay to the next period can represent as little as a few minutes to as much as two hours. Two examples will explain this variation. Assume period 1 is 10:01 to 11:00 and period 2 is 11:01 to 12:00. Suppose a flight is to depart at 11:00 and is ground delayed to the next period. That may mean a delay to 11:01 or to 12:00 or a delay of between one minute to 60 minutes. The other case can be examined by considering a flight due to depart at 10:01. A ground delay to the next period could mean a delay of 60 minutes or a delay of 119 minutes or almost two hours.

To arrive at an equitable delay scheme for all the airlines, the controllers now randomly select the flight(s) to be delayed and assume all delays are for one period (one hour) until further notification from the CF. This assumes that in the long run all airlines have an equal
chance of being delayed, but as will be discussed later, this is not the case in a scheduled environment.

1.3 Problems Faced Using Present Central Flow Control Procedures

Due to the present limitation on the accuracy of the demand and capacity estimates, it is difficult for the CF$^2$ controllers to locate all true overload situations. In fact, only the most dramatic overload cases are predicted in advance and sometimes even they are not located. Suppose the CF$^2$ controllers predict overloads at several terminal-periods over a three-period time span. An example will illustrate the situation.

Suppose that Atlanta, Boston and New York terminals are being considered in a simplified problem situation involving the 11:00 a.m. to 2:00 p.m. time span. Further suppose the controllers have found that the 11:01 to 12 (noon) hour at Boston is overloaded by two operations. The 12:01 to 1:00 p.m. hour at Atlanta is overloaded by one operation, and the 1:01 to 2:00 p.m. hour at New York is overloaded by two operations.

The number of aircraft flying between the terminals may be in the hundreds for the three-period span (for the actual sixteen terminal problem this number reaches 3,000 to 3,500 flight operations or an average of approximately 70 per hour per terminal). The CF$^2$ controllers in this case sort through the hundreds of flights and select the
least costly combination of flight delays which will relieve all overloads. Their reasoning for selection of a flight to delay is as follows:

Search for flights scheduled to either take off or land at the overloaded terminal-periods, note the flights' points of departure, and issue commands to ground delay \( k \) flights (headed for the overloaded terminal-period) at the points of departure.

The CF\(^2\) controllers are concerned with the relief of known overloaded terminal-periods and do not consider the effect of the delays at the nonoverloaded terminal-periods. If a flight is denied takeoff in one period, then the demand for the takeoff is backlogged (similar to order backlogging in a production-inventory environment) to the next period. This backlogging effect can result in the creation of overloads at terminal-periods not previously overloaded. Due to the magnitude of the problem and the unreliability of the data the controllers are humanly unable to examine the effects of backlogging. Therefore, the controllers are often trading one overloaded situation for others.

In the example being considered, the ground delay of a flight leaving Atlanta during the 11:01 a.m. to 12 noon hour and scheduled to arrive in New York during the 1:01 to 2:00 p.m. hour results in such a trade-off. Recall that New York is overloaded by two operations during the 1:01 to 2:00 p.m. hour, while Atlanta is not overloaded between 11:01 to 12 noon hour. Using the controllers'
selection rule would result in delaying this flight since it reduces an overload in New York. However, the delay of a departure from Atlanta in the 12:01 to 1:00 p.m. hour

**TABLE 1.1**

<table>
<thead>
<tr>
<th>Overloads Before Delay</th>
<th>Overloads After Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atlanta</td>
<td>1</td>
</tr>
<tr>
<td>Boston</td>
<td>2</td>
</tr>
<tr>
<td>New York</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>5</td>
</tr>
</tbody>
</table>

creates an additional overload in Atlanta. The net change in overloads is zero, but the cost of a ground delay has been incurred (see table 1.1).

On the other hand, ground delay of a flight taking off from Boston during the 11:01 to 12 (noon) hour enroute to Atlanta arriving between 12:01 and 1:00 p.m. results in the reduction of two overloads as shown in table 1.2.

**TABLE 1.2**

<table>
<thead>
<tr>
<th>Overloads Before Delay</th>
<th>Overloads After Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atlanta</td>
<td>1</td>
</tr>
<tr>
<td>Boston</td>
<td>2</td>
</tr>
<tr>
<td>New York</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>5</td>
</tr>
</tbody>
</table>
The controller's selection rule would show no preference for the second delay over the first. The question is then, why is this selection rule used? The answer appears to be that:

1) The prediction of overloads is not accurate.
2) The effect of backlogging a vast number of possible flights is humanly impossible to calculate in the short time required to make final decisions.
3) The human mind is unable to consider the effects of combinations of delays superimposed on all terminal-periods in a short time span.

To improve the delay-selection rule requires correction or improvement in these three areas. The first area, overload prediction, can be improved by implementation of the flight tracking, demand prediction, and capacity prediction features of the proposed FAA project. The FAA ten-year systems improvement plan is described in table 1.3 and the following excerpt from The 1973 National Aviation System Plan (31):

An interim automation capability for central flow control is being provided to the Central Flow Control Facility by use of input-output terminals connected to a remote time-shared computer service. The prototype flow control computer programs are being evaluated and modified with operator participation. Programs which prove to be effective in the operational context will be retained and will form the basis for the next levels of software development. Functions currently planned for the prototype program include terminal flow control, en route flow control and automation of the airport
reservation office functions.

Level I capability will also be based on time-shared data processing, but will begin to handle discrete flight plan information, as a supplement to the stored data on scheduled traffic. The capacities of ARTCC's, major jet routes and major terminals will also be available to enable the system to predict overloads. Direct interfaces with ARTCC computers will be developed to permit transmission of both ATC system status and progress of selected flights to the Central Flow Control Facility. Other system command center functions (e.g., central altitude reservations and contingency command post operations) will be assessed and automated if required.

When the dedicated computer (or alternative capability) has been selected, the Level II software will be implemented for operational use. This will include the capability to match projected demand to capacity, calculate expected delays and test alternative flow control strategies. The complete Level II program implemented on the dedicated machine will comprise the operational flow control automation system.

### TABLE 1.3

FAA ESTIMATED SYSTEM IMPROVEMENT COSTS
IN THOUSANDS OF DOLLARS FOR CENTRAL FLOW CONTROL FACILITY

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$1,000</td>
<td>$2,000</td>
<td>$500</td>
<td>$500</td>
<td>$500</td>
<td>$2,500</td>
<td>$7,000</td>
</tr>
</tbody>
</table>

The Level I systems are concerned with capacity, demand and overload prediction. The Level II systems are to include algorithms to resolve overloads. But overload resolution requires consideration of the second area, backlogging effect, which relies first on accurate overload prediction and secondly on a computationally efficient means of predicting the backlogging effect for hundreds
or thousands of flight delays. The computer is well suited for this type of calculation.

The third area involving the total effect of from one delay to several hundred delays is a complex optimization problem and requires an algorithmic solution to yield computationally feasible results. Computational feasibility (of an algorithm) in this problem environment can be defined as the ability to yield decisions in a period of one to ten minutes as the CF\(^2\) controllers are now forced to do.

This research is most concerned with the latter two areas. An algorithm is presented in chapter 4 which addresses these two areas. The computational efficiency of the algorithm is measured in chapter 5.
2. REVIEW OF THE LITERATURE

This chapter reviews the methods and techniques developed to solve the central flow control problem and surveys other systems research in the air transportation area. The central flow control literature can be divided into two basic approaches; optimization and heuristic or data processing procedures.

Before the literature is discussed, the general definition which was stated in chapter 1 is revised and stated here for convenience:

Given a set of terminal-period capacities and the demands for landings and departures at each terminal-period determine first if overloads exist. If they exist then determine the least costly set of delays which results in the reduction of all overloads (i.e., zero overloads) of all terminal-periods.

The terminology used in this definition was defined in the Introduction. This problem statement is referred to in all future discussions as the central flow control problem or simply the problem.

2.1 Heuristic Approach

Of the two approaches suggested in the literature, the heuristic is the more computationally feasible. Medeiros and Sussman 1973 (31) of the Department of
Transportation have developed one possible heuristic procedure as part of the Airport Information Retrieval System (AIRS). A basic requirement of the heuristic method is that each terminal-period has a predicted quota of landing slots for landing either previously held aircraft or arriving aircraft. The procedure then is to apply the landing slots to the arriving and holding flights on a first-come-first-serve basis. During this procedure of simulating landings, each flight's predicted air waiting time is computed. This waiting time is then compared to a user predefined threshold time and, if the waiting time exceeds the threshold time, the procedure computes ground delay allocations. These allocations are in terms of the number of aircraft which must be ground delayed at their point of departure to avoid excessive air waiting times.

Two fundamental points should be noted here. First, the procedure is concerned with the quota for landings only and avoids the departure quota by assuming the departure quota is fixed when developing the landing slot quota prediction.

In addition, when determining the ground delay allocations no effort is made to determine the least costly set of delays. Instead, delays are to be issued according to a "first-come-first-serve" policy without regard to the effect these delays may have on the other terminal-periods.
This procedure then considers on a one-at-a-time basis each terminal-period's landing overloads and attempts to solve this subportion of the problem without regard to all other terminal-periods' landing or departure overloads and without regard to its own departure overloads. Hence, the methodology does not minimize delay costs but does yield answers quickly due to the "data processing nature" of the approach. This illustrates the fact that many feasible delay plans exist and delay costs need not be considered to arrive at feasibility.

The general nature of the above approach is quite similar to the methods employed by the CF² personnel as discussed in the Introduction. This appears to be the motivation behind the AIRS development.

The AIRS procedures are now being evaluated in the Central Flow Control Facility. At this time no other procedures have been tested (31). Improvements in the above heuristics will likely result from this testing.

2.2 Optimization Approaches

In an entirely different vein, Eyster (13) in 1972 proposed an optimization scheme which attempts to consider both the interrelationship between terminals and the actual delay costs as integral parts of the problem. His justifications include the fact that air carrier delay costs amount to $118 million per year in 1968 (54). This
report (54) has not been issued since 1968. However, data from (24) show a growth of 27 percent in flight operations over the five-year span from 1968 to 1973. With this large increase in number of flight operations and with the advent of the fuel cost increases of 1973 and 1974, it is expected that the $118 million is a conservative figure for 1974. This large cost emphasizes the importance of cost effective delay selection.

Eyster's insights into the problem of advanced flow control gave rise to this research. His most significant contribution is in the area of problem formulation. Eyster directed himself toward the interconnection of all terminal-periods so that a delay in one terminal-period can be reflected in the demand-capacity relationship at other relevant terminal-periods. This allowed Eyster to select, using a set of linear equilibrium equations and linear programming, the optimal set of air and ground delays. As Eyster points out, the number of equations and variables is extremely large and for several small problems of five terminals and twelve time periods the storage requirements become so large that the storage limits of today's computer are exhausted. Eyster recognized these facts and suggested the use of network theory which could take advantage of the special structure the linear programming matrix exhibited and would also yield integer solutions.

Eyster's network problem formulation of the problem
is used extensively in the following chapter on problem formalization and therefore will not be presented here.

Another optimization approach was proposed by Ellis and Rishel (12). They attacked a much narrower problem than Eyster in terms of numbers of terminals. Their approach considers a two-terminal situation with flow from terminal 1 to terminal 2 only and where capacities for landing at terminal 2 and takeoffs at terminal 1 are considered to be random variables. This is unlike Eyster's approach where the capacities were assumed to be known with certainty and flows were allowed to occur in either direction. Eyster proposed that his algorithm be rerun periodically to update the delay plan (i.e., solution) according to updates in capacity and demand information. On the other hand, Ellis and Rishel attempt to develop policies rather than specific plans for each terminal-period and, therefore, would not require the quantity of reruns, but this is not adequate for tactical decision-making.

As previously noted, Ellis and Rishel consider the one-way flow between two terminals. They utilize dynamic programming to develop the optimal policies for the terminal-periods where these policies are functions of several parameters. For example, the optimal landing policy for terminal 2 during period i is a function of the number waiting to land at terminal 2 in period i,
the number waiting to depart from terminal 1 in period i, the number which departed from terminal 1 in the previous periods, and the random variable representing the maximum number which can land at terminal 2 during period i. Once the parameters are known then the optimal plan is known via the functional policy relationship developed by the dynamic programming algorithm.

Two notes on the Ellis and Rishel approach should be made. Their approach does not consider individual flight delay costs and their assumptions that the landing and takeoff capacities for the two terminals are independent random variables is not founded on practical grounds. For example, a frontal system can affect wide areas and can simultaneously alter capacities at more than one terminal. Finally, as Ellis and Rishel point out, the extension of this approach to more than two terminals or to a two-way flow results in a high dimensional nonlinear programming problem and the dynamic programming algorithm is inappropriate due to loss of the decomposition structure. At the present state of development in nonlinear programming, practical solution methods for high dimensional nonlinear programming problems are unavailable. This is particularly true if practical solution times must be in the range of one to ten minutes as required by the CF$^2$ environment.

Prior to these three approaches, little effort in
finding mathematical or data processing solution pro-
cedures to solve the advanced flow control problem
appeared in the literature. Eyster (13) provides a brief
sketch of what was accomplished prior to 1972.

2.3 Other Research in Air Transportation
Not Directly Related to the CF\textsuperscript{2} Problem

Several interesting and important research projects
related to air transportation have been carried out since
1972. Although they are not directly related to the CF\textsuperscript{2}
problem, it is useful to examine these research projects
in terms of problem definition and solution procedure.

A first-prize winner in the 1973 Operations Research
Society Doctoral Dissertation Contest is the research
performed by Richardson (83) which deals with development
of schedules from the viewpoint of an individual airline.
The problem is one of routing in a long haul low density
market via intermediate stops. The objective is to
minimize operating costs in the construction of routes for
a single aircraft from an origin to a destination.
Extension to several aircraft is given.

Another extension considered by Richardson is that
of passenger inconvenience. A constraint set is defined
which insures that a passenger is flown from his origin to
his destination without changing aircraft if such a routing
exists. If the routing does not exist, that passenger's
demand is not satisfied and a cost is incurred.
The methodology employed is a combination of branch-and-bound and the simplex method. The simplex method considered the passenger distribution aspects while the branch-and-bound procedure solved an integer network model.

Pollack (77) and Lee (55) consider the same problem Richardson (83) approached but did not consider passenger inconvenience. Lee's approach was theoretical, while Pollack's approach utilizes heuristics to obtain computational and dimensional feasibility for large problems.

Other recent (1972 to 1974) air transportation research not directly related to prescriptive scheduling includes Ferrar's (35) work on capacity allocation with emphasis on environmental quality, Hockaday's and Kanafani's (43) research in airport capacity analysis, and de Neufville's and Mira's (10) examination of optimal pricing policies for air transport networks. In addition, Ratchford (78) developed a model for estimating general aviation demand. Nodi (61) analyzed the air traffic control sectors using a nested queueing model, Daellenbach (7) used dynamic programming to determine optimal runway exit locations, and Schmidt (84) considered optimal control of aircraft for terminal area approach.

This recent thrust in research in the air transportation field is due in part to the emphasis the Department of Transportation has placed on system automation and effective system's management as described in The
National Aviation System Plan - Ten Year Plan 1973-1982

(31) and The National Aviation Systems Plan - Policy Summary (32). The Policy Summary (32) emphasized particularly research in the following areas:

i) Automation of Air Traffic Control to allow the controllers to be observers/monitors.

ii) Relief of disruptions due to the peak demand problem by smoothing demand (the topic of this research) and increasing capacity.

iii) The impact of environmental concerns on aviation system.

iv) Equitable accommodation of all user's including both civil and military requirements.

2.4 Specific Areas of the Flow Control Literature Requiring Further Research

Several areas of the CF\(^2\) literature require further research, but of particular importance is the development of a practical solution procedure. To date a computationally practical solution procedure for a practical problem involving many terminals and time periods has not been developed. A practical problem size is forecasted to involve approximately 20 terminals, a five-hour time horizon, and approximately 3,500 flights. This forecast is based on projections of the present problem size using the present growth rate over a ten-year period. The forecast is further verified through discussions with experienced CF\(^2\) personnel, and discussions with other experienced FAA (62), (66) personnel. A practical solution
procedure must be able to solve a problem of such magnitude in seconds.

The development of a computationally efficient procedure is, therefore, the topic of most concern in chapter 4 of this paper. Of secondary interest in chapter 4 is the incorporation of several extensions into the solution procedure which insure more realistic answers. It is premature to discuss these extensions at this point; however, chapter 3 develops them in detail.

In addition, chapter 3 provides a detailed problem formulation utilizing Eyster's network definition as a starting framework. The problem is then formulated as a decision tree in preparation for discussion of the combinatorial programming (or branch-and-bound) solution procedure proposed in chapter 4.
3. PROBLEM FORMULATION

The purpose of this chapter is to define and discuss the problem in more formal terms and to demonstrate how the problem can be structured in a combinatorial or tree diagram framework.

3.1 Eyster's Network Diagram

As indicated in the previous chapter, Eyster has developed a convenient way of defining the problem. He utilizes a network diagram structure where each node represents a particular terminal during a particular time period (see figure 3.1).

A period can be any unit of time, for example twenty minutes, while the terminals may be Atlanta, Baltimore and New York. It is necessary to define the meaning of each symbol in figure 3.1. A node is utilized to represent one unit of time at one specific terminal, while the number inside the node represents the estimated capacity of the terminal (and the terminal controllers) to handle both landings and takeoffs from the terminal during the unit of time. To illustrate (see figure 3.1) the node for terminal 1, period 1 in the upper left hand corner of figure 3.1 might represent the 9:00 o'clock to 9:20
o'clock period in Atlanta. The number four (4) inside that node indicates that the Atlanta terminal during that twenty-minute interval can handle a maximum of four takeoffs and/or landings. The underlying assumption here is that a takeoff or a landing consumes the same amount of controller and terminal time and attention.

The arcs connecting the nodes represent individual flights where the beginning of the arc represents a takeoff.
from one terminal-period and the arrowhead portion of the arc represents a landing at another terminal-period. To define a particular flight, it is only necessary to specify the origin terminal and period and the destination terminal and period. For example (see figure 3.1), the arc originating at terminal 1, period 2 and terminating at terminal 2, period 3 represents one specific flight.

To summarize, figure 3.1 is a representation of a three-terminal and four-time period problem where each flight is defined by specifying the origin terminal, the origin time period, the destination terminal, and destination time period. If the terminals are defined by the index i where i = 1, 2, or 3 and the time periods are similarly defined by t where t = 1, 2, 3, or 4 then a terminal-period is defined by the index pair (i,t). The capacity of a terminal-period can now be defined as 

\[ C_{i,t} \]

or more simply as \( C_{i,t} \). In addition a particular flight can be defined by specifying two index pairs such as (1,2) and (2,3). More generally, a flight can be represented by \((i,t) \rightarrow (i',t+k)\) where i represents the origin terminal, t the origin time period, i' represents the destination terminal, and t+k the destination time period. Here k is the number of time periods required to make the flight. Finally, the arrow (\(\rightarrow\)) is used to emphasize the direction of the flight from the origin to the destination. To confirm this flight notation requires
an example: consider (1,1) → (2,3) which represents a flight departing from terminal 1 (e.g., Atlanta) during time period 1 (e.g., the 9:00 to 9:20 o'clock period) and landing at terminal 2 (e.g., Baltimore) during time period 3 (e.g., the 9:40 to 10:00 o'clock) taking a total of two time periods in flight (i.e., from 40 to 60 minutes in flight). The actual flight may have departed at 9:05 and landed at 9:57 or 52 minutes flight time, but since the major concern is the demand during a particular time period the accuracy of the flight time is not of significance. This variation in flight time is due to the discrete time scale being used. The accuracy of the flight time is determined entirely by the size of the unit of time. If the unit of time were say one minute, the accuracy of the flight time would be within one minute.

3.1.1 The Three States of a Terminal-Period

Recall that advance flow control is concerned with the elimination of excessive air waiting times due to demand for landings and takeoffs exceeding terminal and controller capacity. Demand for landings and takeoffs for a particular terminal-period can be determined easily using a network diagram like figure 3.1. The demand for landings is determined by simply counting the number of arcs pointing into the terminal-period node. Similarly, the number of takeoffs are determined by counting the
number of arcs emanating out of the terminal-period node. The total demand for both landings and takeoffs at a particular terminal-period \((i, t)\) is then simply the sum of the two counts and is defined here as \(D_{i, t}\).

The estimated capacity of a terminal-period \(C_{i, t}\) is assumed to be a given quantity determined by the controllers. If the demand exceeds the capacity for a particular terminal-period \((i, t)\) then an overloaded situation is said to exist. That is,

\[
\text{If } D_{i, t} > C_{i, t}, \text{ then } O_{i, t} = D_{i, t} - C_{i, t} \quad (3.1)
\]

where \(O_{i, t}\) is the number of excess operations requested above the maximum which can be performed, \(C_{i, t}\).

If the demand \(D_{i, t}\) is strictly less than the capacity \(C_{i, t}\) the terminal-period is considered to be underloaded. In this case the terminal and controllers are capable of handling additional operations (i.e., landings and takeoffs) during the time period. Mathematically,

\[
\text{If } D_{i, t} < C_{i, t}, \text{ then } U_{i, t} = C_{i, t} - D_{i, t} \quad (3.2)
\]

where \(U_{i, t}\) is the number of extra operations which could be performed at terminal \(i\) during time period \(t\).

The last case to consider is when demand precisely equals capacity and will be referred to henceforth as the equilibrium condition. More specifically,

\[
\text{Equilibrium exists if } D_{i, t} = C_{i, t}. \quad (3.3)
\]
It is now possible to label and quantify the condition of each terminal-period as underloaded, overloaded or in equilibrium, and in the case of overloading or underloading the extent or magnitude of each is defined by $O_{i,t}$ or $U_{i,t}$, respectively. In figure 3.1 (1,1) is in equilibrium, (1,2) is overloaded by two (i.e., $O_{1,2} = 2$), (1,3) is overloaded by one (i.e., $O_{1,3} = 1$), and (1,4) is underloaded by three (i.e., $U_{1,4} = 3$).

Recognizing that in the cases of underloading or equilibrium the demand can be handled without excessive waiting eliminates much of the concern with these two states. The overloading case is, however, of major concern. How to reduce the demand to the maximum capacity level and thus convert overloaded terminal-periods to either equilibrium or underloaded terminal-periods is the topic of the next section.

3.1.2 Reduction of Overloads Using Flight Delays

To reduce an overloaded terminal-period to either equilibrium or to an underloaded condition requires that the demand be reduced at the overloaded terminal-period. Recall that capacity is already estimated at its maximum safe level and cannot therefore be increased to accommodate any additional operations. Thus, demand is the only controllable variable.
3.1.2.1 Ground delays to reduce demand and overloads

To reduce demand means to reduce either the number of landings or takeoffs or a combination of both. The reduction of the number of takeoffs requires that flights be delayed on the ground until the next time period. But this causes increased demand for takeoffs in the next period.

To illustrate suppose \( O_{i,t} = 1 \), indicating one overload at terminal \( i \) during period \( t \), then a ground delay of a flight starting at \( (i,t) \) will result in a reduction in demand at \( (i,t) \) by one unit to \( D_{i,t} - 1 \). At the same time, however, the demand at the same terminal \( i \) during time period \( t+1 \) is increased by one unit to \( D_{i,t+1} + 1 \). Thus the ground delay of the flight at terminal \( i \) for one period has simply shifted one unit of demand away from period \( t \) and into period \( t+1 \). The result at \( (i,t) \) is the relief of all overloads (i.e., equilibrium), but the result at \( (i,t+1) \) is dependent on the state \( (i,t+1) \) was in prior to the shift.

Table 3.1 shows the four possible cases which can occur at \( (i,t+1) \) before and after a shift of demand. Cases i) and ii) resulted in increases in overloads, an unfortunate but sometimes necessary condition as will be shown in section 3.1.3, Projections of Overloads to Future Periods. In the two remaining cases the underloading at \( (i,t+1) \) absorbed the demand shift, resulting in the favorable conditions of equilibrium and underloading. Regardless
TABLE 3.1

FOUR CASES RESULTING AT (i, t+1) FROM A SHIFT IN ONE UNIT OF DEMAND FROM PERIOD t TO PERIOD t+1 AT TERMINAL i

<table>
<thead>
<tr>
<th>Before Shift</th>
<th>After Shift</th>
</tr>
</thead>
<tbody>
<tr>
<td>i) $o_{i,t+1} \geq 1$</td>
<td>$o_{i,t+1} \geq 2$</td>
</tr>
<tr>
<td>ii) EQUILIBRIUM</td>
<td>$o_{i,t+1} = 1$</td>
</tr>
<tr>
<td>iii) $U_{i,t+1} = 1$</td>
<td>EQUILIBRIUM</td>
</tr>
<tr>
<td>iv) $U_{i,t+1} \geq 2$</td>
<td>$U_{i,t+1} \geq 1$</td>
</tr>
</tbody>
</table>

of the state of (i, t+1), if (i, t) is overloaded, it is absolutely necessary to make a delay which relieves the overload at (i, t). The effects on (i, t+1) are discussed in detail in section 3.1.3.

Another aspect of the ground delay of a flight is the ramification at the destination terminal. Recall that a flight is defined by two index pairs (i, t) + (i', t+k). At this point only the effects at the origin terminal-period (i, t) have been considered. But the destination terminal-period (i', t+k) must also be considered on an equal basis. Observing that a ground delay of one period forces both the departure time and the arrival time to be altered by one period shows the importance in considering the shift of demand at the destination terminal i' as well as the origin terminal i (see figure 3.2).

One of the three possible states of overloading,
underloading or equilibrium can exist at terminal $i'$ during period $t+k$. If $(i',t+k)$ is overloaded the delay has the favorable result of reducing the magnitude of the overloads at $(i',t+k)$. If either underloading or equilibrium exists at $(i',t+k)$ then no reduction in overloads occurs. Section 3.1.3 considers the effects at $(i,t+1)$ and $(i',t+k+1)$.

To summarize the above ground delay discussion, refer to table 3.2 where the four important combinations of states at terminal-periods $(i,t)$ and $(i',t+k)$ are examined in terms of reduction of overloads. Notice that the maximum possible reduction of overloads using a ground delay is two and occurs when both $(i,t)$ and $(i',t+k)$ are overloaded. If overloading occurs at one and only one of the two terminal-periods (cases ii and iii) then the
**Table 3.2**

**Overload Reductions Due to a Ground Delay, as a Function of the Terminal-Period States**

<table>
<thead>
<tr>
<th>Case</th>
<th>Condition at Origin $i,t$</th>
<th>Condition at Destination $i',t+k$</th>
<th>Overload Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>i)</td>
<td>OVERLOADED</td>
<td>OVERLOADED</td>
<td>2</td>
</tr>
<tr>
<td>ii)</td>
<td>OVERLOADED</td>
<td>UNDERLOADED OR EQUILIBRIUM</td>
<td>1</td>
</tr>
<tr>
<td>iii)</td>
<td>UNDERLOADED OR EQUILIBRIUM</td>
<td>OVERLOADED</td>
<td>1</td>
</tr>
<tr>
<td>iv)</td>
<td>UNDERLOADED OR EQUILIBRIUM</td>
<td>UNDERLOADED OR EQUILIBRIUM</td>
<td>0</td>
</tr>
</tbody>
</table>

Maximum overload reduction is one unit. If no overloads occur at either of the two terminal-periods, then a ground delay produces no overload reduction and can in fact cause increases in overloading as will be shown later in this chapter.

**3.1.2.2 Air Delays to Reduce Demand and Overloads**

Recall that demand at a particular terminal-period can be reduced by either reducing the number of landings or the number of takeoffs. A ground delay of a flight was found to be potentially effective in reducing the number of takeoffs at the flight's origin terminal and landings at the flight's destination terminal. It is now appropriate to examine the effects of an air delay of a flight.
Unlike a ground delay, an air delay cannot reduce the number of takeoffs at its origin terminal. This is due to the obvious fact that a flight must have taken off from the origin in order to be air delayed. However, an air delay of a flight enroute to the destination terminal can reduce the number of landings at the destination terminal-period. Figure 3.3 illustrates the demand shift at the destination terminal due to the air delay (compare to figure 3.2) of flight \((i,t) \rightarrow (i',t+k)\). Figure 3.3 shows that an air delay has no effect on the demand at the origin terminal-period. Therefore, no reduction in overloads at the origin terminal-period can occur due to an air delay.

![Diagram showing demand shifts due to an air delay](image)

**Fig. 3.3.** Shifts in demands due to an air delay of a flight.
To summarize the effects of an air delay on overloads requires only the examination of the condition of the destination terminal-period. Table 3.3 similar to table 3.2 shows the two possible cases and the overload reduction related to each. Notice that the maximum possible overload reduction using an air delay is one (as compared to two for a ground delay) and occurs when an overload exists at the destination terminal-period. If either the equilibrium or the underloaded condition exists then no overload reduction is possible.

**TABLE 3.3**

OVERLOAD REDUCTIONS DUE TO AN AIR DELAY, AS A FUNCTION OF THE TERMINAL-PERIOD STATES

<table>
<thead>
<tr>
<th>Case</th>
<th>Condition at Destination (i, t+k)</th>
<th>Overload Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>i)</td>
<td>OVERLOADED</td>
<td>1</td>
</tr>
<tr>
<td>ii)</td>
<td>EQUILIBRIUM OR UNDERLOADED</td>
<td>0</td>
</tr>
</tbody>
</table>

3.1.2.3 Summary of delay effects on overloads

From the discussion presented above it was found that a ground delay of the appropriate flight can result in a maximum of two overload reductions, one for the overloaded origin and one for the overloaded destination. The three possible effects of a ground delay on overload
reductions were found to be dependent on the states of the origin and destination terminal-periods and ranged from zero to two (0, 1, and 2) overload reductions.

An air delay on the other hand was found to result in either a zero or one (0 or 1) overload reduction. This was observed by noting that an air delay can only reduce the number of landings at the destination terminal-period and has no effect on the demand at the origin terminal-period.

The above discussion ignored the problem of increasing overloads with an air or ground delay. An increase in overloads can happen under several conditions one of which is when (i,t) and (i',t+k) are not overloaded but (i,t+1) and (i',t+k+1) are overloaded or in equilibrium. Of course, if the delay of a flight creates a more overloaded situation than existed before the delay, then the flight would not be a desirable delay.

The next section is critical to an understanding of the algorithm in chapter 4.

3.1.3 Projection of Overloads to Future Periods

This section attempts to arrive at a better estimate of the number of delays required to reduce all overloads to zero. It is somewhat more abstract than the previous discussion of overloads. A good understanding of each concept in this section is crucial because it is
relied upon in chapter 4 (refer to appendix 8.1).

Up to this point the state of a terminal-period has been determined by comparing the demand at the terminal-period to the capacity at the terminal-period and has neglected one important factor. That factor is that demands for takeoffs and landings at a particular terminal-period will change when overloads are reduced by flight delays. Fortunately, a method for determining these changes a priori is possible as discussed below.

It has been shown that demands can be passed from period \( t \) to period \( t+1 \) for terminal \( i \) by using flight delays. Likewise, by using other flight delays period \( t+1 \) demands can be passed to period \( t+2 \), and \( t+2 \) demands can be passed to \( t+3 \), etc. Suppose two overloads exist at terminal \( i \) during period \( t \) (\( O_{i,t} = 2 \)), then two demands could be passed to period \( t+1 \) to reduce overloads for period \( t \) to zero, but this increases the demands in \( t+1 \) by two \( (D_{i,t+1} + 2) \) thus altering the original demand for landings and takeoffs at \( (i,t+1) \). Therefore, the computation of demands put forth in section 3.1 must be modified to include these additional demands due to overload corrections. An example using terminal one of figure 3.1 will illustrate the previous discussion.

Following the logic of section 3.1 the terminal 1 capacities, demands, and states by period are as follows:
TABLE 3.4
STATE COMPUTATION USING SECTION 3.1 RULES

<table>
<thead>
<tr>
<th>Period</th>
<th>Capacity</th>
<th>Demand</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>4</td>
<td>$E_{1,1}$</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>$O_{1,2} = 2$</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>$O_{1,3} = 1$</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>2</td>
<td>$U_{1,4} = 3$</td>
</tr>
</tbody>
</table>

In table 3.4, $O_{i,t} = k$ means $k$ overloads at terminal $i$ during period $t$, $E_{i,t}$ implies equilibrium at $i,t$ and $U_{i,t} = L$ implies underloading at $i,t$ of $L$ units.

Note in table 3.4 that overloads exist in periods 2 and 3. Using the procedure put forth in this section requires that demand in period 2 must be reduced by two units using flight delays regardless of what happens in period 3. This means demand in period 3 is increased by two units resulting in a total demand of five units rather than three. The five units of demand in period 3 modifies the overloads in period 3 from one ($O_{1,3} = 3 - 2$) to three units ($O'_{1,3} = 5 - 2$). Again applying the procedure put forth in this section results in period 4 being in equilibrium ($E_{1,4} = 0$) rather than underloaded ($U_{1,4} = 3$).

Referring back to table 3.4 it would seem appropriate to say that at least three flights taking off or landing at terminal 1 must be delayed due to the two overloads in
period 2 and the one overload in period 3. However, considering the modifications in the demand and overload calculations just discussed, it is necessary to conclude that at least five flights taking off or landing at terminal 1 must be delayed. Then, instead of three overloads existing, five exist. Figure 3.4 illustrates this reasoning, using arcs to indicate necessary shifts in demand (i.e., delays).

\[ F_1, 1 = 0 \]
\[ O_1,1 = 2 \]
\[ O_1, 3 = 1 \]
\[ U_1, 4 = 3 \]

Fig. 3.4. Example showing how overloads in one period cause overloads in later periods.

In figure 3.4 at least two flights must be delayed (either on the ground or in the air) in period 2. This results in an increase in demand in period 3 due to the two additional demand units transferred from period 2. Since period 3 is already overloaded by one unit, these two additional demands cause two additional overloads in period 3. This is unfortunate but necessary. Therefore, at least three delays must be executed in period 3 for a total of five delays thus far (i.e., two in period 2 and three in period 3). The projected state of period 2 is two overloads and the projected state of period 3 is three
overloads, not one overload. The last period is underloaded by three units \( U_{1,4} = 3 \) and since only three flights must be delayed from period 3 \( (O_{1,3} = 1 + 2 = 3) \) the projected state of period 4 is equilibrium \( (E_{1,4} = U_{1,4} - O_{1,3} = 3 - 3 = 0) \). Table 3.5 shows the simple states of table 3.4 and the computed projected states which are of most concern.

**TABLE 3.5**

**PROJECTED STATE COMPUTATION USING SECTION 3.1.2.3 RULES**

<table>
<thead>
<tr>
<th>Period</th>
<th>Capacity</th>
<th>Demand</th>
<th>Simple State</th>
<th>Projected State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>4</td>
<td>( E_{1,1} )</td>
<td>( E_{1,1} )</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>( O_{1,2} = 2 )</td>
<td>( O_{1,2} = 2 )</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>( O_{1,3} = 1 )</td>
<td>( O_{1,3} = 3 )</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>2</td>
<td>( U_{1,4} = 3 )</td>
<td>( E_{1,4} )</td>
</tr>
</tbody>
</table>

This procedure of projecting overloads over future time periods allows for the calculation of a more accurate estimate of total overloads and, therefore, the total delays required at a terminal during a specified time frame. This fact is utilized heavily in the solution procedure discussed in chapter 4.

3.2 Delay Costs

The mechanics of a flight delay and the effects a delay has on shifting demand and reducing overloads were considered in the preceding sections. The cost of a flight
delay was neglected in the previous sections in favor of stressing these mechanics. In this and several following sections, the cost of a flight delay is discussed and integrated into the total problem definition.

Flight delay costs are dependent on several intangible factors which make an "exact" cost of a flight delay difficult or impossible to calculate. However, from a practical viewpoint it is necessary to have some reasonable "estimate" of individual flight delay costs in order to proceed with the development of an algorithmic solution procedure. In addition, a discussion of the delay costs should provide further insights into the CF problem.

It is important to note here that the algorithm developed in chapter 4 accepts individual air and ground delay costs for each flight with the objective of minimizing total delay costs. The algorithm in chapter 4 is able to accept any delay cost estimates, but obviously the more accurate the cost estimates (input) the more reliable the computed results are. Sensitivity analysis can be utilized to determine the effect of inaccurate cost estimates.

The objective in the following discussion is to consider several possible procedures for estimating delay costs, but recommendations are not made as to the "one best" estimating procedure. The selection of such a delay cost estimating procedure is a research question in itself
and is an important portion of the FAA data collection and automation research presently being conducted.

Efforts are made in the following discussion to find an estimation procedure which balances the accuracy of the cost estimates against the computational requirements in terms of computer time and cost. Again, the procedures presented are not submitted as preferred or as the methods which best balance the accuracy of the estimates against computation time and cost. Other procedures may yield superior estimates with less computations.

3.2.1 Air Delay Cost Estimation

The air delay cost of a flight depends on five factors: fuel costs, maintenance costs, crew costs, safety costs and passenger related costs. The fuel costs and maintenance costs are easily quantifiable, while the safety costs and the passenger related costs are difficult to quantify.

The fuel costs of an air delay of a particular flight are based on the aircraft's characteristics, such as the number, size and efficiency of the engines, the weight of the aircraft, the natural lift it possesses and the weight of the load it is carrying. The fuel consumption figures based on the above factors are available from airline sources and can be utilized to obtain close estimates for each individual flight $f$. For example, a
Western B-707-300C estimate of fuel cost is $1.1667/air mile while the industry average is $1.2571/air mile (6), (14).

Define:

$\phi_f$ as the estimated fuel cost of air delaying flight $f$ for one unit of time.

The maintenance costs related to an air delay are definable in terms of the preventative maintenance schedule and costs. An air delayed aircraft increases the preventative maintenance costs by consuming portions of the preventative maintenance cycle creating the need for more maintenance costs per year.

Define:

$\omega_f$ as the preventative maintenance cost for an air delayed flight, $f$, for one period of time.

Crew costs are significant for air delays, but since crews are salaried personnel, the cost is pertinent only when crew turnaround is altered.

Define:

$\psi_f$ is a incremental crew cost (0 if turnaround unaltered or an idleness cost plus holdover expenses if turnaround is altered).

The quantification of safety costs related to air delays is extremely difficult in that it requires the evaluation of risk and the placement of a value on human life. One crude quantification is accomplished as follows:
$S_f = \beta + KN_f + \gamma_e$ \hspace{1cm} (3.1)

where $\beta$ is a significantly large safety cost applied to the delay of any aircraft regardless of number of passengers. This might include the value of the lives of the crew and the ground damage if a crash occurred.

$K$ is an additional safety cost applied per passenger on an air delayed flight.

$N_f$ is the number of passengers on flight $f$.

$\gamma_e$ is an air collision cost which is a function of the delay environment (e.g., number of flights already circling the terminal).

$S_f$ is the total safety cost related to flight $f$.

Due to the qualitative nature of $\beta$ and $K$ it may be necessary to resort to questionnaires submitted to pilots, controllers and FAA officials to obtain estimates. To obtain the value of $N_f$, the number of passengers on flight $f$, can be accomplished by utilizing the seating capacity limitations for the type of aircraft used on the flight, or by referencing the airlines reservation system for exact passenger demands.

The dynamic nature of airspace congestion over high density terminals makes it extremely difficult to estimate $\gamma_e$. However, if it is possible to keep track of the number of aircraft air delayed during each time period, then an estimated collision cost for each additional air
delay could be applied.

Estimation procedures for passenger related costs can cover a wide range between two extremes. One extreme is to use a simple constant for each flight which neglects such considerations as the number of passengers on a flight and the number of missed connections caused by a single flight delay or several periods of flight delay.

The most precise method for estimating passenger related costs for an air delay, involves an examination of each passenger's itinerary to determine if an air delay of one period (e.g., an hour) results in a missed connection. The cost of missing a connection may include such items as overnight accommodations, meals, taxi service, reticketing of passengers, additional luggage handling and expediting, and lost customer goodwill, each of which is dependent on delay time. The evaluation of each passenger's situation and computation of a cost related to any missed connections would require an extensive data bank and an extravagant amount of computations, and is probably impractical.

Between these two extremes exist many alternatives, one of which is to base the passenger related cost for a flight entirely on a proportion of the average number of passengers frequenting the flight. That is,

\[ C_f = LPN_f \]  

(3.2)
where \( L \) is the average cost of a missed connection for an average length of delay.

\( P \) is the average proportion of passengers experiencing missed connections when they experience an average length of delay.

\( N_f \) is as defined previously.

\( C_f \) is the total estimated passenger related delay cost for flight \( f \).

The values of \( L \) and \( P \) could be obtained through examination of historical expenditures by airlines on passenger accommodations and other costs related to missed connections.

Total estimated air delay costs for an individual flight, \( f \), is the combined costs related to fuel, maintenance, safety and passengers and can be defined as

\[
T_f = \phi_f + \omega_f + \psi_f + S_f + C_f
\]

\[
= \phi_f + \omega_f + \psi_f + \beta + (K + LP)N_f + \gamma_e
\]

where, \( T_f \) is the total estimated air and the other terms are as defined previously. Equation (3.4) consists of costs dependent on the flight, costs dependent on number of passengers on the flight, costs dependent on airspace congestion, and a constant cost independent of the aircrafts characteristics or the number of passengers.

3.2.2 Ground Delay Cost Estimation

The major costs incurred by initiating a ground delay
are crew and passenger related. Fuel, maintenance, and safety costs appear insignificant for a ground delay when compared to an air delay. Hence, the estimated total cost of a ground delay can be written simply as

\[ T_f' = C_f + \psi_f \quad (3.5) \]

\[ = LPN_f + \psi_f \quad (3.6) \]

where \( T_f' \) is the total estimated ground delay cost.

\( L \) is the average cost of a missed connection for an average length of delay.

\( P \) is the average proportion of passengers experiencing a missed connection when an average delay length is insured.

\( N_f \) is the number of passengers on flight \( f \).

\( \psi_f \) is an incremental crew cost for flight \( f \).

Equation (3.6) is dependent only on the number of passengers on the flight and the crew size and salaries. Contrasting this to equation (3.4) indicates that ground delay costs are independent of aircraft characteristics other than seating capacity and crew size and salaries. This is justified in that a ground delayed aircraft is idle, and is therefore unaffected by characteristics such as engine efficiency, or lift characteristics. This discussion assumes that during ground delays all engines are shut down.

Recognizing the subjective nature of the discussion of delay costs makes it appropriate to restate that no
attempt is made here to propose an optimal cost evaluation method. Instead the attempt here is to provide further insights into the problem.

3.3 Objective of Problem

The objective of the $CF^2$ problem has been alluded to throughout the previous discussion. Further clarifying of the objective in verbal terms is necessary before discussing the abstraction of the problem, which follows in the next section. The objective is to find the least costly set of air and ground delays if such set exists which will smooth demand forward in time to insure that no terminal-periods are overloaded. An extension and an assumption which guarantees the existence of a feasible solution are provided in section 4.1.

3.4 Formulation of the Problem as a Tree

A convenient way of examining the structure and the magnitude of the problem is provided by a tree structure. First the flights are numbered sequentially from one to $N$, where $N$ is the number of flights. For simplicity assume that a flight can be delayed only once, either an air or a ground delay.

An example to illustrate the tree structure is now convenient. Figure 3.5 is a three-flight problem with overloads existing at terminal 1, during periods 1 and 2. The flight data is provided in table 3.6. For example,
TABLE 3.6
INITIAL DATA FOR A THREE-FLIGHT PROBLEM

<table>
<thead>
<tr>
<th>Flight</th>
<th>From</th>
<th>To</th>
<th>Air Cost</th>
<th>Ground Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1,1)</td>
<td>(1,2)</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>(1,1)</td>
<td>(2,2)</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>(2,1)</td>
<td>(1,2)</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 3.6 states that Flight 1 departs from terminal 1, during period 1 (i.e., (1,1)) and is scheduled to land at terminal 1 during period 2 (i.e., (1,2)). A flight departing from and landing at the same terminal is meaningful when the plane visits and returns from a secondary terminal outside the major terminal problem.

The costs in Table 3.6 are air and ground delay costs and are arbitrarily chosen for this example. If Flight 1 is delayed in the air over its destination, the cost is assumed to be five units. A ground delay of Flight 1 is assumed to cost only three units.

The tree in figure 3.6 represents the possible solutions (some feasible for the problem in figure 3.5).

Fig. 3.5. A three-flight problem.
Fig. 3.6. Tree diagram for a three-flight problem.
In figure 3.6 the nodes contain the flight number and the type of delay. For example, 3a corresponds to the air delay of Flight 3. The node with a dash means no delay. The arcs in figure 3.6 connect the nodes or delays to represent the possible delay combinations. For example, arcs connecting nodes 1g, 2g, and 3g form one possible combination; that is, ground delay Flights 1, 2, and 3. The arcs ending with an S indicate that one possible solution exists. For example, the arc leaving node 1g and having an S represents the possible combination, ground delay Flight 1 only.

It is observed that for this simple three-flight problem that 27 possible solutions exist. Of course, not all these solutions are feasible. Recall that feasibility depends on the terminal capacities and demands at various time periods. Examining the feasibility of a solution is accomplished by moving the arcs in figure 3.5 as specified by the combination. For example, moving the arcs 1g, 2a, 3a results in figure 3.7 which is a feasible solution.

![Diagram of feasible solution](image)

Fig. 3.7. A feasible solution to the three-flight problem of figure 3.5.
The feasibility of the solution in figure 3.7 is provided by counting the number of arcs (demands) entering or leaving a node and noting that the count is less than or equal to the capacity number inside the node. Table 3.7 provides the modified flight data corresponding to delays lg, 2a, 3a, and table 3.7 corresponds to figure 3.7.

**TABLE 3.7**

<table>
<thead>
<tr>
<th>Modified Flight</th>
<th>From</th>
<th>To</th>
<th>Applicable Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1,2)</td>
<td>(1,3)</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>(1,1)</td>
<td>(2,3)</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>(2,1)</td>
<td>(1,3)</td>
<td>6</td>
</tr>
</tbody>
</table>

Comparing table 3.6 to table 3.7 indicates that flight 1 departs a period later and lands a period later and flights 2 and 3 depart on time but arrive a period later. The costs associated with the ground delay of flight 1 and the air delays of flights 2 and 3 are shown under the title "Applicable Costs." For this particular delay combination the total delay cost was 13 units. If all combinations in figure 3.6 are evaluated and tested for feasibility, it is found that lg, 2a, 3g is the optimal feasible solution with a value of 10.

A three-flight problem with only one air or ground delay per flight permitted has a total of 27 possible solution combinations. For the general case of N flights,
the number of combinations, \( K \), is defined as

\[
K = 3^N
\]  

(3.7)

The logic behind (3.7) can be seen by recognizing that three possibilities exist for each flight as follows:

1) air delay
2) ground delay
3) no delay.

Using (3.7) for the arbitrary case of \( N=108 \) yields approximately \( 3.375 \times 10^{40} \) possible combinations, an astronomically large tree. If the assumption that each flight can be delayed only one period is relaxed to allow two-period delays per flight the formula for \( K \) is

\[
K = 6^N
\]  

(3.8)

since there are six possible two-period delays for one flight as follows:

1) Air, air
2) Air, none
3) Ground, air
4) Ground, ground
5) Ground, none
6) None, none

Obviously (3.8) yields even more inflated figures. For the general case of unlimited delays of a single flight a formula is meaningless since the number of combinations becomes enormous, and since a flight would not normally be so delayed but would instead be canceled. Cancellation of flights is considered as an extension in section 3.5.2.

The explosive nature of the tree structure demonstrates the need for a methodology more effective
computationally than total enumeration provides. Chapter 4 is devoted to the development of such a method. However, it is first necessary to consider several extensions not considered in the literature.

3.5 Extensions Not Considered in the Literature

3.5.1 Multiple Period Delays of the Same Aircraft

The previous section's discussion of two or more periods of delay of the same flight is an extension not considered in the literature. The desirability of two or more periods of delay of the same flight is dependent on the length of the time period. For example, using a time period of fifteen minutes yields a thirty-minute two-period delay which in practice often occurs. Using a time period of one hour yields a two-hour lapse for a two-period delay, and from a practical standpoint this is less desirable. The algorithm discussed in chapter 4 considers this extension explicitly. The algorithm assumes that for each additional delay of the same flight, a proportionately higher delay cost is incurred. This can be expressed as

\[ T_{f2} = qT_{f1} \]  \hspace{1cm} (3.9)

\[ T_{f3} = qT_{f2} = q^2T_{f1} \]  \hspace{1cm} (3.10)

where \( q \) is a constant and is determined by experience and consensus. In practice, \( q \) will always be
greater than or equal to one.

\( T_{fi} \) is the cost of delaying flight \( f \) the \( i \)-th successive period.

An example illustrates the above equations. Consider a \( q \) value of 1.5, which means each successive flight delay of the same aircraft is 1.5 times more expensive than the previous delay cost. Further consider an initial delay cost of five units (i.e., \( T_{fi} = 5 \)). Then the first delay costs five units, the second delay costs 7.5 units (i.e., \( T_{f2} = qT_{fi} = 5(1.5) \)), and a third delay costs 11.25 units (i.e., \( T_{f3} = qT_{fi} = 7.5(1.5) \)). The total delay cost of delaying the aircraft for three periods is then 23.75 units (i.e., \( 5 + 7.5 + 11.25 \)).

The rationale for this successively increasing cost of delay is to discourage the algorithm from delaying the same aircraft repeatedly and thus distributing the delay costs more equitably among all aircraft. Certain situations such as severe weather arise causing repeated delays of the same flight to be necessary. The assumption just stated permits this situation.

The algorithm in chapter 4 permits the case in which each successive delay cost is unchanged. This can be accomplished by assigning a value of one for \( q \).

3.5.2 Cancelation of Flight Departures

The cancelation of a flight's departure due to
excessive waiting is applicable if the airlines specify certain flights which may be canceled when delay time reaches a specific threshold value. This decision to cancel must balance two costs: the cost of cancelation versus the cost of extended delay. The cancelation cost is likely to include passenger goodwill, passenger housing and board, and additional crew costs. The delay costs include the costs discussed previously under ground delay costs.

The incorporation of this extension into the algorithm is examined in section 4.9.1. The algorithm assumes that cancelation costs are at least as large as delay costs. This assumption is also examined in detail in section 4.9.1.

3.5.3 Multiple Leg Extension

Up to this point it is implicitly assumed that each flight leg is serviced by a unique aircraft. That is the case in which one aircraft services two, three or more flight legs is not considered. For example, a single aircraft might depart from New York, stop in Pittsburgh, depart again for Cleveland, and finally terminate in Chicago. In the initial formulation, this would be considered as three separate flights:

i) New York to Pittsburgh
ii) Pittsburgh to Cleveland
iii) Cleveland to Chicago

With this initial formulation it can happen that the New
York to Pittsburgh flight is delayed to the extent that
the arrival time in Pittsburgh is later than the departure
time. That is, with this initial formulation a delay can
be prescribed which specifies that an aircraft should
depart before it lands.

To allow for the downstream ramifications of the
delay of a multiple leg flight requires the incorporation
of several features into the algorithm. These features
are considered in section 4.9.2.

An additional complication occurs if slack time
exists between the arrival and scheduled departure times
at an interim terminal. For example, if the delay time
for the New York to Pittsburgh flight is less than the
slack time available between landing and departure at
Pittsburgh then the delay of the Pittsburgh to Cleveland
and Cleveland to Chicago legs are unnecessary since the
slack time can be used as "catch-up" time.
4. COMBINATORIAL PROGRAMMING ALGORITHM

The tree structure introduced in chapter 3 demonstrated the magnitude of this combinatorial problem. In addition, chapter 3 discussed a means for evaluating the effect of each flight delay in reducing overloads.

Using these foundations, this chapter builds a complete algorithm which either implicitly or explicitly searches each branch of the tree and finds an optimal solution. The algorithm is first presented for the problem described in section 3.1 with the extension that a flight can be delayed more than one period. The algorithm is then modified to include the two additional extensions discussed in chapter 3.

In this chapter, the term optimality refers to the minimization of total delay costs. Efficiency refers to computation times of not more than five minutes on a high speed computer for practical problems having as many as twenty air terminals and 2,500 to 3,500 flight operations. The computer utilized to test the efficiency of the algorithm was a Control Data CDC 6400. The results of these tests are discussed in chapter 5. In chapter 5, the computer program for the combinational programming algorithm includes the extension that a flight can be
delayed more than one period. A zero-one algorithm is introduced in section 4.10 which incorporates all three extensions and in section 5.2, a test computer run is provided.

4.1 General Discussion of Methodology

The algorithm utilizes the principles of implicit enumeration or combinational programming [see (39) and (58)]. The philosophy of these approaches is to begin searching the one branch among all the possible branches (or paths) in a tree which "heuristically appears" to lead to an optimal and feasible solution. In the context of the CF² problem the search begins with a set of infeasible (i.e., overloaded) conditions and searches for a feasible solution which "heuristically appears" to be a candidate for optimality.

Once a feasible, but not necessarily optimal solution is found the search is reduced to finding a feasible solution which is lower in cost than the least costly solution yet found. If no solution of lesser cost can be found then the least costly solution yet found is optimal and search is terminated.

This method is primal in nature since it moves from one feasible solution to another solution each time reducing cost until the optimal solution is reached. Converging to an optimal solution is then guaranteed by the
incremental improvement since a non-negative lower limit exists (i.e., no solution exists with a value less than zero).

The existence of a feasible solution is guaranteed by assuming an infinite capacity for the last time period for each terminal. Hence, one feasible solution is to delay all flight operations to the last time period. This assumption is realistic since an optimal feasible solution will minimize total delay cost and, therefore, utilize the capacity of these last time periods in an optimal manner. The assumption is also well founded on the basis that accurate prediction of capacity and demand more than five hours in advance is not presently possible.

In figure 3.6 a tree diagram shows all possible solutions for a three-flight problem. Certainly many of these solutions are infeasible. That is, they do not relieve all overloaded situations. For example, the solution no delays is normally infeasible since overloads usually exist. Likewise, if three overloads exist it is obvious that one flight delay will not relieve all overloads since in table 3.2 it was shown that one flight delay can at most reduce two overloads. Recognizing these facts, it is now useful to define two terms which are referred to repeatedly in this chapter. A partial solution or solution is any solution which contains zero or more flight dealys. A feasible solution is a solution in which
all overloads are relieved. These new terms accurately reflect the methodology of the algorithm which begins with a partial solution and appends delays until a feasible solution is attained.

This methodology is combined with the concept of deleting delays from a complete or partial solution to arrive at a previously generated partial solution. The appending procedure is termed construction and the deletion is referred to as backtracking. Backtracking is important in the process of backing up from a feasible solution to a partial solution in order to construct new feasible solutions and, thereby, implicitly enumerate all possible feasible solutions represented by the tree.

To completely specify the algorithm requires discussion of the following methods:

1) A method for computing projected overloads (see section 3.1.3) for all terminals and time periods.

ii) A method for computing the overload reductions or increase for each air or ground delay of each flight.

iii) A method of selecting a flight delay from all possible delays not yet considered and appending it to a partial solution to form a new partial solution or a feasible solution. The method of selection must insure the reduction of overloads and should heuristically move toward one of the candidates for the optimum solution.

iv) A method for determining when a feasible solution has been reached.
v) A method for saving the least costly feasible solution yet found.

vi) A method for systemically decomposing a feasible solution and searching for a lower cost feasible solution if one exists.

vii) A method for eliminating search of portions of the tree by showing they do not contain candidates for the optimum solution. This in turn requires a method for computing a lower cost bound for any node in the tree. This bound for a given node represents a lower limit on the values of any feasible solution which contains that node. That is, no feasible solution containing the node will have a value lower than the node's lower bound. Search is reduced by comparing the lower bound to the value of the least costly solution yet found. If the lower bound for a branch is larger than the least costly solution yet found then search of that branch of the tree is unnecessary. Otherwise, search of that branch is pursued.

To illustrate the methods and the complete algorithm, a simple example problem is followed throughout the discussion. The problem consists of eight flights and allows for multiple period delays of the same flight. The tree size for an eight-flight problem allowing a maximum of only two delays per flight is $6^8$ or 1,679,616 branches.

4.2 Overload Computation and a Simple Lower Bound Allowing Multiple Delays of a Flight

In chapter 3, the concept of projected overloads for terminal-periods is discussed. The discussion leads to the conclusion that reduction of an overload at terminal $i$ during period $t$ by a delay will cause an increase in overloads at $t + 1$ if the state of terminal $i$ during $t + 1$ prior to the delay was either equilibrium or overload.
This leads to a method for computing projected overloads at each terminal-period. Projected overloads can be thought of as the obvious overloads, plus some not so obvious overloads.

The method to be employed is to first compute the difference between demand and capacity and recognize one of the three states overloaded, equilibrium, or underload for each terminal-period. This is referred to as determining the simple state of the terminal-period. Then projection of overloads is accomplished using the methods of section 3.1.3 (also refer to appendix 8.1).

The methods of section 3.1.3 are based on the observation that the only way in which an overload in period t can be relieved is by shifting a demand for an operation from period t to period t + 1, regardless of whether the state of t + 1 is overloaded, underloaded, or in equilibrium. Suppose that t is overloaded by, say, two operations, t + 1 is overloaded by one operation, t + 2 is in equilibrium, and t + 3 is underloaded by three operations. It is possible to say a priori that the actual (or projected) state during t is still two overloads, the actual or projected state during period t + 1 is one overload plus the two overloads transferred from period t, or period t + 1 is overloaded by three operations, the projected state of period t + 2 is three overloads, (i.e., zero overloads plus the three overloads transferred from
period $t + 1$), and the projected state of period $t + 3$ is negative three plus the three overloads transferred from period $t + 2$, so the projected state of period $t + 3$ is equilibrium. Then, for this example, it appeared initially that only three overloads existed (i.e., $2+1+0+0$). But the actual or projected number of overloads is eight (i.e., $2+3+3+0$). This is an important revelation because it provides a more accurate estimate of the number of delays which will be required to reduce all overloads to zero. Also, the projection method locates many overloaded terminal-period situations which were originally thought to be underloaded in equilibrium, or less severely over­loaded.

Perhaps most importantly the projected state provides a "look ahead" feature for computing the effects of a ground or air delay in terms of projected overload reduction. For example, if the projected state of $(i,t)$ is an overload, it is irrelevant what the projected state of $(i,t+1)$ is because the projection method is already considering the result of the demand shift. When tables 4.3 and 4.4 are viewed, these observations should be kept in mind.

Figure 4.1 and table 4.1 define the example problem used throughout the discussion of the projected overloads and the entire algorithm. The notation and concepts introduced in section 3.1.3 are utilized in interpreting
Fig. 4.1 Requested flight patterns for example problem.

TABLE 4.1
PROJECTED OVERLOAD COMPUTATION
TABLE FOR EXAMPLE PROBLEM

<table>
<thead>
<tr>
<th>Terminal</th>
<th>Period</th>
<th>Capacity</th>
<th>Demand</th>
<th>Simple State</th>
<th>Projected State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>$O_{1,1}=1$</td>
<td>$O_{1,1}=1$</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>$U_{1,2}=1$</td>
<td>$E_{1,2}$</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>$U_{1,3}=1$</td>
<td>$U_{1,3}=1$</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>$U_{1,4}=1$</td>
<td>$U_{1,4}=1$</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>$E_{2,1}$</td>
<td>$E_{2,1}$</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>$E_{2,2}$</td>
<td>$E_{2,2}$</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>$O_{2,3}=1$</td>
<td>$O_{2,3}=1$</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>$U_{2,4}=1$</td>
<td>$E_{2,4}$</td>
</tr>
</tbody>
</table>
figure 4.1 and table 4.1. Using the projection method on terminal one of the example problem illustrates the concepts. Referring to table 4.1, note that the simple state of period 1 is overloaded since demand exceeds capacity by one unit \((0_{1,1}=D_{1,1}-C_{1,1}=3-2=1)\). The actual or projected state will also be one. Since period 2 is underloaded by one unit \((U_{1,2}=C_{1,2}-D_{1,2}=4-3=1)\) it could be used to offset the period 1 overload. The result is that the projected state for period 2 is equilibrium \((E_{1,2} = U_{1,2} - O_{1,1}=1-1=0)\). The period 1 projected state is still one overload. Periods 3 and 4 are underloaded and, therefore, no projection is required. For terminal 2 the simple states of period 1 and 2 are equilibrium, so no projection is necessary. Since period 4 is underloaded by one operation, the one projected overload from period 3 results in a projected state of equilibrium for period 4.

From table 4.1 the total number of projected overloads is then two since \((1,1)\) and \((2,3)\) are overloaded by one unit each. For a more general case, refer to section 3.1.3 in which overloads are projected forward by two periods.

Since in the example problem two overloads exist, it can be reasoned that at least one ground delay is required. This is due to the fact that the maximum overload reduction for a ground delay is two units. If only air delays are utilized, at least two air delays are required to reduce
the overloads to zero, since the maximum reduction for an air delay is one unit.

To pursue the analysis further requires the knowledge of air and ground delay costs for each flight. Table 4.2 shows these two costs for each flight. The air delay costs are substantially higher than the ground delay costs as per the explanation in section 3.2 (and its subsections) on delay costs.

<table>
<thead>
<tr>
<th>Flight Index (j)</th>
<th>Flight Definition</th>
<th>Air Delay Cost (a_j)</th>
<th>Ground Delay Cost (g_j)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1,1)-(1,2)</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>(1,1)-(1,3)</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>(1,1)-(2,2)</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>(1,2)-(2,3)</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>(2,1)-(1,2)</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>(2,1)-(2,3)</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>(2,2)-(1,3)</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>(2,3)-(1,4)</td>
<td>10</td>
<td>8</td>
</tr>
</tbody>
</table>

As a by-product of the above analysis, it is now possible to calculate a lower cost bound on any feasible solution to the problem defined by tables 4.1 and 4.2. This particular lower bound is for illustrative purposes only. A more appropriate method is introduced in section 4.4. Since a minimum of one ground delay or two air delays
are required to reduce the two overloads, the lower bound (B) is
\[
B = \min \left[ \min (g_j), 2 \min (a_j) \right] \quad \forall j \quad \forall j
\]
where \( g_j \) is the ground delay cost for flight \( j \) and \( a_j \) is the air delay cost for flight \( j \). In the example problem the lower bound is
\[
LB = \min [4, 2(6)] = 4
\]
(Note: This lower bound applies to this example problem only.)

This implies that no feasible solution exists with a value less than four. In a later discussion the lower bound calculations are refined to achieve the largest possible lower bound which has a value less than or equal to all feasible solutions (see section 4.11).

4.3 Selection of a Flight Delay

In section 4.1 the concepts of a partial solution and a complete solution are discussed. At this point the questions are which flight delay to select to form a partial solution, and is the partial solution a complete solution (i.e., feasible)?

Any flight delay selection procedure which produces a unique ranking of the flight delays is satisfactory to provide a controlled enumeration of all possible solutions. For example ranking the delays by delay cost (and if a tie exists, ranking by index number) is one
possibility. This means that (see table 4.2) a ground
delay of flight index one (G-1) is first in the ranking
(at a cost of four units) followed by a ground delay of
flight index five (G-5). The last delay in the ranking
is an air delay of flight index eight (A-8).

This heuristic ranking has some appeal due to the
cost considerations. But, as discussed in sections
3.1.2.1 through 3.1.2.3, the delay of a flight can result
in either an increase, decrease, or no change in overloads.
This factor is not reflected in the above cost ranking.

A heuristic ranking which reflects both the over­
load and cost factors is more appropriate. One such
heuristic ranking is provided by the ratio, \( F_d \), of over­
load reduction for delay \( d \) divided by the cost of delay
(not flight) \( d \). The ratio is formalized as
\[
F_d = \frac{I_d}{K_d} \quad \text{for } d = 1, 2, 3, \ldots, 2n \tag{4.2}
\]

where \( I_d \) is the overload (or infeasibility) reduction
for a ground delay if delay \( d \) is a member of
the set of ground delays. Otherwise, \( I_d \) is
the overload reduction for an air delay if \( d \) is
a member of the set of air delays.

\[
K_d = \begin{cases} 
g_j & \text{if delay } d \text{ is a ground delay of flight } j. 
a_j & \text{if delay } d \text{ is an air delay of flight } j. 
\end{cases}
\]

Recall that \( g_j \) and \( a_j \) are the ground and air delay
costs of flight \( j \). The \( 2n \) in (4.2) accounts for both
ground and air delays of each of the \( n \) flights.
To calculate $I_d$, the overload reduction for a particular ground or air delay $d$ requires a knowledge of the following:

i) Type of delay (i.e., air or ground)

ii) Origin terminal period and destination terminal period of the flight (Recall: $(i,t)$ for the origin and $(i',t+k)$ for the destination).

iii) The projected states (i.e., overloaded, equilibrium or underloaded) of terminal-periods $(i,t)$, $(i,t+1)$, $(i',t+k)$, $(i',t+k+1)$.

Recall that a ground delay shifts demand at both the origin and destination terminals so overload reduction depends on all four terminal periods stated in (iii) above. Thus the projected state of each of these terminal periods must be considered in calculating overload reductions. The nine relevant combinations of conditions or cases are shown in table 4.3. Table 4.3 uses 0 for overloads, E for equilibrium, and U for underloads.

For an air delay of flight $(i,t) \rightarrow (i',t+k)$ the overload reduction is dependent only on the status of $(i',t+k)$ and $(i',t+k+1)$. This is due to the fact that an air delay shifts demand at the destination terminal only. The three possible cases which can occur are shown in table 4.4 using 0, E, and U as just defined.

Tables 4.1 and 4.2 and the method for computing overload reductions for a ground and air delay just discussed make it possible to compute $F_d$ for each possible delay $d$. For example the ratio for a ground delay of
TABLE 4.3
OVERLOAD REDUCTIONS FOR GROUND DELAYS AS A FUNCTION OF THE PROJECTED STATES (O, E, OR U) OF THE RELEVANT TERMINAL-PERIODS

<table>
<thead>
<tr>
<th>Case</th>
<th>Projected State of ((i, t))</th>
<th>Projected State of ((i, t+1))</th>
<th>Projected State of ((i', t+k))</th>
<th>Projected State of ((i', t+k+1))</th>
<th>Overload Reduction (I_d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0, E or U</td>
<td>0</td>
<td>0, E or U</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0, E or U</td>
<td>E or U</td>
<td>U</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0, E or U</td>
<td>E or U</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>E or U</td>
<td>U</td>
<td>0</td>
<td>0, E or U</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>E or U</td>
<td>U</td>
<td>E or U</td>
<td>U</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>E or U</td>
<td>U</td>
<td>E or U</td>
<td>0 or E</td>
<td>-1</td>
</tr>
<tr>
<td>7</td>
<td>E or U</td>
<td>0 or E</td>
<td>E or U</td>
<td>U</td>
<td>-1</td>
</tr>
<tr>
<td>8</td>
<td>E or U</td>
<td>0 or E</td>
<td>E or U</td>
<td>0 or E</td>
<td>-2</td>
</tr>
<tr>
<td>9</td>
<td>E or U</td>
<td>0 or E</td>
<td>0</td>
<td>0, E or U</td>
<td>0</td>
</tr>
</tbody>
</table>

TABLE 4.4
OVERLOAD REDUCTIONS FOR AIR DELAYS AS A FUNCTION OF THE PROJECTED STATES (O, E, OR U) OF THE RELEVANT TERMINAL-PERIODS

| Case | Projected State of \((i', t+k)\) | Projected State of \((i', t+k+1)\) | Overload Reduction \(I_d\) |
|------|-----------------|-----------------|-----------------|-----------------|
| 1    | 0               | 0, E or U       | 1               |
| 2    | E or U          | U               | 0               |
| 3    | E or U          | 0 or E          | -1              |

flight 1 (see table 4.2) shows the computation of overload reductions. From table 4.2 it is observed that the terminal-periods involved are \((1,1)\) and \((1,2)\). From table 4.1 it is noted that \((1,1)\) is overloaded and \((1,2)\) is in equilibrium while \((1,3)\) is underloaded. The
conclusion is that one overload is reduced (case 2). To
compute $F_1$ for delay one (ground delay of flight 1) it
is only necessary to divide by the ground delay cost $g_1$
for flight one (from Table 4.2). The resulting ratio is
then $1/4$ or $0.25$. Table 4.5 shows overload computations
and the ratio for each delay.

**TABLE 4.5**

OVERLOAD REDUCTION AND RATIO CALCULATIONS FOR
INITIAL AIR AND GROUND DELAYS OF EACH
FLIGHT IN THE EXAMPLE PROBLEM

<table>
<thead>
<tr>
<th>Delay Index (d)</th>
<th>Type Delay (G or A)</th>
<th>Flight Definition</th>
<th>Overload Reduction ($I_d$)</th>
<th>Delay Cost ($K_d$)</th>
<th>Reduction Cost ($F_d$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G</td>
<td>$(1,1) \rightarrow (1,2)$</td>
<td>1</td>
<td>4</td>
<td>$1/4$</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>$(1,1) \rightarrow (1,2)$</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>G</td>
<td>$(1,1) \rightarrow (1,3)$</td>
<td>1</td>
<td>5</td>
<td>$1/5$</td>
</tr>
<tr>
<td>4</td>
<td>A</td>
<td>$(1,1) \rightarrow (1,3)$</td>
<td>0</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>G</td>
<td>$(1,1) \rightarrow (2,2)$</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
<td>$(1,1) \rightarrow (2,2)$</td>
<td>-1</td>
<td>9</td>
<td>$-1/9$</td>
</tr>
<tr>
<td>7</td>
<td>G</td>
<td>$(1,2) \rightarrow (2,3)$</td>
<td>1</td>
<td>5</td>
<td>$1/5$</td>
</tr>
<tr>
<td>8</td>
<td>A</td>
<td>$(1,2) \rightarrow (2,3)$</td>
<td>1</td>
<td>7</td>
<td>$1/7$</td>
</tr>
<tr>
<td>9</td>
<td>G</td>
<td>$(2,1) \rightarrow (1,2)$</td>
<td>-1</td>
<td>4</td>
<td>$-1/4$</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
<td>$(2,1) \rightarrow (1,2)$</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>G</td>
<td>$(2,1) \rightarrow (2,3)$</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>A</td>
<td>$(2,1) \rightarrow (2,3)$</td>
<td>1</td>
<td>8</td>
<td>$1/8$</td>
</tr>
<tr>
<td>13</td>
<td>G</td>
<td>$(2,2) \rightarrow (1,3)$</td>
<td>-1</td>
<td>5</td>
<td>$-1/5$</td>
</tr>
<tr>
<td>14</td>
<td>A</td>
<td>$(2,2) \rightarrow (1,3)$</td>
<td>0</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>G</td>
<td>$(2,3) \rightarrow (2,4)$</td>
<td>1</td>
<td>8</td>
<td>$1/8$</td>
</tr>
<tr>
<td>16</td>
<td>A</td>
<td>$(2,3) \rightarrow (2,4)$</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.5 provides a unique ranking of all delays
first by decreasing ratio value and when ties exist by
increasing index number. The largest ratio is $1/4$ for
delay index one (i.e., a ground delay of flight 1). This
delay is then chosen as the first entry in the first partial solution.

4.4 A Refinement in the Lower Bound Computation

This section uses the additional knowledge provided by table 4.5 to refine and improve the lower bound calculations of section 4.2. Table 4.5 indicates that the maximum overload reduction using any delay is one unit. This fact and the fact that two overloads exist means that at least two delays are required to relieve all overloads. To determine a lower bound for constructing a feasible solution requires a short computation procedure. The flow diagram in figure 4.2 describes the procedure. This lower bound is not the maximum lower bound, but it is computationally efficient. The procedures in figure 4.2 are used to calculate only lower bounds and do not attempt to find feasible solutions. The lower bound for the example problem is (from table 4.5).

\[ B = K_1 + K_3 = 4 + 5 = 9 \]

This value comes from the fact that delay one costs four units and reduces overloads by one unit and delay three (having the next highest ratio) costs five units and also reduces overloads by one unit.

This lower bound calculation is reiterated each time a new partial solution is generated, as will be discussed in the following sections. As each additional partial
SET LOWER BOUND B EQUAL TO CURRENT SOLUTION VALUE

GIVEN THE NUMBER OF OVERLOADS REMAINING IS M AND GIVEN TABLE 4.5

SELECT THE DELAY d NOT YET SELECTED WHICH HAS THE HIGHEST RATIO $F_d$

ENTER THAT DELAY COST IN THE LOWER BOUND $B = B + K_d$

REDUCE THE VALUE OF M BY THE OVERLOADS RELIEVED $I_d$ $M = M - I_d$

IF M EQUALS ZERO

B IS THE LOWER BOUND

STOP

Fig. 4.2. Flow diagram for lower bound calculation.
solution is generated, the lower bound monotonically increases toward the value of the optimal solution.

Further observation of table 4.5 and table 4.1 show that the solution suggested by the lower bound is not feasible since observing the flight definitions of each delay shows that only the terminal-period (1,1) overload is considered (i.e., (1,1)−→(1,2) and (1,1)−→(1,3)). The terminal-period (2,3) overload is ignored, while (1,1) is considered twice. Hence, the lower bound ignores feasibility considerations and minimizes costs only as reflected in the ratios. In this manner, the lower bound is more likely to be less than the optimal solution value rather than equal to the optimal solution value. This characteristic of this particular lower bound can result in substantial search time because it is not as effective as possible in "trimming out" non-optimal candidates. The results quoted in chapter 5 include the lower bound calculations of figure 4.2. A more effective tree trimming lower bound is discussed in section 4.11; however, it is substantially more involved computationally than the flow diagram of figure 4.2.

One additional fact learned from the lower bound of nine units is that if a feasible solution with value nine should be generated it is an optimal solution, then no further search is required.
4.5 Preparation for Further Construction of a Feasible Solution

The partial solution, ground delay flight 1, is not feasible. Therefore, it is necessary to select another delay. This requires that an updated "picture" of the problem, including the above delay, be developed. The new "picture" similar to figure 4.1 is shown in figure 4.3.

The computational procedure does not, of course, require that figure 4.3 be drawn, but the figure does provide further insights. In particular figure 4.3 shows that (1,1) is no longer overloaded, (1,2) is now underloaded and (1,3) is now in equilibrium rather than underloaded. Further, it shows that (2,3) is still overloaded. In fact figure 4.3 shows that the only terminal-periods affected by the delay are (1,1), (1,2) and (1,3). This point is valuable in minimizing the overload reduction and ratio calculations required in reiterating that step of the solution procedure.

Fig. 4.3 Update of figure 4.1 showing flight (1,1)→(1,2) ground delayed one period.
The above hinted that it is now necessary to return to the computation of tables similar to 4.1 and 4.5; that is, recalculate projected overloads, overload reduction for each delay, and the ratio for each delay. Fortunately, the entire recalculation is necessary since only one to four terminal-periods are modified. The computation of new projected states is extremely simple, requiring only a subtraction, or an addition. For example, the ground delay of \((1,1)\rightarrow(1,2)\) modified the condition \(C\) of \((1,1)\) as follows:

\[
C = 0_{1,1} - 1 = 1 - 1 = 0 = E_{1,1}
\]  

(4.3)

This occurred because a ground delay at \((1,1)\) reduces a departure demand to \((1,2)\). Thus, the overload at \((1,1)\) is reduced to equilibrium at \((1,1)\).

The projected state of \((1,2)\) is modified from equilibrium to one underload. This is due to a shift of the destination terminal-period from \((1,2)\) to \((1,3)\). The condition of \((1,3)\) is modified from underloaded to equilibrium due to the increased demand at \((1,3)\). The new projected states are shown in Table 4.6.

Since flight \((1,1)\rightarrow(1,2)\) has been ground delayed to \((1,2)\rightarrow(1,3)\) it is necessary to assign a new delay cost and to compute the air and ground delay reductions and ratios for \((1,2)\rightarrow(1,3)\). Table 4.7 shows the results. Since a ground delay of \((1,2)\rightarrow(1,3)\) to \((1,3)\rightarrow(1,4)\) causes no change in the demand at \((1,3)\) (see fig. 4.3), it is appropriate to assume \((1,3)\) is underloaded when using table 4.3 for this
special case. Note that the delay costs have been doubled 
(q=2) to discourage further delay of the same flights. This 
increase by two-fold was arbitrary. Actually, other modi-
fications are possible.

TABLE 4.6
TRANSFORMING PREVIOUS PROJECTED TERMINAL-PERIOD 
STATE INCORPORATING A GROUND DELAY OF 
FLIGHT (1,1)→(1,2)

<table>
<thead>
<tr>
<th>Terminal</th>
<th>Period</th>
<th>Previous Projected State</th>
<th>New Projected State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0_{1,1}=1</td>
<td>E_{1,1}=0_{1,1}-1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>E_{1,2}</td>
<td>U_{1,2}=E_{1,2}-1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>U_{1,3}=1</td>
<td>E_{1,3}=U_{1,3}+1</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>U_{1,4}=1</td>
<td>U_{1,4}=1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>E_{2,1}</td>
<td>E_{2,1}</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>E_{2,2}</td>
<td>E_{2,2}</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0_{2,3}=1</td>
<td>O_{2,3}=1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>E_{2,4}</td>
<td>E_{2,4}</td>
</tr>
</tbody>
</table>

TABLE 4.7
NEW I_{d}, K_{d} AND F_{d} CALCULATIONS 
FOR DELAYED FLIGHT

<table>
<thead>
<tr>
<th>Delay Index (d)</th>
<th>Type Delay (G or A)</th>
<th>Flight Definition</th>
<th>Overload Reduction (I_{d})</th>
<th>Delay Cost (K_{d})</th>
<th>Ratio of Reduction to Cost (F_{d})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G</td>
<td>(1,2)→(1,3)</td>
<td>0</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>(1,2)→(1,3)</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>
Recognizing that only (1,1), (1,3) and (1,2) terminal-periods have changed allows a significant savings in computation of overload delay reductions and delay ratios. The savings in this small problem are not as significant as would occur for larger more realistic problems.

To illustrate how the savings occur, consider (1,1) which is now in equilibrium (previously overloaded). This means that all ground delays of flights departing or landing at terminal-period (1,1) must have new overload reductions and ratios calculated. Also, all air delays of flights landing at period (1,1) must have similar calculations performed. Table 4.8 shows the changes of table 4.5 required for flights involving (1,1) (refer to tables 4.5 and 4.6).

**TABLE 4.8**

MODIFICATIONS OF $I_d$ AND $F_d$ DUE TO CONDITION CHANGE AT (1,1) FROM OVERLOADED TO EQUILIBRIUM

<table>
<thead>
<tr>
<th>Delay Index $(d)$</th>
<th>Type Delay (G or A)</th>
<th>Flight Definition</th>
<th>Overload Reduction $(I_d)$</th>
<th>Delay Cost $(K_d)$</th>
<th>Ratio of Reduction to Cost $(F_d)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>G</td>
<td>(1,1)→(1,3)</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>G</td>
<td>(1,1)→(2,2)</td>
<td>-1</td>
<td>6</td>
<td>-1/6</td>
</tr>
</tbody>
</table>

Terminal-period (1,3) requires a different recalculation procedure because the state change was from underloaded to equilibrium. In this case, the concern is not for flights landing or departing at (1,3) since these will
not cause overload reduction or increase. Instead, the concern is for terminal-period (1,2) flights, since delay of these flights to (1,3), an equilibrium terminal-period, creates overloads which were previously rectified by the underloading of (1,3). This, therefore, requires that all ground delays of flights departing or landing at terminal-period (1,2) must have new overload reductions and ratios calculated. Also, all air delays of flights landing at terminal-period (1,2) must have similar calculations. Table 4.9 shows the changes required for all flights involving (1,2) except those calculated in table 4.7 (refer to tables 4.5 and 4.6).

**TABLE 4.9**

MODIFICATIONS OF \( I_d \) AND \( F_d \) DUE TO CONDITION CHANGE AT (1,3) FROM UNDERLOADED TO EQUILIBRIUM

<table>
<thead>
<tr>
<th>Delay Index ( (d) )</th>
<th>Type ( (G ) or ( A )</th>
<th>Flight Definition</th>
<th>Overload Reduction ( (I_d) )</th>
<th>Delay Cost ( (K_d) )</th>
<th>Ratio of Reduction to Cost ( (F_d) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>G</td>
<td>((1,2)^+(2,3))</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>G</td>
<td>((1,2)^-(2,1))</td>
<td>-2</td>
<td>4</td>
<td>-2/4</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
<td>((2,1)^-(1,2))</td>
<td>-1</td>
<td>6</td>
<td>-1/4</td>
</tr>
</tbody>
</table>

The above method required recalculation for only seven delays instead of sixteen delays for total recalculation. The procedure is somewhat burdensome for hand calculation of small problems. However, for computer
calculation for any size problem the procedure produces significant time savings over total recalculation of all overload reductions and ratios.

To provide clarity, tables 4.5, 4.7, 4.8, and 4.9 are combined to form the new overload reduction and ratio table, table 4.10.

**TABLE 4.10**

AN UPDATED COMPOSITE OF TABLES
4.5, 4.7, 4.8 AND 4.9

<table>
<thead>
<tr>
<th>Delay Index (d)</th>
<th>Type Delay (G or A)</th>
<th>Flight Definition</th>
<th>Overload Reduction (I_d)</th>
<th>Delay Cost (K_d)</th>
<th>Ratio of Reduction to Cost (F_d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>G</td>
<td>(1,2)→(1,3)</td>
<td>0</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>(1,2)→(1,3)</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>G</td>
<td>(1,1)→(1,3)</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>A</td>
<td>(1,1)→(1,3)</td>
<td>0</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>G</td>
<td>(1,1)→(2,2)</td>
<td>-1</td>
<td>6</td>
<td>-1/6</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
<td>(1,1)→(2,2)</td>
<td>-1</td>
<td>9</td>
<td>-1/9</td>
</tr>
<tr>
<td>7</td>
<td>G</td>
<td>(1,2)→(2,3)</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>A</td>
<td>(1,2)→(2,3)</td>
<td>1</td>
<td>7</td>
<td>1/7</td>
</tr>
<tr>
<td>9</td>
<td>G</td>
<td>(2,1)→(1,2)</td>
<td>-2</td>
<td>4</td>
<td>-2/4</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
<td>(2,1)→(1,2)</td>
<td>-1</td>
<td>6</td>
<td>-1/4</td>
</tr>
<tr>
<td>11</td>
<td>G</td>
<td>(2,1)→(2,3)</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>A</td>
<td>(2,1)→(2,3)</td>
<td>1</td>
<td>8</td>
<td>1/8</td>
</tr>
<tr>
<td>13</td>
<td>G</td>
<td>(2,2)→(1,3)</td>
<td>-1</td>
<td>5</td>
<td>-1/5</td>
</tr>
<tr>
<td>14</td>
<td>A</td>
<td>(2,2)→(1,3)</td>
<td>0</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>G</td>
<td>(2,3)→(2,4)</td>
<td>1</td>
<td>8</td>
<td>1/8</td>
</tr>
<tr>
<td>16</td>
<td>A</td>
<td>(2,3)→(2,4)</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

The procedure now requires the selection of the delay with the highest ratio (see table 4.10) for entry into the solution. That selection is delay index 8 with a ratio of 1/7 which is an air delay of flight (1,2)→(2,3) at a cost
of seven units. Recall that up to this point only one overload existed and since delay index 8 reduces overloads by one unit (i.e., I_8=1) it can be concluded that the addition of this delay to the previous delay produces a complete or feasible delay plan. The feasible solution and its value are

Ground Delay (1,1)→(1,2) at a cost of 4 units
Air Delay (1,2)→(2,3) at a cost of 7 units

For a total cost of 11 units

This feasible solution may or may not be the optimum. However, the value eleven combined with the lower bound nine narrows the search to finding solutions having values between nine and eleven units.

The next procedural step is to examine table 4.10 for a less costly solution containing the ground delay of (1,1)→(1,2). But since the only other candidates having positive ratios are delay indices 12 and 15 and their delay costs are eight units each, these candidates are ruled out at this step and no further search of table 4.10 is required.

4.6 The Backtracking Procedure

At this point, all solutions containing the ground delay (1,1)→(1,2) have been considered and the least costly such feasible solution has been found with a value of eleven units of cost. The procedure required the explicit enumeration of one solution and implicitly
enumerated all others containing a ground delay of (1,1)\rightarrow(1,2). One of the implicitly enumerated solutions is as follows:

Ground Delay (1,1)\rightarrow(1,2) at a cost of 4 units
Air Delay (2,2)\rightarrow(1,3) at a cost of 5 units
Ground Delay (2,3)\rightarrow(2,4) at a cost of 8 units

For a total cost of 17 units

If it is assumed each flight can be delayed a maximum of two periods, the formula for the number of delay combinations, K, is from equation (3.8).

\[ K = 6^N \]

where \( N \) = number of flights

But since flight index 1 has been delayed once, there exist only three possibilities for further delay of that flight as discussed in section 3.4. Hence, the number of combinations implicitly enumerated (as opposed to explicitly enumerated) is

\[ K = (3)6^{N-1}-1 \]

and for \( N = 8 \), the value of \( K \) is 839,807 of which many are infeasible and all are nonoptimal solutions. The minus one is for the one solution explicitly stated. The N-1 term is used to eliminate flight index 1 from the two delay class and the 3 refers to flight index 1 now being in the one delay class.

4.7 Finding Optimality and Completing Search

To further search for the optimal feasible solution or to confirm that the feasible solution with a value of
eleven is optimal requires backtracking. Backtracking simply refers to deleting the last delay entered in the solution which, in the example problem, is a ground delay of (1,1)→(1,2) to continue the search for the optimum.

The computer program developed to carry out the algorithm "backtracks" by using the facts shown in table 4.10 and the delay used in arriving at table 4.10 (i.e., ground delay of (1,1)→(1,2) to arrive at table 4.5. This is necessary to avoid the repeated storage of tables similar to tables 4.5 and 4.10. For large problems of 3500 flight operations, it may be necessary to construct several hundred such large tables and storage of these tables for large problems is impractical. Therefore, backtracking, the reverse of construction, is utilized to move backwards to a previous table. The same time-saving features for constructing one table from another (see section 4.5) are employed in the backtracking process.

In the sample problem the backtracking from table 4.10 simply yields table 4.5 with one exception. That exception is that a ground delay of flight (1,1)→(1,2) is no longer considered since all solutions including this delay have been enumerated.

Examining table 4.5 and ignoring delay index 1 allows for the computation of a new lower bound. Since two overloads exist at this point (see section 4.2), the lower bound calculation (see section 4.4 and figure 4.2)
yields a bound of ten. This was arrived at using delay indices 3 and 7 with delay costs of five and five and overload reductions of one and one, respectively.

Since a lower bound of ten is less than the least costly solution obtained thus far, it is necessary to continue searching using the construction process. From table 4.5 select flight index 3 since it is one of the highest ratio delays not yet considered. The delay is a ground delay of \((1,1)\rightarrow(1,3)\) at a cost of five units.

To form the next tables, the process outlined in section 4.5 is followed and tables 4.11 and 4.12 result. From table 4.12 the maximum ratio is observed for delay index 7 at \(1/5\). This delay and the previous delay (i.e., ground delay \((1,1)\rightarrow(1,3)\)) combine to reduce two overloads and, therefore, a complete solution is attained. The value of this solution is ten (i.e., five units for each delay) which is a less costly solution than the solution previously found at eleven units cost and the new upper bound is ten. The new solution, therefore, unseats the incumbent since it provides a lower cost solution value. The new solution is

\[
\begin{align*}
\text{Ground Delay } (1,1)\rightarrow(1,3) & \text{ at a cost of } 5 \text{ units} \\
\text{Ground Delay } (1,2)\rightarrow(2,3) & \text{ at a cost of } 5 \text{ units} \\
\text{For a total of } 10 \text{ units}
\end{align*}
\]

Further examination of table 4.12 shows \(1/7\) to be the next largest ratio (see delay index 8). The cost of this delay
TABLE 4.11
NEW PROJECTED TERMINAL-PERIOD STATES INCORPORATING
A GROUND DELAY OF (1,1)→(1,3)

<table>
<thead>
<tr>
<th>Terminal</th>
<th>Period</th>
<th>Previous Projected State</th>
<th>New Projected State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>O₁,₁ = 1</td>
<td>E₁,₁ = O₁,₁ + 1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>E₁,₂</td>
<td>E₁,₂</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>U₁,₃ = 1</td>
<td>U₁,₃ = U₁,₃ + 1</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>U₁,₄ = 1</td>
<td>E₁,₄ = U₁,₄ + 1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>E₂,₁</td>
<td>E₂,₁</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>E₂,₂</td>
<td>E₂,₂</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>O₂,₃ = 1</td>
<td>O₂,₃ = 1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>E₂,₄</td>
<td>E₂,₄</td>
</tr>
</tbody>
</table>

is seven units yielding a total of twelve units (7+5) which exceeds the present solution value of ten. Thus, all other solutions containing a ground delay of (1,1)→(1,3) have been implicitly or explicitly enumerated and are either infeasible or nonoptimal.

It is now appropriate to decompose back to table 4.5 again. It is necessary to eliminate both ground delays of (1,1)→(1,2) and (1,1)→(1,3) from further consideration. Referring to table 4.5, a new lower bound (not including the enumerated delays) is calculated to be twelve (i.e., delay indices 7 and 8 with costs of five
### TABLE 4.12
OVERLOAD REDUCTION AND RATIO CALCULATIONS FOR EXAMPLE PROBLEM FOLLOWING A GROUND DELAY OF (1,1)→(1,3)

<table>
<thead>
<tr>
<th>Delay Index \ Delay Type</th>
<th>Flight Definition</th>
<th>Overload Reduction</th>
<th>Delay Reduction Cost</th>
<th>Ratio Reduction Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>(d) \ (G or A)</td>
<td>(V (K_d) (P_d)</td>
<td>_</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>2 A (1,1)→(1,2)</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3 G (1,2)→(1,4)</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4 A (1,2)→(1,4)</td>
<td>0</td>
<td>16</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5 G (1,1)→(2,2)</td>
<td>-2</td>
<td>6</td>
<td>-2/6</td>
<td></td>
</tr>
<tr>
<td>6 A (1,1)→(2,2)</td>
<td>-1</td>
<td>9</td>
<td>-1/9</td>
<td></td>
</tr>
<tr>
<td>7 G (1,2)→(2,3)</td>
<td>1</td>
<td>5</td>
<td>1/5</td>
<td></td>
</tr>
<tr>
<td>8 A (1,2)→(2,3)</td>
<td>1</td>
<td>7</td>
<td>1/7</td>
<td></td>
</tr>
<tr>
<td>9 G (2,1)→(1,2)</td>
<td>-1</td>
<td>4</td>
<td>-1/4</td>
<td></td>
</tr>
<tr>
<td>10 A (2,1)→(1,2)</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11 G (2,1)→(2,3)</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>12 A (2,1)→(2,3)</td>
<td>1</td>
<td>8</td>
<td>1/8</td>
<td></td>
</tr>
<tr>
<td>13 G (2,2)→(1,3)</td>
<td>-2</td>
<td>5</td>
<td>-2/5</td>
<td></td>
</tr>
<tr>
<td>14 A (2,2)→(1,3)</td>
<td>-1</td>
<td>9</td>
<td>-1/9</td>
<td></td>
</tr>
<tr>
<td>15 G (2,3)→(2,4)</td>
<td>1</td>
<td>8</td>
<td>1/8</td>
<td></td>
</tr>
<tr>
<td>16 A (2,3)→(2,4)</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

and seven, respectively). This means that all solutions not including (1,1)→(1,3) and/or (1,1)→(1,2) will generate at best a solution with a value of twelve. Thus, the optimal solution value is ten and further search is unnecessary. A network showing the optimal solution is provided in figure 4.4. The dashed arcs indicate the flights which are delayed. The optimal solution is then,

Ground Delay (1,1)→(1,3) to (1,2)→(1,4)
Ground Delay (1,2)→(2,3) to (1,3)→(2,4)

Total solution cost is 10 units.
The search process is shown in figure 4.5 as a tree diagram. The nodes represent a delay and the contents of the nodes indicate the delay index used in table 4.5. For each node representing a partial solution a lower bound $LB$ is shown and for each node representing a complete or feasible solution the value of that solution, $Z$, is shown.

In the next section, a general solution procedure is introduced.

**Fig. 4.4** Optimal solution to the example problem.

**Fig. 4.5** Tree considered in arriving at optimality for the example problem.
4.8 A General Algorithm

A general solution procedure to problems falling in the class defined in section 3.1 is described in this section with the aid of a flow diagram. Before examining the details of the algorithm, it is necessary to adopt a complete symbol system to aid in flow charting.

Let:

- \( P \) = the partial solution vector of delay indices currently under consideration
- \( Z \) = the least value of the objective function so far; i.e., the incumbent best solution which is hoped to be improved upon
- \( \bar{X} \) = the complete solution associated with \( \bar{Z} \)
- \( M \) = set of all delay indices
- \( m \) = delay index
- \( n \) = the delay indices currently specified in \( P \)
- \( B^P \) = lower bound on completing \( P \)
- \( Z \) = value of current (partial or feasible) solution
- \( K_m \) = cost of delay \( m \)
- \( I_m \) = overload reductions for delay \( m \)
- \( F_m \) = reduction to cost ratio for delay \( m \)
- \( L \) = number of delays in \( P \)
- \( Q \) = number of projected overloads in problem as computed by methods of sections 3.1.3 and 4.2
Fig. 4.6. Flow diagram of general algorithm to problems in the class defined in section 3.1.
\( Q^P \) = number of overloads remaining for solution P
\( N^L \) = set of delays which have been total enumerated as the Lth entry in P.

The flow diagram is shown in figure 4.6. The steps in solving the example problem in sections 4.2 through 4.7 can be followed through using figure 4.6.

The next section discusses the incorporation of two extensions into the solution procedure.

4.9 Incorporation of Two Extensions Into the Algorithm

Two extensions in addition to the multiple delay of the same flight were discussed in sections 3.5.1 to 3.5.3. The cancelation of a flight (if a feasible alternative) is discussed in section 3.5.2, and the use of equipment for multiple legs or flights extension is discussed in section 3.5.3. Neither of these extensions were considered in the algorithm defined in section 4.8. These extensions are incorporated into the algorithm in sections 4.9.1 and 4.9.2.

4.9.1 Flight Cancelation Extension

Recall from section 3.4 that in previous discussions a flight is considered to be in one of the following conditions:

i) Air Delay

ii) Ground Delay
iii) No Delay

Recall also that a cost is associated with each of these conditions.

In this section, a fourth condition, cancelation, is added to the list of possible conditions. For the case of one delay or cancelation per flight this addition expands the number of possible combinations, $K$ for $N$ flights from $3$ [see (3.7)] to

$$K=4^N.$$  \hspace{1cm} (4.4)

For the case of two delays per flight, the formula is

$$K=7^N.$$  \hspace{1cm} (4.5)

The structure of the tree diagram shown in figure 3.6 requires modification to include the option of canceling a flight. Figure 4.7(a) shows a tree diagram for a one-flight problem allowing a maximum of one delay and no cancelation, and figure 4.7(b) shows the same situation except with cancelation permitted.

Throughout this discussion, it is assumed that the cost of a cancelation of a flight exceeds the maximum cost of a flight delay. In this manner, the algorithm is forced to treat a cancelation as a less desirable alternative to a delay.

To incorporate this extension into the algorithm requires the following modifications and additions:

i) A method for computing overload reduction for a flight cancelation.

ii) A means of incorporating a cancelation in the computation of a lower bound.
iii) A method for decreasing the cancelation cost of a flight each time the flight is scheduled for a delay.

iv) A means for including the cancelation option in the search for optimality.

Each of these topics is considered separately.

Fig. 4.7 A comparison of tree structures for a one-flight problem with and without cancelation.

(a) A one-flight problem without cancelation

(b) A one-flight problem with cancelation

4.9.1.1 Overload reduction for flight cancelations

The overload reduction table for a flight cancelation is similar to those for a ground or air delay as shown in tables 4.3 and 4.4. Table 4.13 using the definitions for 0, E and U introduced earlier shows overload reduction as a function of the terminal-period states and the flight origin and destination [i.e., (i,t)+(i',t+k)]. Note that using a cancelation can never result in increased overloads.
(i.e., no negative overload reductions). This is due to the obvious fact that demand is never increased at a terminal-period. Or stated differently, a flight cancellation only relieves the demand for departures and landings and does not transfer the demand to a later period, as in the case of a delay.

From table 4.13 it is also observed that a cancellation can reduce overloads by two and is thus competitive with a ground delay in this respect. However, a ground delay cost is less than a cancellation cost making the ratio of reduction to cost [see (4.2)] favor the ground delay.

**TABLE 4.13**

OVERLOAD REDUCTION FOR A CANCELLATION OF A FLIGHT $(i,t) \rightarrow (i',t+k)$ AS A FUNCTION OF THE RELEVANT TERMINAL-PERIODS

<table>
<thead>
<tr>
<th>Case</th>
<th>State of $(i,t)$</th>
<th>State of $(i',t+k)$</th>
<th>Overload Reduction $I_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>E or U</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>E or U</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>E or U</td>
<td>E or U</td>
<td>0</td>
</tr>
</tbody>
</table>

Thus, the reduction of overloads and the calculation of a reduction to cost ratio for a flight cancellation can easily be incorporated into the algorithm using the cancellation cost and table 4.13. The cancellation cost must be determined by the $CF^2$ controllers and the airlines. If
a flight should not be considered for cancelation (i.e., delays only) a very large cancelation cost should be entered into the data for the algorithm.

4.9.1.2 Lower bound calculation for flight cancelations

To compute a lower bound to include cancelations requires no modifications of the procedure. It does, however, require the inclusion of cancelation costs, reductions and ratios in the flow diagram of figure 4.2.

4.9.1.3 Cancelation cost decrease for each flight delay

In order to provide an undistorted cancelation cost for a particular flight during the solution procedure, it is necessary to decrease the relative cancelation cost of the flight each time the flight is delayed. The amount of the decrease is the size of the delay cost. Thus, as each delay is incurred, the relative cost of cancelation decreases and thus, the preference for a cancelation is increased. This procedure allows the algorithm to reflect accurately the trade-off of the flight cancelation cost to one or more delays of the flight, thus including these factors in the optimization of the problem.

4.9.1.4 Controlling search with cancelation permitted

Figure 4.7 and equation (4.4) demonstrated that additional combinations must be considered in optimizing
when cancelations are included. To control the search of these additional combinations requires the following step be added to the algorithm:

When a cancelation of a flight is entered into a partial solution that flight is eliminated from further search until the partial solution containing the cancelation is decomposed to a previous partial solution not containing the cancelation.

To clarify this, refer to figure 4.8 which represents a three-flight problem in which the cancelation (use the symbol c for cancelation) of flight 2 is shown as 2c. Since flight 2 is canceled, it is unnecessary to consider delaying flight 2 beyond the node containing 2c (i.e., cancelation of flight 2). However, at node la (i.e., an air delay of flight 1) it is necessary to include the delay of flight 2. This is because at node la the cancelation of flight 2 is not part of the solution.

Fig. 4.8 A three-flight problem illustrating controlled search with the presence of cancelations.
This step is simply inserted each time a cancelation node (see figure 4.8) is incurred in the search. Thus, the algorithm including the modifications stated in sections 4.9.1.1 to 4.9.1.4 is now capable of considering flight cancelations in the optimization procedure.

4.9.2 Multiple Leg Extension

To more accurately model the CF\textsuperscript{2} problem, it becomes necessary to include constraints on equipment utilization. This extension is discussed in section 3.5.3, but is further discussed here for convenience.

The basic idea of a multiple leg is that a single piece of equipment or aircraft is often utilized for more than one leg. A leg is defined as a flight from one terminal-period to another terminal-period. Thus, a multiple leg problem is one in which several terminal-period pairs are linked together to form a chain or journey.

For example, an aircraft may be flown from Chicago to Denver, and then terminated in Los Angeles. In this case, two terminal-period pairs are linked together. The original algorithm and the literature treat each pair independently, ignoring the chaining relationship. Thus, it can occur that the first leg of the journey is delayed, but the second, third, etc., are not. The result is that a solution so generated can prescribe the departure of a
flight prior to the equipment landing. Figure 4.9 shows a multiple leg example.

Fig. 4.9 A multiple leg example.

In this example the equipment is utilized for the following multiple leg journey:

\[(2,1)\rightarrow(1,2)\rightarrow(1,3)\rightarrow(3,5)\]

The dashed arrow represents ground operations such as passenger and baggage loading and unloadings and/or refueling. Thus, if leg \((2,1)\rightarrow(1,2)\) is ground delayed to \((2,2)\rightarrow(1,3)\), it is also necessary to delay \((1,3)\rightarrow(3,5)\) to \((1,4)\rightarrow(3,6)\) due to ground operations.

To include this extension in the algorithm requires the following additions:

i) A method for computing overload reductions for an air delay, ground delay or cancelation of a multiple leg journey.

ii) A method which allows for the delay of a second or third leg without a delay of a prior leg(s).
iii) A means of costing a delay or cancelation of a multiple leg journey and of computing the reduction to cost ratio.

iv) A means for controlling search for the multiple leg situation.

Each of these additions is discussed in detail in the following sections.

4.9.2.1 Computing overload reduction for multiple leg journeys

In the case of the single leg problem tables of overload reductions for flight delays and cancelations as a function of terminal-period states are developed (see tables 4.3, 4.4 and 4.13). For the multiple leg problem, a set of tables becomes extremely complex and is therefore abandoned in favor of a generalized procedure. The general procedure which can also be applied to the single leg case is developed here. The procedure is to determine first the terminal-periods affected by the delay or cancelation. For example, an air delay of a single leg journey of \((i,t)\rightarrow(i',t+k)\) affects only the condition of the destination terminal-period, \((i',t+k)\). A ground delay affects both.

Once the terminal-periods affected by the delay are defined, it is necessary to determine if overload reductions or increases occur. This requires an examination of the state of the affected terminal-period [e.g., \((i,t)]
and the state of the period directly following for the same terminal [e.g., (i,t+1)]. The three conditions which can exist for these two terminal-periods and their resulting overload reductions, increases, or lack of either is shown in table 4.14. This table, unlike the tables just mentioned, can be applied to all delays and simplified to consider cancelations.

**TABLE 4.14**

**GENERAL TABLE OF OVERLOAD CHANGES AS A FUNCTION OF THE STATE OF AFFECT TERMINAL-PERIODS**

<table>
<thead>
<tr>
<th>Case</th>
<th>Projected State of (i,t)</th>
<th>Projected State of (i,t+1)</th>
<th>Overload Reductions or Increases (negative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0, E, or U</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>E or U</td>
<td>U</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>E or U</td>
<td>0 or E</td>
<td>-1</td>
</tr>
</tbody>
</table>

To illustrate the concept and use of table 4.14, an example is considered. Consider a journey involving two legs as shown in figure 4.9. Figure 4.9 used U, E, and O for the three states of terminal periods (i.e., symbols in each node). The flight shown in figure 4.9 is

(2,1)→(1,2)→(1,3)→(3,5)

From this it is found that the (i,t) terminal-periods affected by a ground delay are (2,1), (1,2), (1,3) and (3,5) and the (i,t+1) terminal-periods affected are (2,2), (1,3), (1,4) and (3,6), respectively. Since (1,3) occurs
in both the \((i,t)\) and the \((i,t+1)\) class, the net overload change for \((1,3)\) is zero. The other cases require further consideration using table 4.14. Note that \((2,1)\) an \((i,t)\) node is overloaded and \((2,2)\) an \((i,t+1)\) node is in equilibrium. Table 4.14 suggests (see case 1) a net reduction of one overload. Next, since \((1,2)\) is in equilibrium and \((1,3)\) is underloaded, table 4.14 suggests a zero reduction in overloads. To this point then only one unit of reduction has occurred.

Recall \((1,3)\) had a net demand change of zero, but \((1,4)\) must be considered since a unit of demand is being added at \((1,4)\). Since \((1,3)\) is underloaded and \((1,4)\) is in equilibrium, table 4.14 case 3 applies for an increase in overloads of 1. The last concern is \((3,5)\) which is underloaded and \((3,6)\) which is overloaded. Table 4.14 case 3 again calls for a net increase in overloads [i.e., at \((3,6)\)]. Therefore, the net effect of a ground delay of the entire journey, \((2,1)\rightarrow(1,2)\rightarrow(1,3)\rightarrow(3,5)\), is an increase of one overload (i.e., 1+0-1-1).

An air delay of the same flight is equivalent to an air delay of the first leg and a ground delay of the second leg (see figure 4.9). The \((i,t)\) terminal periods affected in this case are \((1,2)\), \((1,3)\), and \((3,5)\), while the \((i,t+1)\) periods affected are \((1,3)\), \((1,4)\) and \((3,6)\). Applying table 4.14 and similar reasoning shows that an air delay produces a net increase in overloads (i.e., at \((1,4)\)
and (3,6)] of two [i.e., 0-1-1]. Following through the reasoning process provides useful insights.

A cancelation of the flight is less involved since only two cases need be considered instead of the three in table 4.14. Since a cancelation involves only the (i,t) terminal-periods, it is necessary to consider only the states of these terminal periods. Because of this, the only relevant cases are as shown in table 4.15. Using table 4.15 calculates a net overload reduction for a cancelation of the journey shown in figure 4.12 of one [i.e., 1+0+0+0].

TABLE 4.15

<table>
<thead>
<tr>
<th>Case</th>
<th>State of (i,t)</th>
<th>Overload Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>E or U</td>
<td>0</td>
</tr>
</tbody>
</table>

Before turning to costing, it is important to note that second, third and more legs can be delayed without affecting the prior leg(s). Thus, the cancelation and delays of leg two in the example [i.e., (1,3)→(3,5)] must also be evaluated for overload reductions, increases, or neither. The following delays and overload changes for
the example in figure 4.9 are shown in table 4.16. It is interesting to note that only a cancelation of the entire journey produces a net reduction in overloads. This, of course, is not the case in general. In the case of a two leg flight, the range of overload changes are as follows:

i) For Air Delays -3 to 3
ii) For Ground Delays -4 to 4
iii) For Cancelations 0 to 4

For a general M leg flight the ranges of overload changes expand to:

i) For Air Delays -(2M-1) to (2M-1)
ii) For Ground Delays - 2M to 2M
iii) For Cancelations 0 to 2M

**TABLE 4.16**

OVERLOAD CHANGES FOR ALL POSSIBLE DELAYS AND CANCELATIONS OF THE TWO LEG EXAMPLE SHOWN IN FIGURE 4.9

<table>
<thead>
<tr>
<th>Type Delay or Cancelation</th>
<th>Journey</th>
<th>Overload Reductions or Increases (negative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground</td>
<td>(2,1)→(1,2)→→(1,3)→(3,5)</td>
<td>1</td>
</tr>
<tr>
<td>Air</td>
<td>(2,1)→(1,2)→→(1,3)→(3,5)</td>
<td>-2</td>
</tr>
<tr>
<td>Cancel</td>
<td>(2,1)→(1,2)→→(1,3)→(3,5)</td>
<td>1</td>
</tr>
<tr>
<td>Ground</td>
<td>(1,3)→(3,5)</td>
<td>-2</td>
</tr>
<tr>
<td>Air</td>
<td>(1,3)→(3,5)</td>
<td>-1</td>
</tr>
<tr>
<td>Cancel</td>
<td>(1,3)→(3,5)</td>
<td>0</td>
</tr>
</tbody>
</table>
A factor not discussed concerns slack time between legs which can be utilized to prevent a prior leg delay from being carried over to all following legs. For example, if a journey consisted of two legs separated by two time periods of which only one time period is required to perform passenger and baggage operations, then one period of slack time or "catch up" time exists between these two legs. This extra time period can be utilized to "catch up" on the second leg if the first leg is delayed. The net result is that each leg can be considered independently as in the single leg case until all slack time has been utilized. Then it reverts to the multiple leg case. This important practical extension requires only a simple "bookkeeping" procedure to monitor slack or "catch up" time and thereby determine if the single or multiple leg case exists. Obviously, "catch up" time must be another input provided by the airlines.

4.9.2.2 Costing multiple leg delays or cancelations

The costing of single leg delays is discussed in detail in sections 3.2 through 3.2.2. The costs included fuel, maintenance, safety, and passenger-related considerations. In the single leg case, each leg delay had an independent cost, but in the multiple leg case without "catch up" time the cost of a delay of one leg is dependent on the delay costs of another leg(s). One obvious
possibility is, given the cost of delaying each leg simply sum up these costs to get a multiple leg delay cost. This is a viable method if it is recognized that an air delay of a multiple leg journey is equivalent to an air delay of the first leg and ground delays of all following legs. Thus, an air delay of a multiple leg journey is not the sum of the air delays of each leg, but is the sum of the cost of an air delay of the first leg and ground delay costs of all following legs.

The ground delay cost of a multiple leg journey without "catch up" time is simply the sum of the ground delay cost of all legs. That is, it is the sum of the passenger and crew related costs for each leg since this is the predominant cost of ground delays.

The cancelation costs for a multiple leg journey are also additive. That is, a multiple leg cancelation is the sum of the individually leg cancelation costs.

With the calculations of overload reductions in section 4.9.2.1 and costs in this section, it now is possible to compute reduction to cost ratios and, in turn, from this a lower bound is computed using figure 4.2.

4.9.2.3 Controlling search for multiple leg journeys

The methodology of the sections concerned with multiple legs have all been of an accounting or "book-keeping" nature and have not changed the basic algorithm
as defined in figure 4.9. Hence, the controlled search for multiple leg journeys is unchanged except for the additional calculations required at each step of the algorithm.

4.9.3 Notes on Extensions

The required modifications of the algorithm to include the cancelation and multiple leg extension have not been included in the branch-and-bound computer program. However, the extensions have been included in a zero-one programming model and computer program. This model is the topic of the next section and computer results are provided in chapter 5. Also, computational results for several problems are provided in chapter 5 for the branch-and-bound model without the cancelation and multiple leg extensions.

4.10 An Alternative Solution Procedure: Zero-One Programming

The CF^2 problem can also be formulated as a zero-one integer programming problem. This formulation serves two purposes, as follows:

i) Demonstrates an interesting alternative solution formulation, and

ii) Provides a tool (zero-one programming) which is used in section 4.11 to compute an improved lower bound.

Realizing that a cancelation simply cancels demand and a delay shifts demand from one period to another allows the development of constraints on demand shifts. The
simple (not projected) overloads previously calculated are utilized to set limits on demand shifts and form the requirement vector of the constraint set. Thus, a constraint is required for each terminal-period.

The zero-one programming formulation requires the development of a set of vectors representing air delays of each leg, another set of vectors representing ground delays, and another representing cancelations. Associated with each vector is the appropriate delay cost. The decision variables are defined as follows:

\[ x_j = \begin{cases} 
1 & \text{if delay or cancellation } j \text{ is selected} \\
0 & \text{otherwise}
\end{cases} \quad (4.6) \]

for \( j = 1, 2, \ldots, N \)

where \( N \) is the total number of possible delays and cancelations.

The objective function for this formulation is then:

\[ \min f = \sum_{j=1}^{N} d_j x_j \quad (4.7) \]

where \( d_j \) is the cost associated with delay or cancelation \( j \).

\( x_j \) is as defined previously.

\( N \) is the total number of possible delays (both air and ground) and cancelations.

\( f \) is the objective function value.

The first set of constraints is concerned with the delays or the demand shifts required to remove all overloads and is of the form.
\[ \sum_{j=1}^{N} a_{ij} x_j \geq S_i \quad (i=1,2,\ldots,k) \tag{4.8} \]

where \( S_i \) is the simple state of terminal-period \( i \)

(negative valued for underloaded positive valued for overloaded and zero valued for equilibrium).

\[
= v \text{ if delay } j \text{ reduces demand at terminal-period } i \text{ by } v \text{ units.} \\
- v \text{ if delay } j \text{ increases demand at terminal-period } i \text{ by } v \text{ units.} \\
0 \text{ if delay } j \text{ does not alter demand at terminal-periods.}
\]

\( x_j \) is defined previously.

\( k \) is the number of terminal-periods in the problem.

The second set of constraints are required to prevent the simultaneous selection of an air delay, a ground delay, and/or a cancelation for one particular flight. A relatively complex subscript notation is used here to be compatible with a computer program introduced in chapter 5. The constraint set also permits the case of no delay or cancelation.

\[ x_j + x_{j+1} + x_{j+2} \leq 1 \quad \text{for } j=1,4,7,10,\ldots,3n-2 \tag{4.9} \]

where \( j \) is an air delay index number.

\[ p = \left\lceil \frac{(j-1)}{3} + 1 \right\rceil, \text{ the flight index number.} \tag{4.10} \]

\[ x_j = \begin{cases} 
1 \text{ if flight } p \text{ is air delayed} \\
0 \text{ if flight } p \text{ is not air delayed}
\end{cases} \]
\[ x_{j+1} = \begin{cases} 
1 & \text{if flight } p \text{ is ground delayed} \\
0 & \text{if flight } p \text{ is not ground delayed} 
\end{cases} \]

\[ x_{j+2} = \begin{cases} 
1 & \text{if flight } p \text{ is cancelled} \\
0 & \text{if flight } p \text{ is not cancelled} 
\end{cases} \]

and \( n \) = number of flights.

Equations (4.6), (4.7), (4.8), and (4.9) define a zero-one integer programming problem for the case of multiple-leg flights and multiple delays of individual flights. In addition, the model can consider cancelations of flights in deriving the optimum solution. The procedure for solving this zero-one problem is that of Balas (1).

A sample problem is used to illustrate the formulation procedure. Figure 4.10 shows a problem with several overloads. The data for the example is provided in table 4.17 and table 4.18.

The data in table 4.17 include an index number for each flight, a description of the flight, the air and ground delay costs and specifies linkages for the multiple leg problem.

The number three in the equipment link column of table 4.17 means that the same aircraft is used to first
service index five, the leg (2,1)→(1,2), and index three, the leg (1,2)→(2,3).

TABLE 4.17
FLIGHT DATA FOR ZERO-ONE EXAMPLE PROBLEM

<table>
<thead>
<tr>
<th>Leg Index</th>
<th>Flight Data</th>
<th>Single Leg Delay Costs</th>
<th>Cancellation Costs</th>
<th>Equipment Linkage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Ground</td>
<td>Air</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>(1,1)→(2,2)</td>
<td>7</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>(1,1)→(2,3)</td>
<td>9</td>
<td>6</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>(1,2)→(2,3)</td>
<td>8</td>
<td>4</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>(2,1)→(1,1)</td>
<td>9</td>
<td>7</td>
<td>99</td>
</tr>
<tr>
<td>5</td>
<td>(2,1)→(1,2)</td>
<td>10</td>
<td>6</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>(2,1)→(2,3)</td>
<td>7</td>
<td>3</td>
<td>15</td>
</tr>
</tbody>
</table>

Note: Assume only one delay of each flight.

TABLE 4.18
TERMINAL-PERIOD SIMPLE OVERLOAD STATES FOR ZERO-ONE EXAMPLE PROBLEM

<table>
<thead>
<tr>
<th>Terminal Index</th>
<th>Terminal Data</th>
<th>Capacity</th>
<th>Demand</th>
<th>Simple Overload State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1,1)</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>(1,2)</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>(1,3)</td>
<td>2</td>
<td>0</td>
<td>-2</td>
</tr>
<tr>
<td>4</td>
<td>(1,4)</td>
<td>10</td>
<td>0</td>
<td>-10</td>
</tr>
<tr>
<td>5</td>
<td>(2,1)</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>(2,2)</td>
<td>2</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>7</td>
<td>(2,3)</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>(2,4)</td>
<td>10</td>
<td>0</td>
<td>-10</td>
</tr>
</tbody>
</table>
Table 4.13 shows the terminal-period data for the example problem. A terminal-period index is sequentially assigned to correspond to the notation of equation (4.8). The state of each terminal-period is the simple state in which a positive number represents an overload, a negative number represents an underload, and a zero represents equilibrium.

The objective function for this problem is:

\[
\min f = 7x_1 + 5x_2 + 15x_3 + 9x_4 + 6x_5 + 19x_6 + 8x_7 + 4x_8 + 25x_9 + 9x_{10} + 7x_{11} + 99x_{12} + 10x_{13} + 6x_{14} + 17x_{15} + 7x_{16} + 3x_{17} + 15x_{18}.
\]

The constraint for terminal-period index 1 [i.e., (1,1)] is determined by examining table 4.17 for all flight delays or cancelations which affect terminal period (1,1). This includes a ground delay of flight indices 1, 2 and 4, air delays of flight index 4 and cancelations of flight indices 1, 2 and 4. To determine the \(x_j\) subscripts, \(j\) requires equation (4.10) modified as follows:

\[j = 3(p-1)+1\]  \hspace{1cm} (4.11)

So, for flight 1 (p=1) the index \(j\) is 1, for 2 (p=2) the index \(j\) is 4, and for flight 4 (p=4) the index \(j\) is 10. \(S_i\) on the right side of equation (4.8) is shown in table 4.18 as one overload (i.e., +1). The constraint can now be written for (1,1) as

\[
x_2 + x_3 + x_5 + x_6 + x_{10} + x_{11} + x_{12} > 1
\]

The exclusion of \(x_1\) and \(x_4\) is required since air delays of
flight 1 and 2 do not affect terminal-period index 1 [i.e., (1,1)]. A similar analysis produces a constraint for terminal-period index 2 [i.e., (1,2)] as follows:

\[-x_2 - x_5 + x_8 + x_9 - x_{10} - x_{11} + 2x_{13} + 2x_{14} + x_{15} \geq 0\]  \hspace{1cm} (4.13)

The negative coefficients indicate an increase in demand when delays 2, 5, 10 and 11 are implemented. This can be reasoned by noting that a ground delay of \((1,1)\rightarrow(2,2)\) creates increased demand at both \((1,2)\) and \((2,3)\).

The remainder of the constraint set is shown in table 4.19 which uses a tableau format with a cost row, a variable row, and a separate row for each constraint. The elements of the \(x_j\) coefficients are shown in each constraint row and delay vector. The second constraint set which prevents inclusion of more than one of the following:

i) air delay  
ii) ground delay  
iii) cancelation

is also shown in table 4.19.

A special note on the three columns corresponding to leg indices 3 and 5 in table 4.19 is required. The column vectors for leg index 3 constitute the effects of delaying only leg 3, while the columns for leg index 5 includes the effects of delaying both leg indices 3 and 5 because if 5 is delayed, so is 3. Note also the constraint for leg index 5 in table 4.19. This constraint contains six 1's:
TABLE 4.19

TABLEAU FOR ZERO-ONE FORMULATION OF EXAMPLE PROBLEM OF FIGURE 4.10 AND TABLES 4.17 AND 4.18

<table>
<thead>
<tr>
<th>Leg Index</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Costs</td>
<td>7</td>
<td>5</td>
<td>15</td>
<td>9</td>
<td>6</td>
<td>19</td>
</tr>
<tr>
<td>Variable</td>
<td>x₁</td>
<td>x₂</td>
<td>x₃</td>
<td>x₄</td>
<td>x₅</td>
<td>x₆</td>
</tr>
</tbody>
</table>

Terminal Period

1 1 1 1 1 1 1
-1 -1 1 1 -1 -1 2 2 1
-1

CAN BE DELETED: NO NON-ZERO ELEMENTS

1 1 1 1 1 1 1
1 1 1 -1 -1 -1
-1 -1 1 1 1 1 1 1
-1 -1 -1 -1 -1

> 1 (1,1)
≥ 0 (1,2)
≥ -2 (1,3)
≥ -10 (1,4)

> 1 (2,1)
≥ -1 (2,2)
> 1 (2,3)
≥ -10 (2,4)

Leg Index

1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1

1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1

1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1

1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
1 1 1 1 1 1
three for the columns of leg index 3 and three for the columns of index 5. The constraint then says that one of the following cases is possible:

i) air delay leg 3

ii) ground delay leg 3

iii) cancel leg 3

iv) air delay leg 5 and also delay leg 3

v) ground delay leg 5 and also delay leg 3

vi) cancel leg 5 and also cancel leg 3

vii) delay neither leg 5 nor leg 3

The optimal solution to this problem is \( x_8 = x_{11} = 1 \) or ground delays [using equation (4.9)] of legs 3 and 4 [(1,2) + (2,3) and (2,1) + (1,1)].

The value of the solution is eleven (4+7). Chapter 5 provides a complete computer run for another sample problem.

In chapter 2, the work of Eyster is discussed briefly. Mention of a linear programming and heuristic rounding procedure for the unextended problem is made. The model was found to require large amounts of storage as determined by the two equations Eyster developed for computing the number of variables and constraints. The equation for the number of variables is:

\[
6N^2 \cdot KMAX \cdot M + N \cdot M \tag{4.14}
\]

where \( N \) is the number of terminal.

\( KMAX \) is the maximum number of time periods
a flight might travel.

M is the number of time period.

A similar equation for constraints is:

\[ 4N^2 \cdot \text{KMAX} \cdot M + N \cdot M \]  \quad (4.15)

The zero-one formulation developed in this section is dependent on number of flights, number of terminals and number of periods for the unextended problem. The equation to compute the number of variables is:

\[ 2L \]  \quad (4.16)

where L is the number of flights.

The equation for number of constraints is:

\[ 2L + M \cdot N \]  \quad (4.17)

To compare the dimensionality of the two methods, a realistic problem now faced by the CF^2 controllers over a five-hour time horizon is likely to have the following parameters:

- N = 20 terminals
- M = 20 time periods (4 periods/hour and 5 hours)
- KMAX = 10 maximum flight length in time periods.
- L = 2000 flight legs.

Using these values in Eyster's equations (4.14) and (4.15) reveals a problem having 480,400 variables and 320,400 constraints. The parameters for the zero-one formulation yield a dimensionality from equation (4.16) and (4.17) of 4000 variables and 4400 constraints. While this reduction is significant, the zero-one approach
requires a matrix having 17,600,000 elements, indicating a dramatic need for further reductions.

Noting the special structure and sparseness of the tableau in table 4.19 suggests that specialized procedures such as combinatorial programming (i.e., branch-and-bound methods) are more appropriate in solving the complete problem. However, as is discussed in the next section, the zero-one programming method can be integrated into the combinatorial programming framework to compute improved lower bounds.

4.11 Computing Improved Lower Bound Using Zero-One Programming

In the previous section a zero-one programming formulation of the CF² problem is defined. This section examines the use of zero-one programming in combination with branch-and-bound procedures. The zero-one program is used to simply compute a larger lower bound and does not generate a feasible or optimal solution to the CF² problem. A larger lower bound helps reduce the number of solutions which must be explicitly searched.

The methodology of using zero-one programming for lower bound calculation is straight-forward. It requires constraints for only those terminal-periods suffering projected overloads while ignoring underloaded and equilibrium terminal-periods. Recall that lower bounds, not feasible solutions, are the central concern. Thus,
it is appropriate to ignore the underloaded and equilibrium terminal-periods when computing lower bounds. Hence, the formulation in section 4.11 requiring 2L+M·N constraints is reduced substantially to a maximum of M·N constraints. The column vectors are then developed for only those delays which reduce overloads. The $a_{ij}$ are determined as follows:

\[
a_{ij} = \begin{cases} 
1 & \text{if delay } j \text{ reduces demand at terminal-period } i \\
0 & \text{if delay } j \text{ does not alter demand at terminal-period } i \\
-1 & \text{if delay } j \text{ increases demand at terminal-period } i 
\end{cases}
\]

The costs are simply the associated delay costs for each delay $j$ and are to be minimized.

Using the problem introduced in chapter 4 illustrates the procedure (see figure 4.1 and tables 4.1 and 4.2). The initial lower bound calculation requires constraints for terminal-periods (1,1) and (2,3). The complete tableau is shown in table 4.20. For this particular problem, the lower bound is obvious by inspection of table 4.20. The optimal solution, which may or may not be a feasible solution for the CF² problem, is

\[
\begin{align*}
X_2 &= 1 \\
X_8 &= 1 \\
\text{all other } X_i &= 0
\end{align*}
\]

and the lower bound is 9 (i.e., 4+5), which was also computed using figure 4.2.
<table>
<thead>
<tr>
<th>Cost</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>5</th>
<th>8</th>
<th>5</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
<td>(X_2)</td>
<td>(X_4)</td>
<td>(X_6)</td>
<td>(X_7)</td>
<td>(X_8)</td>
<td>(X_{11})</td>
<td>(X_{12})</td>
<td>(X_{16})</td>
</tr>
<tr>
<td>Terminal Period</td>
<td>&gt;1</td>
<td>(1,1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>&gt;1</td>
<td>(2,3)</td>
<td></td>
</tr>
</tbody>
</table>

In the general case, the zero-one program will yield greater lower bounds than those computed by figure 4.2.
5. COMPUTATIONAL ASPECTS: COMPUTER PROGRAMS, COMPUTATION TIMES, IMPLEMENTATION AND UTILIZATION CONSIDERATIONS

In this chapter, the computational aspects of the methods discussed in chapter 4 are examined, such as computer program descriptions, computation times for various types and sizes of problems, and how the programs can be implemented and utilized in the Central Flow Control Facility. In the section on implementation and utilization of the program's specific data processing requirements, man-machine interaction, and how the programs can be used to improve decision-making in the CF, are considered.

The branch-and-bound algorithm and zero-one programming model have both been computerized. A battery of five quite different test problems are solved using the branch-and-bound procedure, and a "bench mark" problem solved by the zero-one programming model is also provided.

5.1 The Branch-and-Bound Computer Program

5.1.1 Program Description

The branch-and-bound algorithm of section 4.8 was programmed in Fortran IV on a time sharing CDC6400 system.
The program is complicated by the use of list processing sort procedures which allow the flights, terminals and time periods to be arranged in convenient hierarchies. The use of conventional sort procedures was considered inappropriate due to the large amounts of data in a realistic CF² problem, and the time-consuming and burdensome resorting required when using conventional sorts. List processing allows sorting and updating of only those elements in the list requiring updating and by doing so saves important computational steps and time. The time-saving is significant in both solution construction (i.e., ratio ranking of flights) and the decomposition process (i.e., backing up to a previous ratio ranking of flights).

The major savings using list processing is time, while the major disadvantage is increased bookkeeping and program complexity. It should be stated that the computer program developed for this research is not "maximally efficient" since, being a prototype, many patches of the original code were required in the evolutionary process. It is believed that a rewrite of the program using the facts and "short-cuts" gained from the prototype development will result in a substantial reduction in computation times. This is typical of many computer-oriented research projects, since a "maximally efficient" computer code is not easily achieved in most complex applications.

The branch-and-bound computer program is designed
to handle up to 25 terminals, 25 periods, and 6500 flight operations (landings and departures) or 3250 flights. For a problem containing the maximum number of flights, the use of secondary disk storage is required and computation time increases. Instead of using secondary disk storage, a procedure which develops an equivalent smaller problem is used. The equivalent problem is arrived at by recognizing and eliminating flights which cannot possibly appear in the optimal solution. This is accomplished by first noting all terminal-periods where projected overloads do not occur, and then eliminating all flights which traverse a route between two such terminal-periods. If a terminal-period which was not overloaded becomes overloaded during the solution construction process, it becomes necessary at that point to input the additional flight data which corresponds to the new overloaded terminal-period. Thus, the procedure is implicitly considering all flights in the optimization.

The largest example problem yet considered contains 3041 flights. The equivalent problem contains only 426 flights, a 75 percent savings in flight data storage and probably computation time. Using internal storage only (100,000 words for CDC6400) the maximum possible problem size is 600 flights, 25 terminals, and 25 time periods. Of course, larger internal storage machines are available to expand this size considerably.
The equivalent problem and the original problem contain the same number of terminals and time periods, 17 and 3 respectively.

5.1.2 Computational Experience

In chapter 4 an example problem having two terminals, five time periods and eight flights is solved manually. This problem and four other problems were solved using the Fortran program. The computation time in central processing unit (CPU) seconds on a CDC6400 for the problems are provided in table 5.1. The number in parentheses for problem 5 is the number of flights in the larger-equivalent problem. The parameters used in table 5.1 to define problem size are the number of terminals, the number of periods and the number of flights. Because most of the calculations deal with flight data, it is believed that the number of flights is the parameter which has the greatest influence on computation time. The appearance in table 5.1 of an almost linear relationship between the number of flights and CPU seconds may be misleading. Generally, efficient combinatorial programming algorithms exhibit the following computational characteristics:

i) Generate a "good" or "near optimal" feasible starting solution in seconds.

ii) The generation time for an optimal solution is almost linearly proportionate to parameters measuring problem size.
iii) In proving optimality of a solution, the computation time begins growing at an increasing rate when compared to the growth of the problem size.

Further computational experience could provide similar results.

**TABLE 5.1**

**COMPUTER RUN TIMES FOR FIVE SAMPLE PROBLEMS**

<table>
<thead>
<tr>
<th>Problem Number</th>
<th>Number of Terminals</th>
<th>Number of Periods</th>
<th>Number of Flights</th>
<th>Time in CPU Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>13</td>
<td>.066</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>.150</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>5</td>
<td>8</td>
<td>.210</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>16</td>
<td>119</td>
<td>9.595</td>
</tr>
<tr>
<td>5</td>
<td>17</td>
<td>4</td>
<td>426 (3041)</td>
<td>28.273</td>
</tr>
</tbody>
</table>

A comparison of computation times between the algorithm proposed here and those found in the literature is possible only in a narrow sense. This is due to several factors as follows:

i) Eyster is the only researcher to provide computation times.

ii) For Eyster's algorithm the computation time is most sensitive to the number of terminals and periods, while the branch-and-bound procedure is most influenced by number of flights.

iii) The computer Eyster utilized is an IBM 360/70 while this research was performed on a slightly slower machine in terms of central processing time, a CDC6400.

iv) The efficiency of the two computer codes are unknown quantities.
v) The problems considered by Eyster are restricted to a maximum of three terminals and twelve time periods due to binding dimensionality and storage requirements. This contrasts to Table 5.1.

With these factors in mind any attempt at comparison is questionable. However, such an attempt is made for reasons of completeness. Table 5.2 shows the computational results of five problems solved using Eyster's (13) linear programming formulation and an ad hoc rounding procedure to attain integer solutions. Table 5.2 does not include the number of flights as a parameter because it does not relate directly to computation time in Eyster's case. Instead a maximum flight length in periods is an influencing parameter. For example, if a flight takes two periods to traverse a path between the origin and destination terminal, then that flight length is two periods. In all of Eyster's examples, the maximum flight length is assumed to be three periods. For the combinatorial programming algorithm the maximum flight length in periods was allowed to vary from zero (i.e., take off and land in the same time period) to the total number of time periods in the problem. Theoretically, then, in problem 4 of table 5.1 (which contains sixteen time periods each representing fifteen minutes) it is possible to include a flight corresponding to sixteen periods. This was not done in problem 4, but a wide mixture of flight lengths from zero periods
to as many as ten periods was considered.

TABLE 5.2
EYSTER'S COMPUTATION TIMES AS A FUNCTION OF THREE PARAMETERS

<table>
<thead>
<tr>
<th>Problem Number</th>
<th>Number of Terminals</th>
<th>Number of Periods</th>
<th>Maximum Flight Length in Periods</th>
<th>Time in CPU Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>47.0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>48.0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>108.0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>175.0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>12</td>
<td>3</td>
<td>583.0</td>
</tr>
</tbody>
</table>

In table 5.2 in problem 1, Eyster considers one time period. This comes about by his assuming a cyclical nature in which a flight departing in time period 1 (perhaps a 24-hour period) arrives back at period 1 (perhaps the next day), usually at another terminal. From a practical viewpoint the cyclic one period case has little validity but for multiple period problems (e.g., 24 periods of one hour each) the cyclical nature more closely approximates reality.

From table 5.2 the largest problem reported by Eyster required 583 seconds. The size of this problem is three terminals and twelve time periods (number of flights not considered relevant by Eyster). The two largest problems solved by the branch-and-bound algorithm are problems 4 and 5 of table 5.1. Problem 4 contained ten
terminals and sixteen time periods, while problem 5 contained seventeen terminals and four time periods. The number of flights for problems 4 and 5 are 119 and 426, respectively. The maximum computation time of 28.273 seconds was observed for problem 5 (see table 5.1). Thus, within the limitations of this comparison the percentage reduction in computation time using the branch-and-bound algorithm is

\[
\frac{583 - 28.273}{583} \times 100\% = 95.15\%.
\]

This is believed to be a conservative estimate because of problem size and central processing unit speed variations.

5.1.3 An Example Computer Run

The problem of most interest is problem 5 because it closely simulates actual size. The terminal demand data for this problem was taken from actual statistical data collected on a Department of Transportation computer (62). The specific data date was randomly chosen to be that of January 7, 1974.

The actual demand for each terminal during the peak periods of 1900, 2000, 2100, 2200 hours is shown in table 5.3. Also shown is a capacity figure for each terminal and period. Note that for period 2200 hours at each terminal, capacity is assumed to be 999. The 2200 hour represents the overflow period used to guarantee that a feasible solution can be generated. Obviously, the
capacity for the 2200 hour is not 999. When a realistic estimate of capacity for the 2200 hour becomes available, then the 2300 hour will become the overflow period and a new optimal solution will be calculated. Recall in chapter 4 that the last period excess capacity assumption is not overly restrictive. This is due to the fact that the algorithm is designed to minimize delay cost and will therefore optimize the use of the last period capacity. Because actual capacity variations are not collected by the Department of Transportation it was necessary to simulate what might have happened due to weather and runway conditions. The three letter terminal codes used in table 5.3 constitute the Federal Aviation Administration's accepted abbreviations for the sixteen terminals listed in section 1.1. An example from table 5.3 is used to clarify the data. Line eleven of the data in table 5.3 corresponds to the CLE or Cleveland terminal at 21 hours or the 9:00 to 9:59 p.m. time frame. The capacity for Cleveland during this hour is assumed to be 55 operations including both landings and departures. For the same period the demand is 58 operations. Thus, since demand exceeds capacity the symbol 0 for overload is shown in column 5 of table 5.3. The number (3) in parenthesis on line 11 represents the number of overloads. If a U is shown in column 5, the terminal is underloaded and the number in parenthesis represents the size of the underload.
## TABLE 5.3
CAPACITY, DEMAND, AND STATE DATA

<table>
<thead>
<tr>
<th>Terminal</th>
<th>Hour</th>
<th>Operations Capacity</th>
<th>Operations Demand</th>
<th>Simple State</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATL</td>
<td>19</td>
<td>88</td>
<td>67</td>
<td>U (21)</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>94</td>
<td>97</td>
<td>O (3)</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>109</td>
<td>100</td>
<td>U (9)</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>999 (Overflow)</td>
<td>84</td>
<td>U (915)</td>
</tr>
<tr>
<td>BOS</td>
<td>19</td>
<td>67</td>
<td>63</td>
<td>U (4)</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>71</td>
<td>70</td>
<td>U (1)</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>71</td>
<td>69</td>
<td>U (2)</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>999</td>
<td>73</td>
<td>U (926)</td>
</tr>
<tr>
<td>CLE</td>
<td>19</td>
<td>48</td>
<td>38</td>
<td>U (10)</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>49</td>
<td>43</td>
<td>U (6)</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>55</td>
<td>58</td>
<td>O (3)</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>999</td>
<td>32</td>
<td>U (967)</td>
</tr>
<tr>
<td>DAL</td>
<td>19</td>
<td>78</td>
<td>63</td>
<td>U (15)</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>78</td>
<td>74</td>
<td>U (4)</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>80</td>
<td>84</td>
<td>O (4)</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>999</td>
<td>79</td>
<td>U (920)</td>
</tr>
<tr>
<td>DEN</td>
<td>19</td>
<td>57</td>
<td>56</td>
<td>U (1)</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>56</td>
<td>52</td>
<td>U (4)</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>56</td>
<td>51</td>
<td>U (5)</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>999</td>
<td>86</td>
<td>U (913)</td>
</tr>
<tr>
<td>DTW</td>
<td>19</td>
<td>56</td>
<td>58</td>
<td>O (2)</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>52</td>
<td>51</td>
<td>U (1)</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>58</td>
<td>54</td>
<td>U (4)</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>999</td>
<td>50</td>
<td>U (949)</td>
</tr>
<tr>
<td>ERW</td>
<td>19</td>
<td>52</td>
<td>48</td>
<td>U (4)</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>52</td>
<td>49</td>
<td>U (3)</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>52</td>
<td>44</td>
<td>U (8)</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>999</td>
<td>35</td>
<td>U (964)</td>
</tr>
<tr>
<td>JFK</td>
<td>19</td>
<td>71</td>
<td>71</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>71</td>
<td>70</td>
<td>U (1)</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>71</td>
<td>75</td>
<td>O (4)</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>999</td>
<td>71</td>
<td>U (928)</td>
</tr>
<tr>
<td>Terminal</td>
<td>Hour</td>
<td>Operations Capacity</td>
<td>Operations Demand</td>
<td>Simple State</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>---------------------</td>
<td>-------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>LAX</td>
<td>19</td>
<td>87</td>
<td>90</td>
<td>O (3)</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>87</td>
<td>76</td>
<td>U (11)</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>67</td>
<td>68</td>
<td>O (1)</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>999</td>
<td>73</td>
<td>U (926)</td>
</tr>
<tr>
<td>LGA</td>
<td>19</td>
<td>55</td>
<td>52</td>
<td>U (3)</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>55</td>
<td>48</td>
<td>U (7)</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>57</td>
<td>61</td>
<td>O (4)</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>999</td>
<td>57</td>
<td>U (942)</td>
</tr>
<tr>
<td>MIA</td>
<td>19</td>
<td>62</td>
<td>69</td>
<td>O (7)</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>65</td>
<td>58</td>
<td>U (7)</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>65</td>
<td>43</td>
<td>U (22)</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>999</td>
<td>22</td>
<td>U (977)</td>
</tr>
<tr>
<td>ORD</td>
<td>19</td>
<td>151</td>
<td>132</td>
<td>U (19)</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>151</td>
<td>145</td>
<td>U (6)</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>151</td>
<td>156</td>
<td>O (5)</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>999</td>
<td>149</td>
<td>U (850)</td>
</tr>
<tr>
<td>PHL</td>
<td>19</td>
<td>58</td>
<td>47</td>
<td>U (11)</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>58</td>
<td>58</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>58</td>
<td>52</td>
<td>U (6)</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>999</td>
<td>54</td>
<td>U (945)</td>
</tr>
<tr>
<td>PIT</td>
<td>19</td>
<td>75</td>
<td>50</td>
<td>U (15)</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>75</td>
<td>79</td>
<td>O (4)</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>75</td>
<td>59</td>
<td>U (16)</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>999</td>
<td>57</td>
<td>U (942)</td>
</tr>
<tr>
<td>STL</td>
<td>19</td>
<td>68</td>
<td>45</td>
<td>U (23)</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>63</td>
<td>56</td>
<td>U (12)</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>68</td>
<td>68</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>999</td>
<td>62</td>
<td>U (937)</td>
</tr>
<tr>
<td>OUTSIDE</td>
<td>19-22</td>
<td>99999</td>
<td>2080</td>
<td>U (99198)</td>
</tr>
</tbody>
</table>

Total Operations Demanded 6082
Total Number of Flights 3041
Total Simple Overloads 40
Total Projected Overloads 41
In the case of equilibrium, the symbol $E$ is placed in column 5 with no number in parenthesis. At the bottom of table 5.3 is a summary of the number of operations, flights, simple overloads and projected overloads. For this problem the number of projected overloads is 41. Recall that a flight delay can reduce overloads by a maximum of two units. Thus, at least 21 delays are required since:

$$\left\lfloor \frac{\text{Number of Overloads} + 1}{\text{Maximum Possible Reduction}} \right\rfloor = \left\lfloor \frac{41+1}{2} \right\rfloor = 21 \quad (5.1)$$

where $\lfloor \cdot \rfloor$ means integer part.

Table 5.3 does not provide flight data. To simulate actual flight data, the Official Airline Guide (67) was used. This guide provides airline travelers with a listing of most commercial flights. Since military and civil flights are not included, it was necessary to randomly generate many of the flights. The demand figures for January 7, 1974 shown in table 5.3 were used as constraints in generating all such flights to insure that total flight operation demand equals that of January 7, 1974. By following this rough simulation procedure, a problem of realistic dimension was generated. Table 5.4 shows the flight schedule which precisely equals the demand of table 5.3. The only other data input concerns delay costs. Due to the size of this problem, it is assumed for convenience that all ground delays of a flight cost one unit.
<table>
<thead>
<tr>
<th>TABLE 5.4 FLIGHT DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ATL</strong></td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>21</td>
</tr>
</tbody>
</table>
and all air delays cost two units.

This assumption is made for two additional reasons:

i) In practice a similar costing procedure will probably be necessary due to the qualitative nature of the delay costs.

ii) Eyster made the same assumption.

It is important to note, however, that the FORTRAN program can accommodate zero or any positive costs without extension.

The effects of the uniform cost assumption on computation time are pro and con. The uniform costs are likely to cause the existence of alternate optima which may aid in quickly finding an optimal solution. However, the bounding procedure is likely to be less effective when alternate optima exist due to the existence of many feasible solutions which may neighbor or equal the optimal solution value. Thus proving optimality in this case may actually consume more time than in the case in which wide variations in delay costs exist. Another way of stating the above is to say that when uniform delay costs are used, the dominance (lower bound) procedures are less effective in saying one branch dominates another.

The algorithm can be used to generate all alternate optima. This can be accomplished by searching for solutions with values less than or equal to the lower bound, rather than strictly less than the lower bound.

It is assumed that a second delay of a flight costs
twice that of the first \( (q=2) \), a third twice that of a second, and so on.

Table 5.4 requires further discussion. Along the top of the table and down the side of the table, the airport terminal abbreviations and time periods are listed. The last column on the right is labelled OUT. This stands for a "dummy terminal" which performs two functions:

i) It is used to denote the destination terminal for all flights departing from one of the sixteen actual terminals during the 19 to 21 hours and landing at one of the sixteen terminals during the 22 hour or beyond. The out terminal then serves as all periods beyond 21.

ii) It is also used to represent other or minor terminals besides the sixteen listed during the 19 hour and beyond. Thus, flights can depart from one of the sixteen major terminals with a destination of a minor terminal, a frequent occurrence in reality.

The last row in table 5.4 represents the IN terminal which has two functions similar to the OUT terminal, as follows:

i) It is used to denote the origin terminal for all flights departing from one of the sixteen terminals before the 1900 hour and landing at one of the sixteen during the 1900 to 2100 hours. The IN terminal serves as all time periods before the 1900 hour.

ii) It is also used to represent all other terminals outside the sixteen listed which have flights scheduled to land at one of the sixteen terminals during the 19 to 21 hours.

Once the columns and rows are understood, the matrix reads similarly to a transportation problem matrix. For
example, the element at the intersection of the row corresponding to PHL (Philadelphia) during the 20 hour and the column corresponding to LGA (New York-LaGuardia) during the 21 hour is a two. This simply means that two flights are scheduled to depart from Philadelphia during the 20 hour (i.e., 8:00 to 8:59 p.m.) and are due to land at LaGuardia during the 21 hour (i.e., 9:00 to 9:59 p.m.). The two flights may have different flight durations. For example, one flight may depart at 8:15 p.m. and arrive at 9:10 p.m. for a 55-minute duration. The other may depart at 8:50 and arrive at 9:31 for a 41-minute duration.

Tables 5.3 and 5.4 and the above-mentioned delay costs represent a complete problem definition. These data were then placed on a disk file via time sharing, and the branch-and-bound computer program was executed. The resulting optimal delay plan is shown in table 5.5 with a total delay cost of 25 units.

The branch-and-bound procedure found the optimal solution on the first pass (i.e., Solution No. 1 in table 5.5) through the construction procedure and then spent most of the 28.273 seconds proving optimality using the bounding process.

For most "practical" problems, the optimal solution should be generated on the first few passes since the
### OPTIMAL DELAY PLAN FOR PROBLEM DEFINED BY TABLES 5.3 AND 5.4

**SOLUTION NO. 1** with value **25** is as follows:

<table>
<thead>
<tr>
<th>DELAY FLIGHT</th>
<th>FROM</th>
<th>TO</th>
<th>AIR=1</th>
<th>GROUND=0</th>
<th>COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>1233</td>
<td>ATLANTA</td>
<td>CLEVELAND</td>
<td>2100</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>123C</td>
<td>ATLANTA</td>
<td>CLEVELAND</td>
<td>2100</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>333C</td>
<td>CLEVELAND</td>
<td>J.F.K.</td>
<td>2100</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>611D</td>
<td>WASHINGTON</td>
<td>DALLAS</td>
<td>2100</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>611F</td>
<td>WASHINGTON</td>
<td>J.F.K.</td>
<td>2100</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>621E</td>
<td>WASHINGTON</td>
<td>NEW YORK LGA</td>
<td>2100</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>933B</td>
<td>J.F.K.</td>
<td>DALLAS</td>
<td>2100</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>101E</td>
<td>LOS ANGELES</td>
<td>DALLAS</td>
<td>2100</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>101M</td>
<td>LOS ANGELES</td>
<td>OHARA-CHICAGO</td>
<td>2100</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>101N</td>
<td>LOS ANGELES</td>
<td>OHARA-CHICAGO</td>
<td>2100</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>121M</td>
<td>MIAMI</td>
<td>LOS ANGELES</td>
<td>2100</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>121N</td>
<td>MIAMI</td>
<td>NEW YORK LGA</td>
<td>2100</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>121O</td>
<td>MIAMI</td>
<td>NEW YORK LGA</td>
<td>2100</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>121P</td>
<td>MIAMI</td>
<td>OHARA-CHICAGO</td>
<td>2100</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>121Q</td>
<td>MIAMI</td>
<td>OHARA-CHICAGO</td>
<td>2100</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1113</td>
<td>ATLANTA</td>
<td>NEW YORK LGA</td>
<td>2100</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2111</td>
<td>BOSTON</td>
<td>NEW YORK LGA</td>
<td>2100</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3113</td>
<td>CLEVELAND</td>
<td>PITSBURG</td>
<td>2000</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>311Z</td>
<td>CLEVELAND</td>
<td>PITSBURG</td>
<td>2000</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>411D</td>
<td>DALLAS</td>
<td>MIAMI</td>
<td>2000</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>411E</td>
<td>DALLAS</td>
<td>MIAMI</td>
<td>2000</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>111M</td>
<td>NEW YORK LGA</td>
<td>PITSBURG</td>
<td>2000</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>111N</td>
<td>NEW YORK LGA</td>
<td>PITSBURG</td>
<td>2000</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>121A</td>
<td>MIAMI</td>
<td>ATLANTA</td>
<td>2000</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**OPTIMAL SOLUTION FOUND**

**STOP**
heuristic ratio calculation directs search towards the "most likely candidate" for optimality. In the next section, the practical benefits of this characteristic are discussed.

Table 5.5 requires further explanation. The column headings provide a reasonable guide to understanding the solution. The DELAY FLIGHT column shows flight numbers for those flights to be delayed. They do not provide meaningful theoretical data but in a practical environment the flight numbers are extremely relevant. The FROM column indicates the scheduled origin terminal and period of the delayed flight. For example, the first flight originates in Atlanta and was scheduled to depart during the 2000 hour. The TO column shows the scheduled destination terminal and period of the delayed flight. For the first flight delay the scheduled landing is in Cleveland during the 2100 hour.

The AIR=1, GROUND=0 column indicates whether the delay is an air or a ground delay. In table 5.5 all the delays are ground and, hence, the column is filled with all zeros. A ground delay of the first flight just discussed results in an altered flight schedule as follows:

<table>
<thead>
<tr>
<th>FROM</th>
<th>TO</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATLANTA</td>
<td>CLEVELAND</td>
</tr>
<tr>
<td>2100</td>
<td>2200</td>
</tr>
</tbody>
</table>

where the hours are updated to reflect a one-period delay.
The COST column simply records the cost of the delay. The sum of the elements in this column is the solutions value which is printed in the first line of table 5.5.

It provides further insight to confirm that the table 5.5 solution is feasible. This requires that the delays of table 5.5 be imposed on tables 5.3 and 5.4. The result in this case is the verification of feasibility.

5.1.4 Implementation and Application Considerations

This section futuristically discusses several practical aspects of using the branch-and-bound program in the CF² facility. In particular data processing considerations, timely results, man-machine interaction and time-saving "short cuts" are examined.

Suppose the data processing time required to set up a CF² problem, including flight data processing and terminal capacity calculation, is large, say an hour. Then little or no advantage is gained by having a computationally efficient algorithm, since the timeliness of a solution is lost, due to the dynamic nature of the problem. Recognizing that collection of large quantities of data and manipulation of these data to a tractable form requires a great deal of computation time, and recognizing
that the CF$^2$ problem requires large quantities of data, leads to the conclusion that methods of bypassing some of the data processing steps must be considered.

5.1.4.1 Using man-machine interaction

The CF$^2$ controllers are highly trained in capacity and demand estimation. In addition, they receive immediate information from terminal controllers concerning overloads. This valuable overload information is not presently collected on computer file, but could easily be fed in using a time sharing terminal. If, in addition to these overload data, a partial list of normal daily flights was kept on file and updated once daily, then the ingredients for a partial data processing bypass exist.

The procedure for the CF$^2$ controller is to simply inform the computer of overloads and then ask the computer, using a time sharing terminal, the following question:

Given the following terminals are overloaded (e.g., ATL, CLE, PIT) during the following time frames (e.g., 9:00-9:30, 10:00-10:45, 10:15-11:10, respectively), what flights do you suggest be delayed?

The algorithm is immediately executed, but skips over the overload calculations since these are provided by the controllers and immediately reads in the available flight data and constructs a first solution which is printed out
as in table 5.5. He may choose to accept this first "good" solution and stop the algorithm or proceed step by step to a final solution. If this procedure is followed, optimality is not guaranteed for two reasons:

i) Not all relevant flight data are considered

ii) And the CF^2 controller may terminate the procedure when he becomes satisfied with the solution results.

However, "good solutions" are generated and the data processing and computation time is minimized by using the controller's knowledge instead of large quantities of data manipulation. The result is a man-machine system which provides both the computation and search power of the computer and the knowledge and experience of the CF^2 controllers. This system could be used to fill the void between the present lack of automation and "complete" automation. It also could provide large savings in data processing equipment and software development now planned by the FAA to cost $11.6 million.

5.2 The Zero-One Computer Program

5.2.1 Program Description

The zero-one programming formulation of section 4.11 presumed the existence of a zero-one computer code.

McMillan (58) provides a definition of a "good solution."
Fortunately, McMillan (59) provides such a code (see FORTRAN IV listing in Appendix). It was necessary to modify the FORTRAN code and implement it on a Control Data CDC6400 in order to obtain computational results. The "benchmark" problem tested contained two terminals, five time periods and eight flights and required 11 seconds of computation time as compared to .21 seconds for the branch-and-bound procedure.

5.2.2 An Example Computer Run

The example problem solved manually in chapter 4 is used as the "benchmark" problem. Figure 4.1 and tables 4.1 and 4.2 completely define the problem.

To formulate the problem as a zero-one program requires the development of a tableau similar to table 4.19. The required tableau is shown in table 5.6.

The results of a photo-reduced computer run are shown in table 5.7. The resulting solution value is ten and the solution calls for \((X_4=1, X_8=1)\) ground delays of

\[(1,1)+ (1,3) \text{ and } (1,2)+ (2,3)\]

which is the same solution computed manually in chapter 4.

Table 5.7 requires some explanation for the columns and rows. The row constraints are in the form

\[
\sum_{j=1}^{16} a_{ij} X_{ij} - S_i > 0 \quad i = 1, 2, \ldots, 17.
\]
TABLE 5.6
TABLEAU FOR ZERO-ONE EXAMPLE PROBLEM OF FIGURE 4.1 AND TABLES 4.1 AND 4.2

<table>
<thead>
<tr>
<th>Flight Index</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Costs</td>
<td>6</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>9</td>
<td>6</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>Variables</td>
<td>$X_1$</td>
<td>$X_2$</td>
<td>$X_3$</td>
<td>$X_4$</td>
<td>$X_5$</td>
<td>$X_6$</td>
<td>$X_7$</td>
<td>$X_8$</td>
</tr>
<tr>
<td>Requirement Vector</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Terminal Period</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Cost</td>
<td>6</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>9</td>
<td>6</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>Requirement Vector</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Terminal Period</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DELETED: NO NON-ZERO ELEMENTS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flight Index</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Costs</td>
<td>6</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>9</td>
<td>6</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>Variables</td>
<td>$X_1$</td>
<td>$X_2$</td>
<td>$X_3$</td>
<td>$X_4$</td>
<td>$X_5$</td>
<td>$X_6$</td>
<td>$X_7$</td>
<td>$X_8$</td>
</tr>
<tr>
<td>Requirement Vector</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Terminal Period</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Cost</td>
<td>6</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>9</td>
<td>6</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>Requirement Vector</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Terminal Period</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DELETED: NO NON-ZERO ELEMENTS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Leg Index</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement Vector</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Terminal Period</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Cost</td>
<td>6</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>9</td>
<td>6</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>Requirement Vector</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Terminal Period</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DELETED: NO NON-ZERO ELEMENTS</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
TABLE 5.7
RESULTS OF ZERO-ONE PROBLEM OF TABLE 5.6

<table>
<thead>
<tr>
<th>CONSTRAINT NUMBER</th>
<th>CONSTRAINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>x(1) x(2) x(3) x(4) x(5) x(6) x(7) x(8) x(9) x(10)</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.000 1.000 0.000 1.000 0.000 1.000 0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>2</td>
<td>1.000 0.000 0.000 -1.000 0.000 -1.000 0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>3</td>
<td>-1.000 -1.000 1.000 1.000 1.000 0.000 0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>4</td>
<td>0.000 0.000 -1.000 -1.000 0.000 0.000 0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>5</td>
<td>0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>6</td>
<td>0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>7</td>
<td>0.000 0.000 0.000 0.000 1.000 1.000 0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>8</td>
<td>0.000 -1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>9</td>
<td>0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>10</td>
<td>1.000 1.000 0.000 -1.000 0.000 1.000 0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>11</td>
<td>0.000 0.000 0.000 0.000 -1.000 -1.000 0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>12</td>
<td>0.000 -1.000 0.000 0.000 1.000 0.000 0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>13</td>
<td>0.000 -1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>14</td>
<td>0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>15</td>
<td>0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>16</td>
<td>0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>17</td>
<td>0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000</td>
</tr>
</tbody>
</table>

SOLUTION FOUND

x(4) = 1
x(5) = 1

ALL OTHER VARIABLES INITIAL 1.000
Hence, table 5.6 required that the elements of the requirement vectors be placed on the left of the inequality size by subtraction. The last column table 5.7 represents the negative of the requirement vector. The last eight constraints of table 5.6 were also multiplied by -1 to reverse the inequality signs.
6. SUMMARY AND CONCLUSION

6.1 Summary of This Research

This research is directed toward the development of a computationally efficient algorithm capable of minimizing delay costs and thus aid the Central Flow Control Facility Controllers in their decision-making tasks. A branch-and-bound algorithm is developed which is capable of considering problems of realistic magnitude. That is, the algorithm is capable of considering the air traffic between 25 terminals over 25 time periods (of arbitrary duration) and up to 3500 flights. The input for the algorithm is the capacity of each terminal during each time period and all flight data defined in terms of origin and destination terminal-periods and air and ground delay costs. The algorithm then recognizes overload situations in which demand exceeds capacity, and prescribes a least costly set of air and ground delays which reduces all overloads.

The basic model considered in the research allowed specification of individual delay costs for each flight. These costs are assumed to reflect fuel consumption characteristics, safety costs, crew delay costs and passenger inconvenience costs. In addition, the model
allows for the case of multiple delays of the same flight with increasing costs for each successive delay.

Two important extensions of the basic model are considered in a revised algorithm. A cancelation cost and the optimal use of flight cancelations is one such extension. The second and most important extension is that of multiple leg journeys in which equipment (i.e., an aircraft) is utilized to provide service over several legs. The extended algorithm is able to consider the effect of a delay of one leg on all following legs and, hence, considers the downstream overload reduction affect on several terminal-periods simultaneously. These extensions must be evaluated computationally and funding is being sought to pursue such evaluation. The FORTRAN programming of the extended algorithm will involve significantly more sophisticated and complicated "bookkeeping" procedures to insure against inflated computation times. Such programming is a high priority concern for future research.

The basic algorithm just mentioned and a zero-one formulation (a benchmark program) have been tested on a CDC6400. It is estimated that the basic algorithm reduces computation time by 95 percent over previous results reported in the literature. This estimation is limited as described in chapter 5. In addition, the basic algorithm expands the magnitude of the problem which can be solved from three terminals and twelve time periods to
25 terminals and 25 time periods (i.e., large enough to consider present $CF^2$ problem magnitude). The relaxation of Eyster's (13) requirement that each flight can be delayed at most once is included in the basic algorithm.

6.2 Recommendations for Further Research

From a practical viewpoint it is important that further computational experience with the algorithm (including the two extensions) be accumulated. Once such experience, on-line testing by $CF^2$ controllers, would be of unlimited value in obtaining a practical solution procedure. Trimming of input requirements as discussed in chapter 5 should be considered during the on-line testing to reduce the magnitude and uncertainty of the data. Particular attention should be focused on how quickly new solutions can be generated as the dynamic situation changes. Any $CF^2$ control mechanism must be capable of adjusting in minutes to dramatic capacity or demand perturbations.

Fine tuning of computer programs is another category very important for practical reasons. The use of various computer languages and special purpose computers should also be considered.

Recognizing that many overloads occur because the airlines schedule large numbers of landings and takeoffs during particular time periods leads to another interesting
research question. Can the algorithm developed here or another algorithm be used to minimize the possibilities of overloads occurring by prescribing more desirable airline schedules? Of major concern in any such research must be equitable allocation of departures and landings to all airlines recognizing customer demand. Unlike the CF2 problem which assumes an initial solution which must be modified to attain feasibility, the above question is directed at prescribing an initial solution which may then require CF2 modification due to perturbation of capacities.

One objective which might be considered in answering this research question is maximization of the minimum amount by which capacity exceeds demand for all terminal-periods. This objective would tend to prevent over-utilization of a terminal during one period and under-utilization during another period. Social costs of people arriving during early morning hours and other off hours must also be considered.

Theoretical research into the extension of the algorithm to a dynamic flow shop or job shop environment would be interesting. Consider the situation in which each terminal represented a machine center and each period a "least common denominator" of time. Then the capacity of a machine center during a period can be determined and entered into the nodes. The demand for work on each
machine center is represented by the arcs as in the CF formulation. Also, delay costs must be entered.

A starting solution is to begin each job as soon as materials permit. But in many cases overloads will occur if this is done, hence, job delays will be required to relieve the overloads. The basic algorithm must be modified to calculate overload reduction for each job delay and must include the multiple leg extension to represent the job sequences through the machine centers.

6.3 Conclusion

The attempt of this research was to bring one step closer to reality the use of man-machine interaction in the determination of least costly solutions to Central Flow Control Facility decision problems. It is hoped that the methods and procedures described here will be modified, improved and eventually evolve into a practical and cost effective solution procedure.
7. LIST OF REFERENCES


21. Federal Aviation Administration. Airport Capacity Criteria Used in Preparing the National Airport Plan, No. AC 150/5060-1A, 8 July 1968.


68. Oliver, R. M. "Delays in Terminal Air Traffic Control." *Journal of Aircraft* 1, No. 3 (May-June 1964): 134-140.


8. APPENDICES
8.1 Alternative Notation

Let $X_{i,t}$ represent the simple state of terminal $i$ during period $t$. Then $X_{i,t}$ is defined as

$$X_{i,t} = D_{i,t} - C_{i,t},$$

where $D_{i,t}$ is the demand for landings and takeoffs at terminal $i$ during period $t$.

and $C_{i,t}$ is the capacity of terminal $i$ during period $t$. A positive $X_{i,t}$ indicates that overloads exist at terminal-period $(i,t)$. If $X_{i,t}$ is zero then terminal-period $(i,t)$ is in equilibrium. Finally if $X_{i,t}$ is negative then terminal-period $(i,t)$ is underloaded.

Computing projected states for each terminal-period is the next major concern. Let $P_{i,t}$ be the projected state of terminal $i$ during period $t$. It is possible to define $P_{i,t}$ as a function of the simple state $X_{i,t}$ and the projected state of terminal-period $(i,t-1)$. The recursive formula which is used to compute projected states is then

$$P_{i,t} = X_{i,t} + f(P_{i,t-1})$$

where $X_{i,t}$ is the simple state of $(i,t)$.

$$f(a) = \begin{cases} 
  0, & \text{if } a \leq 0 \\
  a, & \text{if } a > 0
\end{cases}$$

$P_{i,t}$ is the projected state of $(i,t)$. and $P_{i,0}$ is zero. This formula is applied for each terminal $i$ starting at period $t=1$ and progressing to the final period.
8.1 Computer Program
PROGRAM RANDOM (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT)
DIMENSION A(50,90), JF(50,90), CS(100), XL(90), D(90), E(90,90)
DIMENSION H(100), C(100), BS(100), S(100), SH(100), NS(100), NF(100)
DIMENSION JTEMP(3), JTEMP(3), ATTEMP(3), SMAX(100), SMAXB(100), T(100)
DIMENSION JH(100), XX(100), Y(100), PE(100), K0(5)
INTEGER S, SMAX, MSC, T
COMMON /BLS/ MS(90), ZBAR
DATA HCIB/6MB /
DATA BLANK/6H /
DO 30 I=1, 90
   D(I)=0.0
   JH(I)=0.0
   PE(I)=0.0
   XL(I)=0.0
   XX(I)=0.0
   Y(I)=0.0
   MS(I)=0.0
   E(I)=0.0
   H(I)=0.0
   C(I)=0.0
   HS(I)=0.0
   S(I)=0.0
   SH(I)=BLANK
   NS(I)=0
   NF(I)=0
   SMAX(I)=0
   SMAXB(I)=BLANK
   T(I)=0
   DO 30 J=1, 50
      A(J*I)=0.0
      JF(J*I)=0.0
   30 CONTINUE
   11 = 0
   POST = 1
   INS=5
   IFIRST=1
   NOPT = 0
   READ 870, M, N, L, ISCMAX, ISCFR, ZHAR, ZKAR
   IF (M.EQ.0) GO TO 820
   MO=M
   M1=M0*1
   JSCFR=ISCFR
   ZKBAR=ZKAR+.99999
   IF (L.EQ.0) GO TO 80
   IFIRST=0
   80 READ 880, (C(J), J=1,N)
   READ 880, (M(I), I=1,M)
   50 READ 890, ((ITEMP(K), JTEMP(K), ATTEMP(K)), K=1,3)
   END=0.0
   DO 60 K=1, 3, 1
      KI=ITEMP(K)
      KJ=JTEMP(K)
      IF (KI.EQ.0) GO TO 60
      IF (KJ.EQ.0) GO TO 60
      KJF=NF(KI)+1
      NF(KI)=KJF
      JF(KI,KJF)=KJ
      A(KI,KJ)=ATEMP(K)
      END=1.0
   60 CONTINUE
   IF (END.NE.0.0) GO TO 50
   PRINT 430, (I, I=1,N)
   PRINT 440, (M(I), I=1,N)
   PRINT 450
   DO 70 I=1, M+1
   PRINT 860, ((A(I,J), J=1,N), B(I)
CONTINUE
DO 90 J=1,N+1
CS(J)=C(J)
IF (C(J).GE.0.) GO TO 90
C(J)=-C(J)
DO 100 I=1,M+1
BI(I)=BI(I)+A(I,J)
90 A(I,J)=-A(I,J)
CONTINUE
IF (ZBAR.GT.0.0) GOTO 110
ZBAR=0.0
DO 100 J=1,N+1
100 Z6AP=ZBAR*C(J)
110 ZS=0.0
DO 120 I=1,M+1
120 BS(I)=B(I)
DO 130 J=1,N+1
130 NS(J)=J
IF (M0+ISCMAX.GT.50) ISCMAX=50-M0
11=MO+ISCMAX
DO 140 I=M1,I1+1
NF(I)=N
DO 140 J=1,N+1
JF(I,J)=J
CONTINUE
GO TO 390
JSCF=JSCFR+1
M1=N-L
IF (M1.LE.1) GO TO 550
JSCFW=1
DO 160 J=1,N+1
160 MS(J)=0
IF (L.EQ.0) GO TO 180
DO 170 1=1,1,1
70 MS(J)="( § (  j  j  *
180 CALL SIMPLE (II,N,M0,A,C,B,K0*XL,D,JH,XX,Y,OBJ,E)
11=1 IMPOST
IF (K0(1).EQ.2) GO TO 660
IF (K0(1).EQ.4) GO TO 660
IF (K0(1).EQ.6) GO TO 240
VLPS=-OBJ
IF (VLPS.LE.(-ZBAR)) GO TO 230
DO 190 I=1,N+1
190 IF (D(I).NE.AINT(D(I)).AND.NS(I).NE.0) GO TO 240
CONTINUE
DO 220 J=1,N+1
IF (NS(J).EQ.0) GO TO 220
I=J
L=L+1
NS(J)=0
SH(L)=ACIB
IF (D(I).NE.0.0) GO TO 200
S(L)=-J
GO TO 220
S(L)=J
ZS=ZS+C(J)
DO 210 I=1,M+1
210 BS(I1)=BS(I1)+A(I1,J)
CONTINUE
GO TO 510
K0(1)=6
M0=M0-LT.4)
DO 250 I=1,M0+1
250 BMP1=ZBAR
DO 270 I=1,M0+1
270 BMP1=BMP1+XL(I)*H(I)
IF (ABS(BMP1-BM(I)).LE.0.0005) GO TO 300
IF (M=M0.0.0005) GO TO 270
DO 260 J=1,N,1
260 A(I,J)=A(I,J-1)
M=M-1
270 B(M+1)=AMP1
DO 280 J=1,N,1
ZJH=XX(J)
IF (JH(J)*GE.(*-N)) ZJH=-ZJH
IF (JH(J)*GT.*0) ZJH=0.0
280 A(M+1,J)=ZJH
M=M+1
BS(M)=B(M)
DO 290 K=1,L,1
K1=S(K)
IF (K1.LE.0) GO TO 290
BS(M)=BS(M)+A(M,K1)
290 CONTINUE
300 IF (K0(I).EQ.0) GO TO 660
TD=5
F=Z5
F1=HS(M)
DO 310 J=1,N,1
IF (NS(J).EQ.0) GO TO 310
IF (D(J).LT.TD) GO TO 310
F=F+C(J)
F1=F1+A(M,J)
310 CONTINUE
IF (F.GE.ZBAR) GO TO 550
IF (F1.GE.0.0) GO TO 320
GO TO 550
320 DO 340 J=1,M0,1
F2=HS(I)
DO 330 J=1,N,1
IF (NS(J).EQ.0) GO TO 330
IF (D(J).LT.TD) GO TO 330
F2=F2+A(I,J)
330 CONTINUE
IF (F2.LT.0.0) GO TO 550
340 CONTINUE
IF (M.EQ.M0) GO TO 360
DO 350 I=1,M,1
B(I)=H(I)+F-ZKBAR-ZBAR
350 BS(I)=HS(I)+F-ZKBAR-ZBAR
360 ZBAR=F-ZKBAR
DO 370 J=1,L,1
SMAX(J)=SH(J)
370 SMAX(J)=S(J)
K=L
DO 380 J=1,N,1
IF (NS(J).EQ.0) GO TO 380
K=K+1
SMAX(K)=BLANK
SMAX(K)=J
IF (D(J).LT.TD) SMAX(K)=-J
380 CONTINUE
NOBJ=OBJ
Z0BJ=NOBJ
IF (OBJ.NE.Z0BJ) Z0BJ=Z0BJ+1.0
IF (F.GE.Z0BJ) GO TO 660
GO TO 550
390 J0K=0
400 IF (ZS.GE.ZBAR) GO TO 660
DO 410 I1=1,M0,1
IF (HS(I1).LT.0.0) GO TO 420
410 CONTINUE
GO TO 510
DO 430 J=1,N+1
IF (NS(J).EQ.0) GO TO 430
IF (ZS+C(J).LT.ZBAP) GO TO 430
NS(J)=0
L=L+1
SB(L)=BC1B
S(L)=-J
CONTINUE

KINS=0
IF (IJK.EQ.1) GO TO 450
IF (IJK.EQ.2) GO TO 450
MSC=0
II=I
IJ=I
I2=I
GO TO 450

KINS=0
IF (IJK.EQ.1) GO TO 450
IF (IJK.EQ.2) GO TO 450
MSC=0
II=I
IJ=I
I2=I
GO TO 450

Q=BS(I)
DO 460 J=1,N+1
IF (NS(J).EQ.0) GO TO 460
IF (A(I,J).GT.0.0) Q=Q*A(I,J)
CONTINUE

DO 470 K=1,N+1
IF (NS(J).EQ.0) GO TO 470
IF (A(I,J).GT.0.0) S(L)=-J
S(L)=J
ZS=ZS*C(J)
DO 480 I9=1,M,1
BS(I9)=BS(I9)*A(I9,J)
CONTINUE

KINS=KINS+1
CONTINUE

IF (MSC.EQ.1) GO TO 440
IF (KINS.EQ.1) GO TO 150
IJK=2
GO TO 400

IF (M.EQ.M0) GO TO 530
DO 520 I=M1,M+1
BS(I)=BS(I)+ZS-ZBAP-ZBAR
CONTINUE

ZBAR=ZS-ZKBAR
DO 540 J=1,N+1
SMAX(J)=S(J)
NOPT=NOPT+1
CONTINUE

IF (IFIRST.NE.0) GO TO 580
IFIRST=1
DO 570 J=1,N+1
IF (NS(J).EQ.0) GO TO 570
J1=0
IF (D(J).EQ.1.0) J1=J
IF (D(J).EQ.0.0) J1=-J
IF (J1.EQ.0) GO TO 570
L = L + 1
NS(J) = 0
S(L) = J1
IF (J1 .LT. 0) GO TO 570
ZS = ZS + C(J)
DO 560 I = 1, M, 1
560 HS(I) = BS(I) + A(I, J)
570 CONTINUE
580 IF (JSCFR .NE. 0) GO TO 600
DO 590 J = 1, N, 1
590 IF (NS(J) .GT. 0) GO TO 590
K1 = K1 + 1
T(K1) = J
590 CONTINUE
GOTO 620
600 DO 610 J = 1, N, 1
IF (NS(J) .NE. 0) GO TO 610
K1 = K1 + 1
T(K1) = J
610 CONTINUE
620 IF (K1 .GT. 0) GO TO 640
P = -1.0E10
DO 640 K = 1, K1, 1
J = T(K)
P1 = 0.0
DO 630 I = 1, M, 1
P2 = HS(I) + A(I, J)
IF (P2 .GT. 0.0) GO TO 630
P1 = P1 * P2
630 CONTINUE
IF (P1 .LE. P) GO TO 640
P = P1
J1 = J
640 CONTINUE
NS(J1) = 0
L = L + 1
S(L) = J1
ZS = ZS + C(J1)
DO 650 I = 1, M, 1
650 HS(I) = HS(I) + A(I, J1)
GO TO 690
660 IF (L .EQ. 0) GO TO 690
IF (S(L) .EQ. BLANK) GO TO 780
J = IABS(S(L))
NS(J) = J
S(L) = 0
ZS = ZS + C(J)
DO 670 I = 1, M, 1
670 HS(I) = HS(I) - A(I, J)
680 SB(L) = BLANK
S(L) = 0
L = L - 1
IF (L .LT. 0) GO TO 660
DO 700 J = 1, N, 1
700 S(J) = 0
DO 710 J = 1, N, 1
K = IABS(SMAX(J))
IF (K .EQ. 0) GO TO 720
710 S(K) = 1
720 DO 730 K = 1, N, 1
IF (S(K) .NE. 0) GO TO 730
SMAX(J) = -K
J = J + 1
730 CONTINUE
IF (NOPT.GT.0) GO TO 735  
PRINT 900  
GO TO 20  
735 ZBAR=0.0  
DO 740 J=1,N+1  
K=IABS(SMAX(J))  
IF (CS(K).LT.0.0) SMAX(J)=-SMAX(J)  
IF (SMAX(J).GT.0) ZBAR=ZBAR+CS(K)  
740 CONTINUE  
DO 750 K=1,N+1  
750 T(K)=0.0  
DO 760 K=1,N+1  
K1=IABS(SMAX(K))  
760 IF (SMAX(K).GT.0) T(K1)=K1  
PRINT 920 ZBAR  
DO 770 I=1,N+1  
IF (T(I).EQ.0) GO TO 770  
PRINT 930 I  
770 CONTINUE  
PRINT 910  
GO TO 20  
780 SH(L)=HCIB  
S(L)=-S(L)  
J=IABS(S(L))  
IF (S(L).LT.0.0) GO TO 800  
ZS=ZS-C(J)  
DO 790 I=1,M+1  
790 BS(I)=BS(I)-A(I,J)  
GO TO 340  
800 ZS=ZS+C(J)  
DO 810 I=1,M+1  
810 BS(I)=BS(I)+A(I,J)  
GO TO 340  
820 STOP  
**************************FOR P MAIN PROGRAM**************************  
830 FORMAT(19H10JECTIVE FUNCTION.((//,5X,10(3X,2HX(I3,1H),3X)))  
840 FORMAT(7X,9(F10.3,2X,F10.3//))  
850 FORMAT(11H0CONSTRAINT,49X,12H CONSTRAINTS.//,8H NUMBER)  
860 FORMAT(//,7X,13,10(2X,F10.3),(//,5X,10(2X,F10.3)))  
870 FORMAT(5(2X,I3),2(4X,F6.0))  
880 FORMAT(10X,7F10.0)  
890 FORMAT(13(2X,I3,2X,I3,F10.0))  
900 FORMAT(23H1 NO FEASIBLE SOLUTION )  
910 FORMAT(10X,32H ALL OTHER VARIABLES EQUAL ZERO )  
920 FORMAT(23H1 SOLUTION FOUND .5X,2HZ=.,F15.8,//)  
930 FORMAT(10X,2HX(I3,5H) = 1.//)  
END
SUBROUTINE SIMPLE (INFLAG, MX, NN, A, B, C, K0, KB, P, JH, X, Y, OBJ, E)
REAL B(1), C(1), P(1), X(1), Y(1), OBJ, E(90, 90), A(50, 90), AA, AJT, BB, DT
REAL COST, RCOST, TEXP, TPIV, TY, X0LD, XX, XY, YI, YMAX, EM
INTEGER INFLAG, MX, NN, K0(6), KP(1), JP(1), JA, INVC, IR, ITER, J, JT, K, M
INTEGER KBJ, L, N, JT, NCUT, NUMVR, NVER, NUMPV
LOGICAL TRIG, FIV, FINV, FFRZ, SCH
EQUIVALENCE (XX, LL)
COMMON /BLS/ MS(90), ZBAR
DIMENSION NF(90)
DATA NF*92*0/
FINV = .FALSE.
TRIG = .FALSE.
ITER = 0
LPSEQ = LPSEQ + 1
NUMVR = 0
NUMPV = 0
M = MX
N = NN
TEXP = .5**16
NVER = M/2 + 5
NCUT = 4*M + 10
IF (INFLAG.EQ.0) GO TO 170
FFRZ = .TRUE.
L = 1
20 IF (MS(L).EQ.NF(L)) GO TO 150
IF (MS(L).EQ.NF(L)) GO TO 150
30 IF (MS(L).GT.O .AND. MS(L).EQ.O.AND.X(L).GE.O.) GO TO 1*0
I = L
IF (NF(L).NE.O) GO TO 30
40 J = 1
P(J) = P(J) + E(I, J)
E(I, J) = -E(I, J)
GO TO 450
60 EN = 1
GO TO 400
70 SCH = .FALSE.
IF (COST.GT.0.) GO TO 90
GO TO 450
80 IF (IR.NE.O .OR. SCH) GO TO 110
SCH = .TRUE.
90 EN = EN
DO 100 J = 1, M + 1
Y(J) = -Y(J)
100 CONTINUE
GO TO 450
110 IF ((SCH.AND.ABS(COST).GT.TPIV) .OR. IR.EQ.0) GO TO 630
120 IF (EN.GT.0.) GO TO 490
Y(J) = -Y(J)
GO TO 450
GO TO 490
140 NF(L)=MS(L)
150 IF (JH(L),LT.0) GO TO 160
   IA=JH(L)
   KB(IA)=L
160 L=L+1
   IF (L,LE.M) GO TO 20
   FFRZ=.FALSE.*
   GO TO 560
170 DO 180 J=1,N,1
   KB(J)=0
   CONTINUE
180 FFRZ=.FALSE.*
   DO 190 I=1,M,1
   IF (NF(I),LT.O.OR.(NF(I),EQ.0.AND.B(I),LT.0.))  JH(I)=-I-M
   CONTINUE
190 CONTINUE
200 VER=.TRUE.*
    INVC=0
    NUMVR=NUMVR+1
    TRIG=.FALSE.*
    OBJ=0.
    DO 230 I=1,M,1
    DO 210 J=1,M,1
   E(J,I)=0.
210 CONTINUE
   IF (JH(I),LT.(-M))  GO TO 220
   IF (JH(I),GT.O) JH(I)=0
   P(I)=0.*
   X(I)=B(I)
   CONTINUE
220 E(I,I)=-1.*
   P(I)=1.*
   OBJ=OBJ+B(I)
   X(I)=-M(I)
230 CONTINUE
240 IF (KB(J),EQ.0) GO TO 290
   GO TO 360
250 TY=TPIV
   IR=0
   COST=C(JT)
   DO 260 I=1,H,1
   COST=COST+A(JT,I)*P(I)
   IF (JH(I),NE.0.OR.X(I),LE.0.OR.ABS(Y(I)),LE.TY) GO TO 260
   TY=ABS(Y(I))
   IR=I
260 CONTINUE
   IF (IR,NE.0) GO TO 280
   TY=0.
   DO 270 I=1,M,1
   IF (JH(I),NE.0.OR.X(I),EQ.0.OR.ABS(Y(I)),LE.TPIV)GO TO 270
   IF (ABS(Y(I)),LE.TY*ABS(X(I))) GO TO 270
   TY=ABS(Y(I)/X(I))
   IR=I
270 CONTINUE
280 IF (IR,NE.0) GO TO 500
   FINV=.TRUE.*
   PRINT 640,LPSEQ
   GO TO 170
290 JI=JI+1
   IF (JI,LE.N) GO TO 240
300 VER=.FALSE.*
   DO 310 I=1,M,1
   IF (NF(I),EQ.0.AND.X(I),LT.0.)) X(I)=0.
310 CONTINUE
JT = 0
BB = 0.0
DO 330 J = 1, N, 1
IF (KB(J) .NE. 0) GO TO 330
DT = C(J)
DO 320 I = 1, M, 1
DT = DT + A(J, I) * P(I)
320 CONTINUE
IF (DT .GE. BB) GO TO 330
BB = DT
JT = J
330 CONTINUE
DO 350 I = 1, M, 1
IF (JH(I) .LT. 0) GO TO 350
IF (P(I) .LT. BB) GO TO 340
IF ((1 - P(I)) .GE. BB) GO TO 350
BB = 1 - P(I)
JT = I - M
GO TO 350
340 BB = P(I)
JT = -1
350 CONTINUE
COST = BB
IF (JT .EQ. 0) GO TO 600
IF (ITEP .GE. NCUT) GO TO 590
ITER = ITER + 1
IF (JT .LT. 0) GO TO 400
360 DO 370 I = 1, M
Y(I) = 0.0
370 CONTINUE
DO 390 I = 1, M, 1
AIJT = A(JT, I)
IF (AIJT .EQ. 0.) GO TO 390
DO 380 J = 1, M, 1
Y(J) = Y(J) * AIJT * F(J, I)
380 CONTINUE
390 CONTINUE
GO TO 430
400 JT2 = -JT
EM = 1.
IF (JT2 .LE. M) GO TO 410
JT2 = JT2 - M
EM = -1.
410 DO 420 I = 1, M
Y(I) = EM * E(I, JT2)
420 CONTINUE
430 YMAX = 0.
DO 440 I = 1, M
YMAX = AMAX1(ABS(Y(I)), YMAX)
440 CONTINUE
TPIV = YMAX * TEXP
IF (FFRZ) GO TO 70
IF (VER) GO TO 250
RCOST = YMAX/RB
IF (TRIG .AND. YB .GE. (-TPIV)) GO TO 600
IF (BB .GE. (-TPIV)) GO TO 601
TRIG = .FALSE.
GO TO 602
601 TRIG = .TRUE.
602 CONTINUE
450 AA = TPIV
IR = 0
460 CONTINUE
DO 460 I = 1, M, 1
IF (X(I) .NE. 0 .AND. Y(I) .LE. AA .OR. NF(I) .NE. 0) GO TO 460
AA = Y(I)
IR = I
IF (IR.NE.0) GO TO 480
AA=0.
470 I=1:M
IF (INF(I).NE.0.OR.Y(I).LE.TPIV.OR.Y(I).LE.AA*X(I)) GO TO 470
AA=Y(I)/X(I)
IR=I
470 CONTINUE
480 IF (FFRZ) GO TO 80
IF (IR.EQ.0) GO TO 570
490 IA=JH(IR)
IF (IA.GT.0) KB(IA)=0
500 NUMPV=NUMPV+1
JH(IR)=JT
IF (JT.GT.0) KB(JT)=IR
YI=-Y(IR)
XY=XY(I)
DO 520 J=1,M,1
XY=E(IR,J)/YI
IF (XY.EQ.0.) GO TO 520
P(J)=P(J)*COST*XY
E(IR,J)=0.
DO 510 I=1,M,1
E(I,J)=E(I,J)*XY*Y(I)
510 CONTINUE
520 CONTINUE
XY=X(IR)/YI
DO 530 I=1,M,1
XOLD=X(I)
X(I)=XOLD*XY*Y(I)
530 CONTINUE
YI=-Y(I)
XY=XY(I)
OBJ=OBJ*XY*COST
FR=FR+1
IF (VER) GO TO 290
IF (FR.EQ.2) GO TO 550
X0=X(IR)
X(IR)=X(JT)
X(JT)=XY
DO 540 I=1,M,1
XY=E(IR,I)
E(IR,I)=E(JT,I)
E(JT,I)=XY
540 CONTINUE
IA=JH(JT)
JH(JT)=JT
JH(IR)=IA
550 INVC=INVC+1
IF (FFRZ) GO TO 140
IF (OBJ.GE.ZBAR) GO TO 580
IF (FINV) GO TO 300
560 IF (INVC.GE.NVER) GO TO 200
IF (RCOST.GE.COST) GO TO 600
570 IF (RCOST.LE.(-1000.)) GO TO 600
K=2
GO TO 610
580 K=6
GO TO 610
590 P=INT 650*LPSEQ
GO TO 610
600 K=0
610 J=1,N,1
XX=0.*O
KB=KB(J)
IF (KB.NE.0) XX=X(KB)
KB(J)=LL
620 CONTINUE
K0(1) = K
K0(2) = ITER
K0(3) = INV
K0(4) = NUM
K0(5) = NUM
K0(6) = JT
RETURN

630 PRINT 660, LPSEQ, L, IP, SCH, COST
IF (IR NE 0) GO TO 120
GO TO 170

640 FORMAT (15H0INVERT FAIL LP, I4)
650 FORMAT (3H0ITERATION LIMIT EXCEEDED ON LP, I4)
660 FORMAT (3H0LP, I4, 12H FAIL, SLACK, 13, 4H IR=I3, 5H SCH=L1, 3H C=F19.6)
END