# DESIGN OF AN OPTIMALITY-BASED DECOMPOSITION PROCEDURE FOR AN AIR FREIGHT SCHEDULED PLANNING

A THESIS

Presented to

The Faculty of the Division

of Graduate Studies

By

Charles Brent Hirschman

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Operations Research

Georgia Institute of Technology

June, 1977

# DESIGN OF AN OPTIMALITY-BASED DECOMPOSITION PROCEDURE FOR AN AIR FREIGHT SCHEDULED PLANNING

Approved:

R. L. Rardin, Chairman

R. G. Parker

G. P. Sharp

V. E. Unger

Date approved by Chairman: 4/11/77

## ACKNOWLEDGMENTS

The author wishes to express his sincere appreciation to his thesis advisor, Dr. R. L. Rardin, whose patience, encouragement, and counsel contributed immeasurably toward the completion of this work. The members of the author's reading committee, Dr. R. G. Parker, Dr. G. P. Sharp, and Dr. V. E. Unger are also thanked for their constructive comments and support during this research.

Sincere gratitude is also extended to the author's family for their constant support and encouragement in this endeavor.

Mrs. Sharon Butler receives sincere thanks for her work in typing and preparation of this thesis for final submission.

## TABLE OF CONTENTS

Chapter Page

# LIST OF TABLES

## LIST OF ILLUSTRATIONS

Figure Page

# SUMMARY

This thesis presents an exact mathematical formulation of the mixed integer air freight model. A proposed solution approach is also discussed, but no exact solution is given.

The air freight model includes many features that, even though separate developmental work has been done, when combined, present quite a complex problem. An unlimited fleet size and composition using various type of aircraft is permitted. Routing and scheduling aircraft between the designated cities of the system must also be accomplished. Cargo introduced into the network must be identified by origin-destination designation to assure the correct multi-commodity flow. A time constraint on the amount of time an item has in which to reach its destination location exists also. Freight routing includes the ability to transfer between aircraft at intermediate locations and even to remain in ground storage at such a location where a "best" routing construction results from such action. Finally, aircraft schedules by type of aircraft are required to repeat at predetermined intervals.

The mathematical statement of the model seeks to minimize the cost of aircraft operation and fleet costs plus the freight inventory and transfer costs incurred subject to several constraints. First, the amount of cargo placed

on an aircraft cannot exceed the aircraft's cargo capacity. Second, the number and type of aircraft operating in the system is limited to the size and composition of the fleet available. Third, all the freight entering the system must reach its destination within the stated time limits by utilizing combinations of the aircraft routes to be flown.

An early assumption presented in an effort to outline a solution approach is the fixing of the fleet size and composition. Next, a decomposition procedure utilizing a time window within which all freight movement is feasible is presented. Using this smaller time window, a linear relaxation approach is developed using a column generation scheme. This scheme allows the implicit evaluation of non-basic flight routes and schedules, and freight paths along those routes. However, in order to use such a scheme the costs of the path must be assignable to each of the arcs along the path. Freight transfer costs cannot be so assigned. Thus two cases are developed; one for zero transfer costs and the other for non-zero transfer costs.

Other special characteristics of the air freight model require modifications to the shortest path procedure of implicitly evaluating the non-basic routes and paths. Three algorithms are developed to handle these special conditions. The first identifies the shortest path for freight movement from any node to a group of selected destination nodes where transfer costs are fixed at zero. Second, a flight

path algorithm is described to find the least cost path across the time window. Finally, a special algorithm for non-zero transfer costs is presented.

# CHAPTER I

# INTRODUCTION

In the realistic world of air freight operations the total problem of routing and scheduling of aircraft in order to assure the processing of multicommodity items through the system within critical time constraints has not yet been evaluated from an optimality standpoint. This thesis approaches the overall problem by first formulating the model and then proposing a heuristic algorithm based upon the exact formulation.

## General Problem Statement

Movement of freight using aircraft is a highly competitive market and operational costs can be very critical especially in light of recent fuel price increases. As a result of this competition some companies have been forced to use commercial airlines to carry the majority of the cargo. However a few firms have held firm to the concept of an internal fleet of aircraft. This thesis is concerned entirely with such unimodal independent transportation systems; unimodal in that only aircraft are used without ground augmentation, and independent in the aspect of trying to meet cargo demand requirements only and not passenger demands or timetables. Firms falling into this category are sensitive to area supply and demand requirements, servicing only cities that can generate sufficient cargo movement to justify being included. Usually, only twenty to twenty-five cities or areas are serviced at any one time. In routing aircraft between cities there are no restrictions on the path to be taken such as highways or rail lines. The actual path can be from any one city to any other city under consideration. This results in a large combination of possible routes that a plane can take in fulfilling the shipping requirements. If the fleet were allowed to be as large as needed, then a plane might be obtained for every possible combination. Even though it may be possible to lease an essentially unlimited variety of aircraft, the acquisition cost for such a fleet would be enormous. Thus the problem seeks to minimize the fleet size and associated operating cost for the aircraft. Another of the operating costs that has a direct impact on the routing problem is the fact that landing fees must be paid at each location where the plane lands. The routing must also assure that at specified cycle lengths routes are repeated so that, if not the same plane, then at least one of the same type copies the same route on the same schedule each cycle period.

Scheduling of the routes, once they have been determined, is a critical part of the problem. Departing from many delivery and scheduling schemes, each location can operate as origin, destination and holding point at the same time. The air freight cargo is also permitted to transfer its cargo between planes. The cargo may thus be permitted to wait in inventory at an intermediate point before being reloaded on a plane. Thus new routes for an item of cargo can be created simply by an adjustment of schedules to allow an arriving plane to land and transfer the cargo to a plane waiting to depart. Such scheduling changes cannot be made as freely as desired since the cargo remaining on the plane must also reach its destination.

Emphasis must be placed on the cargo reaching its proper final destination. This is not simply a single commodity network or warehousing problem. When a company comes to an air freight firm with its item, say a motor, it expects that same motor to be delivered at the destination specified. No substitution of another motor or different item is permitted. Thus, the origin-destination aspects of each item forces the model into a multicommodity classification. In actuality, commodities destined for a common location may be considered as a single item originating at various other locations. Once a procedure has determined the commodity routing for this single item from each of the origin points to the common destination, then the specific individual commodity designation can be reassigned.

Cargo within the system will be considered to have uniform density. This standardization of commodity weight and volume characteristics prevents the consideration of any revenue aspects of the system but focuses directly on operational costs. Also items of common origin and common destination are usually grouped together in a unitized cargo pod to facilitate easier handling and transfer of the items while in the system. Aircraft in the fleet carry integer quantities of these cargo pods. The model for ease of conceptualization will consider the cargo commodity values as continuous and not as integer amounts. In addition, commodity flow units will be considered in terms of these cargo pods.

Perhaps of most importance to an outside company utilizing an air freight system is the time it takes to deliver the item in question. High priority is usually attached to every item and over-night delivery required. This tight time restriction places another constraint on the model. The actual physical time requirement may not always be fixed and will generally vary between 6-12 hours. Also, periods of peak demands such as the evening hours before midnight occur while other periods such as the corresponding morning hours are noted for lack of input. These cycles do not affect the model, but the longer cyclical pattern of recurring commodity flows provide a basis for planning the repetitive schedules. These longer patterns repeat themselves approximately every week.

The problem can be seen as an effort to minimize cost subject to limiting constraints with the assurance of an integer solution on all flight schedule variables. Cost includes aircraft variable and fixed acquisition costs, including maintenance, fuel, landing costs, and commodity costs including inventory and transfer costs. Major constraints to be observed are as follows: aircraft capacity limits; cyclic repetition of the routing schedule; minimum ground time requirements for unloading; loading and transfer of cargo; critical nature of individual commodity identification and time in the system.

One of the cost components in flight routing and scheduling that will not be considered in this thesis is flight crew scheduling and positioning. This has been well developed in the literature. The emphasis of this thesis is on commodity movement and flight arrangement to minimize both commodity and flight costs.

## Related Literature

The initial impetus of this thesis was generated by the Air Force's search for a optimality based model to the LOGAIR problem. Fetter and Steorts [15] first developed an approach that begins with known demand and a fixed set of routes and then solves the multicommodity linear program to assign cargoes. Adjustments to the routing system are then made by experienced personnel and the problem is solved again. This cooperative man-machine approach continues iteratively until the "best" solution is achieved. Demmy and Brant [12] describe the development of a routing scheme, but still the man-machine iterative process continues as a vital part in the solution algorithm.

Other published research indicated specialized development in each of the areas composing the air freight problem, but very little progress in the combining of the methods in an optimal solution search.

Commercial air transport problems have concentrated on scheduling and routing aspects of the air freight problem and have achieved much progress in obtaining efficient algorithms. Burger and Rice [7] present an algorithm involving scheduling aircraft over fixed routes. Levin [27] examines both routing and scheduling for individual aircraft over fixed system routes in order to minimize fleet size. Other works, such as Peters [33] examine aircraft rotation and routing for a passenger airline. None of the approaches consider multicommodity or transfer of passengers in the modeling. Etschmaier [13] in his survey of current mathematical programming applications in the realm of commercial air transport system notes that the current approach to designing the system involves first selecting routes, next assigning frequencies, and then scheduling departure times.

Truck routing problems, such as the vehicle dispatch problem by Gillett and Miller [21], and combining of truck trips by Gavish [19] provide some insights into the air freight network. Differences, however, do exist in the following areas: routes are more channelized over fixed physical systems, transfer of cargo is not usually considered or if considered not on a repetitive basis, and scheduling is most often considered separate from routing.

In a similar manner job shop scheduling literature promises much in the type of aircraft scheduling that is desired; but must have a fixed routing scheme as a prerequisite.

Perhaps the best developed model similar to the air freight problem is the tanker scheduling model. The routing and scheduling of tankers over sea lanes is similar to air routing. Initial work was performed by Dantzig and Fulkerson [10]. The model's objective was to minimize the fleet size of a set of identical tankers carrying a single commodity between ports with no transfer of cargo permitted. Bellmore, et al. [2,3,4] through the course of several papers expanded the model to include non-homogeneous vessels with varying costs, a range of acceptable delivery dates instead of required dates, and a fixed charge for putting a tanker into service. A single commodity and lack of the ability to transfer cargo continue to limit its relevance. Applegren [1] continues the development of the model using a multi-commodity formulation but not allowing partial unloadings nor transfer of cargo and restricting the fleet size. Extending the original Dantzig-Fulkerson model, Briskin [6] assumes homogeneous vehicles, known demand, supply, travel times and routes, and then by clustering groups of demand ports permits in effect partial unloadings.

Finally, McKay and Hartley [30] present a generalized problem model including varying capacity vessels and multiple commodity products. Much emphasis is placed on the freedom of routing and scheduling but in the end a "good" solution over a limited range of feasible routings is all that is obtained. The multi-commodity accountability and ability to transfer cargo, once again, escapes inclusion in the model.

Thus, the published literature has not been able to tie all the aspects of the air freight problem together. Principal issues of routing, scheduling and multicommodity flows have been dealt with separately, but they should be unified if an optimality-based procedure is to be developed. One issue never discussed in the literature, in the context of an optimality-based procedure, is the concept of commodity transfer which is one of the distinguishing features of the air freight model.

The remaining portions of this thesis provide a development of the problem and a procedure that unifies the various elements of the air freight problem. First, the model formulation is constructed in individual pieces and then placed together to show how the components are interrelated. Then, a heuristic approach is introduced in an effort to reduce the problem to a manageable size and still retain an optimality basis. Finally, extensions and application of the results are proposed for work beyond the scope of this thesis.

# CHAPTER II

## MODEL FORMULATION

In the formulation of the air freight problem, relevant concepts will be developed individually before the complete model is presented. Reviewing the problem, the objective function calls for the minimization of the fleet acquisition and operating cost, and commodity cost in the system. Limiting constraints enforce the commodity shipment to be less than or equal to the capacity of the aircraft used, the movement of the commodity through the network from its origin to its destination within a limited time, and the development of repetitious routes. There are no limitations on fleet size or composition among various capacity vehicles and transfer of commodities is permitted.

To develop a mathematical programming formulation of this problem, concepts which are needed in the model formulation will be intuitively and logically developed separately. All the concepts will be unified in a complete air freight model stated at the end of the chapter. To facilitate easy reference by the reader, all notation used is summarized in the nomenclature appendix at the back of the thesis. The convention of using capital letters for constant terms and lower case letters for variable terms is adopted throughout.

## Time Space Network

Everything in the air freight model has a time factor associated with it. Planes take off and land at specific times. Commodities enter the system at specified times and must have reached their destination within a time limit. The additional fact that every location can become an origin, destination, and transfer point within the same period of time complicates the picture of the model. If the network was based on space parameters alone, then labelling of the graph would become an unmanageable and confusing task. Thus for the formulation below the network has been expanded to include a node for each city at each discrete epoch in time from 1 through T, the time at which schedules begin to repeat. The cyclical nature of commodities, as was stated before, will be the period length at which schedules and routes will repeat. Cycles may develop within this period among cities, but the entire flight structure must be tied together again at time T to insure the system repeats itself. This system repetition is of a much longer period than the time requirement to move a commodity through the network.

A discrete time epoch approach limits the number of state changes that can occur at any single moment. Two types of arcs, horizontal and diagonal ones, describe the states permitted. Horizontal arcs represent the idea of remaining at the same location over time allowing unloading, loading, transfer and holding of commodities. Horizontal arcs are all of length one and go directly to the next epoch. Diagonal arcs, on the other hand, represent the movement between two different locations over time and the length of the arcs are determined by travel distance. Vertical arcs are not permitted as that would imply travel between locations in zero time. Figure 1 illustrates the structure of the time space network and the type of arcs permitted.

## Commodity Management

Close attention to commodity management is required. As noted in Chapter I, the problem is of a multicommodity nature and not just a multiple transportation problem as the literature has handled. Since a commodity enters the system at a fixed time and location, a specific origin node can be assigned to it. However, a commodity may exit the network at one of several nodes, all of which are at the same location or destination. It is then possible to designate a commodity by its origin node and destination location. Such a unique designation will permit recognition at any point within the network. By inclusion of its origin node value, the length of time the commodity has been in the system can be readily determined and thus the maximum time permitted for the commodity to remain in the system can be enforced. Also, any path from a prospective set of destination nodes can have origin nodes at every node of the path and each commodity can be individually identified. An example can be seen in Figure 1

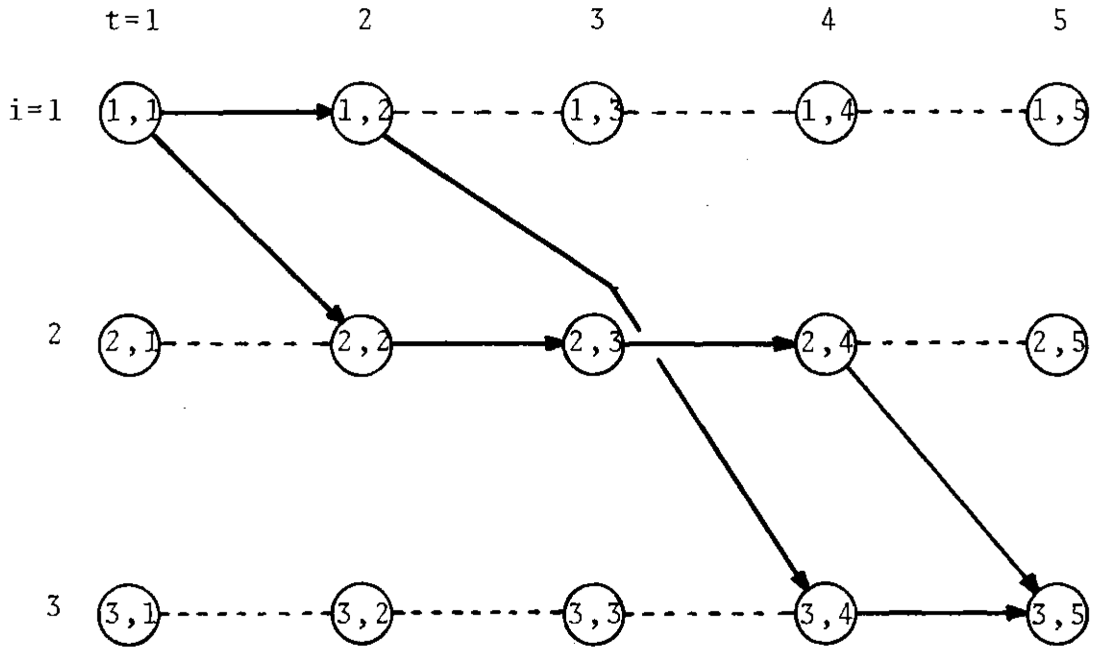


Figure 1. Time Space Network

where if location 3 is the destination node and node (1,1) is the origin, the commodity can be designated as (1,1),3.

The arc from node (1,1) to node (2,2) is a diagonal arc, as is the arc from node (1,2) to node (3,4). The first is one epoch in length while the second requires two epochs to reach its destination. Note that all nodes at the same location, such as from node (1,1) to (1,2) and from node (1,2) to (1,3), have horizontal arcs. The dashed lines indicate connection of the nodes and are distinguished from arcs representing a plane's movement across the network. Although this adaptation increases the number of nodes and the size of the network, the simplicity in manipulation and ease of conceptualization provides for greater benefit. Instead of having only twenty-five nodes and all its overlapping arcs, the model now has twenty-five nodes at every epoch, in other words for a ten epoch network, there would be 250 nodes for the arcs to connect instead of just twenty-five. The network will definitely decrease in its density of arcs using the time space network.

The special structure of the network and associated arcs, flights and paths results in an acyclic network or one without any cycles. This type of network lends itself to very straightforward methods of computing shortest or least cost paths across the network from source to sink. The statement of no cycles is not exactly correct, as at time 1 and time T the routing of flights may form a cycle and all flight arcs at time t and time t + T must be identical. However, a cycle is not the only way to create repetitious routes since a plane of the same type is all that is required.

## Flights and Commodity Paths

Consider the time space network of the air freight model where the nodes and arcs are designated as follows:

(i,t) as the node at location i at epoch t,

(i,t,j,s) as the directed arc from node (i,t) to (j,s) where $t < s$.

A horizontal arc is defined where $i = j$ and $s = t+1$. Defining a diagonal arc is simply where $i \neq j$ and $s = t+L_{ij}$. $L_{ij}$ represents the time distance of the arc from location i to j.

A super source serving as the initiation point of commodities entering the system destined for location u is designated by $SS_u$. Also, the super sink serves as the collection point from a group of destination nodes or sink nodes, u, is designated as $S_u$. The arcs connecting these super nodes with the rest of the network are considered horizontal arcs and designated by ordered triples, $SS_u$,(i,t) and (i,t),$S_u$.

A path within the network is constructed of a sequence of arcs (o,g,d,h), ...,(b,m,u,r) where if the K th arc is (i,t,j,s) then the K+1 arc is (j,s,k,v). Thus, the path is from node (o,g) to (u,r), including the initial arc from the super source, $SS_u$,(o,g) and the final arc to the super sink, (u,r),$S_u$, the path for commodity (o,g),u. A cycle is defined when the same path both starts and ends at the same node within the network. As stated before, this can occur only at one point in the network, that is when schedules and routes repeat at time T.

Both aircraft movement and commodity movement occur along paths in the network. To separate the terminology between the two types of paths, airplane routing through the network is called a plane flight. The term commodity path will be used to describe commodity movement through the network.

## Special Characteristics of Flights

Since we are associating flight with planes then every plane requires a flight path and a flight must extend across the entire network from time 1 to T. Also, a flight can occupy only one location at any epoch. In between airborne travel, a plane must remain at the same location at least one epoch for cargo loading, unloading, refueling, and required maintenance purposes. Thus, horizontal arcs must be included between diagonal arcs of the flight.

Every combination of diagonal and horizontal arcs that constitutes a continuous path across the time space network will be identified as a separate flight. Designate:

$$\Lambda_F = \{(i,t,j,s): (i,t,j,s) \text{ is part of flight f}\}$$

It can be readily seen that the number of flight possibilities is very large since any one arc can be included in several flights. Figure 2 illustrates some possible flight combinations.

Flight 1 consists of the arcs $\Lambda_1 = \{(1,1,1,2),(1,2,3,3),(3,3,3,4)\}$. Another example is flight 2 consisting of $\Lambda_2 = \{(1,1,2,2),(2,2,2,3),(2,3,2,4)\}$.

## Special Characteristics of Paths

The only method of moving a commodity from one location to another is by placing it on a plane and flying it. Thus a commodity path diagonal arc must be associated with a flight diagonal arc. However, at one location across a horizontal arc a commodity may be associated with one plane, remaining on board, or transferred to another plane, or not associated with any plane, simply waiting in holding inventory. In order to transfer a commodity between planes, time is required and it is assumed in this model that only one epoch is needed. The effect of this restriction, as in the case of a flight, is to intersperce horizontal arcs between every diagonal arc in the path of the commodity.

Commodity paths, unlike flights, enter the system at various specified points, and can leave the system at several different time space nodes. Each commodity must reach that destination within a specified time. Early arrival is beneficial, but late arrival is not permitted. A

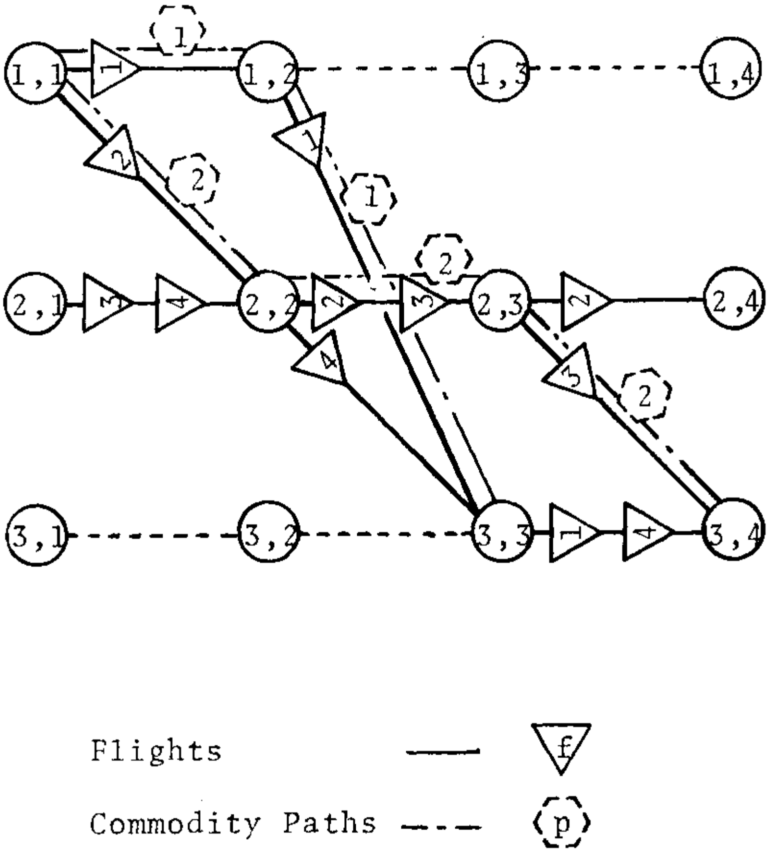


Figure 2. Airplane Flight and Commodity Path Examples

permitted path then, will not exceed the commodity time requirement.

Every permissible combination of diagonal and horizontal arcs between the origin node and destination location that can occur within the commodity time limit in the system is identified as a separate ordered commodity path and designated:

$\Gamma_p$ = {(f,i,t,j,s): if f > 0, path p takes flight f along arc (i,t,j,s); if f = 0, then i = j and path p holds at location i in ground storage.}

Since a single location is represented by several nodes in the network, a set of paths for a single commodity may reach its destination or sink at different points in time. The use of arcs from these different sink nodes u to the super sink $S_u$ for that commodity insures that all flow possibilities are permitted. This special structure creates further complications in solving the problem that will be explained as the model is unified later. Figure 2 illustrates the path structure of a single commodity.

Two paths are shown in the figure for commodity (1,1)3. The first remains on the same flight, $\Gamma_1$ = {(0,1,1,1,2),(1,1,2,3,3)}. The other path transfers from flight

2 to 3 at location 2,

$$\Gamma_2 = \{(2,1,1,2,2),(0,2,2,2,3),(3,2,3,3,4)\}.$$

## Flight Costs

Two types of costs, fixed and variable, comprise the total flight costs. Fixed costs include the fixed charge aspects of plane acquisition, maintenance personnel and space, administrative costs. Of course, fixed cost will vary with different capacity type aircraft, but it is assumed that the fixed cost for the same type of aircraft will remain constant. Variable costs depend upon the actual flight time of the aircraft and the number of stops. Costs prorated over time for in-flight requirements such as fuel are combined with landing costs such as landing fees, ground personnel, and equipment. It is assumed that these variable costs hold constant regardless of quantity of commodity carried, even though in the case of fuel costs this may not be entirely accurate. Diagonal arcs contain the variable cost assignment and are simply the cost per unit time times the length of the arc. Fixed cost charges are assessed either at the first or end of the flight, and horizontal arcs have no cost. Thus, the exact cost of a given flight can be determined as follows:

$$C_f = F_{(a_f)} + \sum_{\substack{(itjs)\varepsilon\lambda_f \\ i\neq j}} [(s-t)K_{(a_f)} + L_{j(a_f)}].$$

In words, this equation takes the fixed acquisition cost for the type of airplane flying this flight and adds the sum of the cost of variable in-flight time costs plus the landing cost at the end of every diagonal arc.

## Commodity Path Cost

Once arcs are established to form feasible paths for a commodity from its source to one of its sink nodes, commodity cost becomes a concern. Of critical interest is the time it takes to arrive at the destination. The inventory cost and transfer cost are the two principal factors. Measuring the charges for processing in and out of the system as well as handling charges over the length of time the commodity is in the system, the inventory cost corresponds to the fixed charge portion of the flight cost. Changes in the inventory cost to occur corresponding to the length of the path or time spent in the system.

$$I_p \equiv \underset{(f,i,t,j,s)\varepsilon\Gamma_p}{\text{Maximum}} \{s\} - \underset{(f,i,t,j,s)\varepsilon\Gamma_p}{\text{Minimum}} \{t\}$$

Transfer costs are quite independent of time and truly represent the most difficult of the charges to deal with in the solution. In the transfer of cargo from one plane to another handling, equipment, and personnel costs are incurred. If this is simply averaged out over all commodity handling costs, then it can be included in the inventory cost above. However, if costs are markedly higher for transfer of cargo and such costs are to be identified and minimized, then the number of such transfers and the locations must be recorded for accurate charging rates. This really implies that one path for a commodity might have a lower inventory cost than another, but that because of transfer cost additions the second path would actually be the better or least cost of the two. Thus, the total unit commodity path cost is the sum of the inventory costs and the total of the transfer costs at each location where a transfer occurs.

$$R_p \equiv I_p + \sum_{\substack{\{(o,i,t,i,t+1)\varepsilon\Gamma_p:\text{ there exists} \\ (f,j,s,i,t)\varepsilon\Gamma_p \text{ with } f > 0\}}} H_i$$

## Model Formulation

A mathematical formulation based upon the concepts previously discussed is now presented.

### Decision Variables

Two decision variables are required in the model. A 0-1 integer variable is used to indicate whether a particular plane flight, f, is used or not.

$$y_f = \begin{cases} 1 \text{ if flight f consisting of the set is used,} \\ 0 \text{ otherwise.} \end{cases}$$

The second decision variable is continuous and is the amount of flow in the commodity path.

$w_p$ = the amount of commodity shipped along path p, where p consists of the set $\Gamma_p$.

## Objective Function

Minimization of the sum of the two cost areas of flights and commodity paths forms the objective function equation

$$\text{Minimize } z = \sum_f C_f \, y_f + \sum_p R_p w_p .$$

## Problem Constraints

Three constraint equations restrict the range of feasible solutions. First of all, the amount of total commodity placed upon a plane cannot exceed its cargo capacity. Also implied is the fact that only actual flights can carry cargo. These constraints apply at every epoch in the time space network.

$$\sum_{\{p:^{P}(f,i,t,j,s)\varepsilon\Gamma_p\}} w_p \le A_a \, y_f \quad \forall f, \forall (i,t,j,s)\varepsilon\Gamma_f \text{ with } i \neq j$$

where $\Lambda_a$ is the capacity of the aircraft of type a.

Second, the fleet size, although unlimited, represents the availability of a plane for a flight. Thus the number of flights must equal the number of planes or the fleet size at each epoch in the network.

$$\sum_{\{f:\ a_f = a;\ (i,t,j,s)\varepsilon\lambda_f \text{ with } t\le r\le s\}} y_f \le N_a \ \forall a;\ \forall r.$$

Third, the sum of flows along the commodity paths connecting the origin node and destination location for a commodity must equal the amount of the commodity to be shipped through the system.

$$\sum_{\{p\varepsilon\Omega_{i,t} \quad (\bigcup_{t<r\le W+t} \Delta_{u,r})\}} w_p = Q_{(i,t)u} \ \forall (i,t);\ \forall u,$$

where $\Omega_{i,t}$ is the set of all commodity paths originating at node (i,t) and $\Delta_{u,r}$ is the set of all commodity paths ending at node u,r and $Q_{(i,t)u}$ is the amount of commodity originating at node (i,t) destined for location u, and W is the time constraint on commodities in the system.

Finally, the routing and scheduling of flights must be repetitive. The same arcs must be repeated again every period length of the cycle.

$$\sum_{\{f:a_f = a;\ (i,t,j,s)\epsilon\lambda_f\}} y_f = \sum_{\{f:a_f = a;\ (i,t+T,j,s+T\epsilon\lambda_f\}} y_f$$

$$\forall a;\ \forall(i,t,j,s)$$

In this form, the model appears quite simple, but actual identification and evaluation of the many flight possibilities that are contained in the scope of all $\Lambda_f$ and in turn the many paths generated in all $\Gamma_p$ present a complex solution procedure, as will be seen in the next chapter.

# CHAPTER III

# MODEL BASED SOLUTION SCHEMES

The model, as developed in the last chapter, seeks to minimize the sum of both commodity path costs and fleet flight costs. One constraint on the model is that cargo capacity between two locations in the system is limited to the capacity of the aircraft flying that arc. Other constraints limit the number of flights at any one time to the number of aircraft in the fleet and require that scheduled routes repeat for similar type aircraft. Finally, all the commodities entering at their origin time-space node must move along permissible paths to reach their appointed destination within a specified length of time.

Proceeding from this basic formulation of the model, solution approaches will be proposed in this chapter. Structural characteristics of this model present opportunities and obstacles requiring special procedural techniques that hopefully do not widely depart from an optimality based solution, but do reduce the problem to a more manageable dimension.

## Fixed Fleet Size and Composition

An initial, reasonable assumption is that the fleet of aircraft available is fixed both as to the total number and composition by various types of aircraft. Perhaps planes could be leased on short notice or purchased on a long term need, but most operations and associated costs work on a fixed fleet basis. Of course, this assumption does not prohibit other fleet size and combination mixtures from being considered. Other options are simply deferred until the problem has been solved under the existing fixed conditions.

Experienced personnel, capital budgeting constraints, and other external sources provide excellent information on fleet requirements and near optimal starting points. Well defined levels of acquisition costs for each additional aircraft and type tend to identify local optimum points rather than a continuous space of feasible solutions. Thus, if fleet composition was entirely variable, a limited number of applications of a solution approach at different levels and mixtures of aircraft in pre-specified regions would likely lead to an adequate solution.

On the other hand, the assumption of a fixed fleet permits better definition of many areas within the model. Once an operational fleet size is fixed, the number of reserve aircraft and maintenance and availability requirements are also fixed. More importantly the number and aircraft mix of all flights across the time space network is fixed so that design of flight routes and commodity paths can be carried out within a defined domain.

## Time Window

Also, by using the fixed fleet assumption, a method of dividing the domain of plane flights and commodity paths into smaller time segments appears via the concept of a time window. Such a concept considers only a limited time span of the entire network at any given point in the optimization. In other words, instead of trying to evaluate an overall network of 250 epochs, the window would look at perhaps only 10 epochs at any one time. The window is optimized locally and then by a predetermined process is reapplied across the entire network.

With the fleet size and composition fixed, much of the problem linkage between time windows is eliminated. The driving constraints of the remaining problem center on the need to move commodities from origin to destination within the allowable time frame. This time frame is much smaller than the cycle period for flights. Thus, the time window concept can be reasonably applied with this relatively short window length. By maintaining feasibility for commodity movement, the time window's local solution hopefully loses very little to an overall optimal solution. However, the small size of the reduced window yields a much simpler and more easily managed problem.

Another interesting facet of the time window is the flexibility in width. It does not have to remain at a fixed number of epochs, but can expand and contract as conditions dictate, as long as it never grows smaller than the time movement constraint limits for commodities. Expansion of the time window can occur in the instance of late morning or early afternoon arrival of items in the system that are not required at their destinations until the next morning. The smallest time window occurs late at night for arrivals that must still be delivered the next morning. Changes in the width of the time window will be controlled from external sources outside the model but will not affect the internal solution procedure.

The management of repeated solution of the time window problem so that a satisfactory feasible solution results for the entire problem can be accomplished in several ways. This thesis adopts a backwards logic method similar to dynamic programming. The initial window application will be at the last time period of the flight cycle, i.e. T. Once the optimal solution for flights and commodity movement within that window have been computed, then the window is moved one time epoch backwards (toward the cycle starting time 1) and resolved. This process is continued one epoch at a time until the leading edge of the window reaches time 1. At this point flights begin to repeat as do commodity flow patterns. The window may be applied on into the previous cycle at time T, to insure satisfactory wraparound characteristics at the ends of the flight period.

By moving the window only one epoch at a time, much of the optimal flight routing and commodity movement determined in the previous window will provide a good feasible starting point for this window. Optimal commodity movement that occurred prior to the edge of the last window, will probably remain optimal within the new window. Similarly, flights will still carry the same cargo across the major portion of the window now that they did before. Additions and changes at the edges of the new window will be required, but the major effort of attaining feasibility has already been established.

Maintenance of feasibility across the same window must remain as an essential factor. Feasibility requires that all commodities regardless of the time or location at which they enter, must reach their respective destinations before the end of the window. Early entering cargo, for example, has many more possible paths to reach its destination than does late entering cargo. For example, a commodity entering the window at the first epoch has more than twice as many feasible paths to reach its destination than does a commodity entering in the middle of the time window having the same destination.

Maintenance of feasibility and the method of moving the time window forces the establishment of intermediate nodes for commodities. That is, once a commodity path is determined and the window moved, then that portion of the path beyond the window boundary is fixed and must be utilized.

Thus, where a commodity path crosses the edge of the window, an intermediate node is established. These intermediate nodes now perform the function of destinations.

An example can be seen in Figure 3. The original commodity destination is at location 3 but the arc of the path that reaches location 3 is outside the window. Node (2,3) is assigned as the intermediate node and the commodity's destination node within the window. The least cost path to node (2,3) is now the objective of the new window, but node (2,3) must be reached in order to place it on arc (2,4,3,5) to its final destination. Note that the next time window edge will occur between epoch 2 and 3 and the intermediate node assigned will be the origin node, thus the commodity need no longer be considered.

Because of the boundary conditions imposed on the commodities a new constraint is imposed. The amount of commodities crossing the edge of the window must be reassigned to the intermediate node which in turn identifies a sink node.

$$\sum_{p\varepsilon\Omega_{(i,t)} \quad \Delta'_{(k,r)}} w_p = q_{(i,t),(k,r)} \; \forall p$$

where $q_{(i,t),(k,r)}$ is the amount of commodity originating at node (i,t) and destined for intermediate node (k,r) and v is the window boundary time and $\Delta'(k,r)$ is the set of all commodity

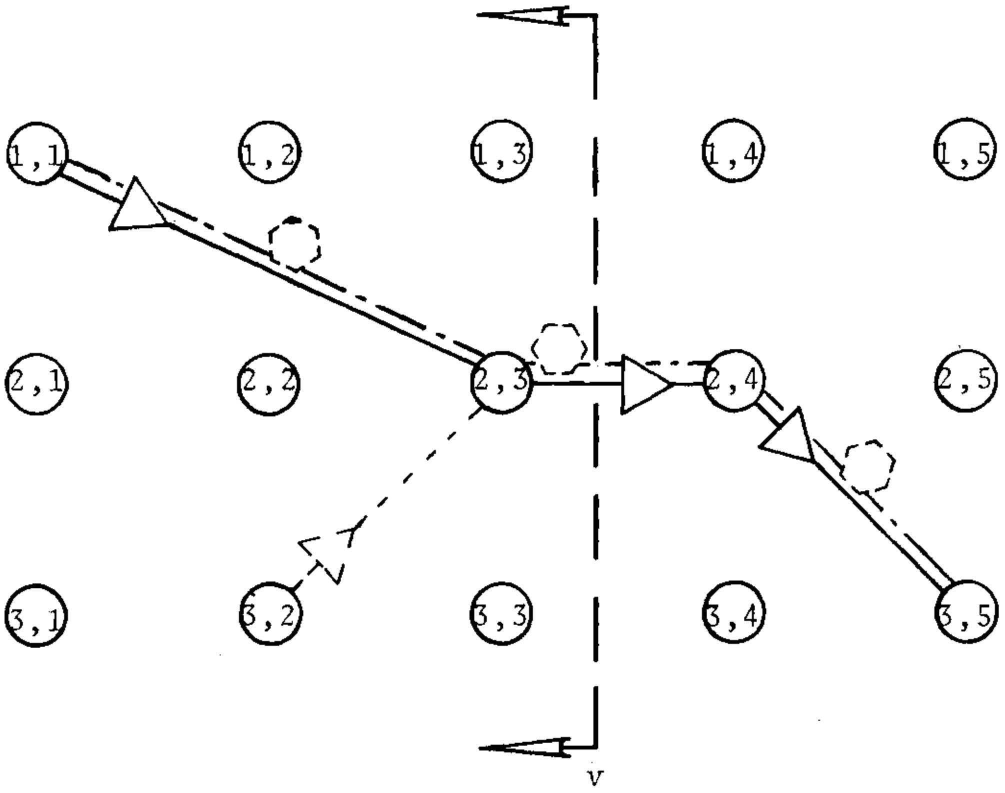


Figure 3. Time Window Boundary Conditions

paths ending at node (k,r) for $r \leq v$, or defined as follows:

$$\Delta'_{(k,r)} \equiv \{p: \ (o,k,r,kr+1)\varepsilon\Gamma_p \text{ and } r = v; \text{ or } (f,k,r+1,j,s)\varepsilon\Gamma_p \text{ with } f\neq o,\ k\neq j, \text{ and } r+1 \leq v < s\}$$

The constraint notes the fact that if the path crosses the edge of the window on a diagonal arc that the commodity must arrive at the intermediate destination one epoch before it departs.

Similar boundary constraints apply for airplane flights as well, but are already included in the model where the sum of flights across any one epoch cannot exceed the number of operational aircraft by type in the fleet. In the last example, if the arcs are now considered flights instead of commodity paths, the boundary node is the same, node (2,3). Thus, in optimizing flights across the window an aircraft of the same type as that flying arc (2,4,3,5) is required to terminate at node (2,3) within the window.

## A Linear Relaxation Approach

Even with the reduction of the problem size via the time window concept, the resulting mixed integer problem remains unmanageably large. A common heuristic approach, which seems appropriate in the air freight problem is to solve the linear programming relaxation of the mixed integer problem and then use a rounding procedure. Use of linear programming permits the solution of much larger and more complex problems than an integer approach. The constraint that $y_f$ be either zero or 1 is relaxed to become a continuous variable between zero and 1. Commodity values are already continuous variables. Large scale specialized linear programming approaches, such as column generation schemes, can now be applied and a solution found. A solution of the LP relaxation may end up with partial flight values and more basic flights than the mixed integer solution allows.

Thus, some rounding of the linear solution will be required to achieve integer feasibility. However, the fact that such a solution derives from an optimal solution, retains the model's optimality based approach sought in this thesis.

It appears the development of a satisfactory rounding heuristic would not be difficult. Solution values already at zero or 1 present no problem. Also, within the time window concept the only areas of concern occur at the boundaries.

The leading edge is of concern only as time 1 is achieved and flights must start repeating. Across the middle of the window, the flights need only provide feasible commodity paths for flow. These values might also change as the window is shifted in time, and there is no immediate need to force them to integer values. However, the trailing edge of the window must have integer values, for once having left an epoch, that epoch is fixed for both flights and commodity movement. In the example of the last section, the boundary conditions established termination nodes for each flight. To obtain boundary feasibility, the number of flights ending at any node must be forced to equal the (integer) number of flights which depart across the time window. Thus, if two fractional flights entered node (2,3), say via arcs (1,1,2,3) and arc (3,2,2,3) and only one flight departed, a round off procedure must be implemented. A reasonable round off rule could eliminate that flight that cost more or the one that was less utilized in terms of aircraft capacity. Once a flight is removed, the new solution could be checked for integer feasibility. If partial values still occur, the round off rule must be applied again until only integer solutions remain at the boundary.

## A Column Generation Approach

The principle ingredient in the linear relaxation approach to be proposed is a column generation scheme. Ford and Fulkerson [16] first proposed a similar column generating approach in solving a maximal multicommodity network flow problem. They employ an arc-path formulation with multicommodity flow requirements similar to the above air freight model. Tomlin, in a paper entitled "Minimum-Cost Multicommodity Network Flows" [42], applies the Ford and

Fulkerson concept to a minimum cost network flow instead of a maximal flow problem. Both node-arc and arc-path formulations are shown to decompose to the same series of subproblems. Jarvis [23] shows how the node-arc formulation becomes the same subproblem as Ford and Fulkerson presented in the arc-path model of maximal flow in a multicommodity network. Wollmer [44] expands the approach of the previous mentioned writers to include joint capacity constraints in which the upper bound on an arc is assigned to some linear combination of the arc flows for certain subsets of arcs.

An initial step in understanding what the column generation approach does is to understand its composition of various columns. The LP relaxation begins by writing the air freight model with its boundary condition constraints in the standard linear programming format.

$$\text{Minimize } z = \sum_f C_f\, y_f + \sum_p R_p\, w_p$$

$$\text{Subject to:} \quad \sum_{\{f \varepsilon \gamma^f_{i,t}:\ a_f = a\}} y_f + n_{a(i,t)} = N_{a(i,t)} \forall (i,t); \forall a,$$

$$\sum_{\{p:\ (f,i,t,j,s) \varepsilon \Gamma_p\}} w_p - A_{(a_f)} y_f + x_{(f,i,t,j,s)} = 0$$

$\forall f\ \forall (i,t,j,s)$ with
t or s < v and $i \neq j$,

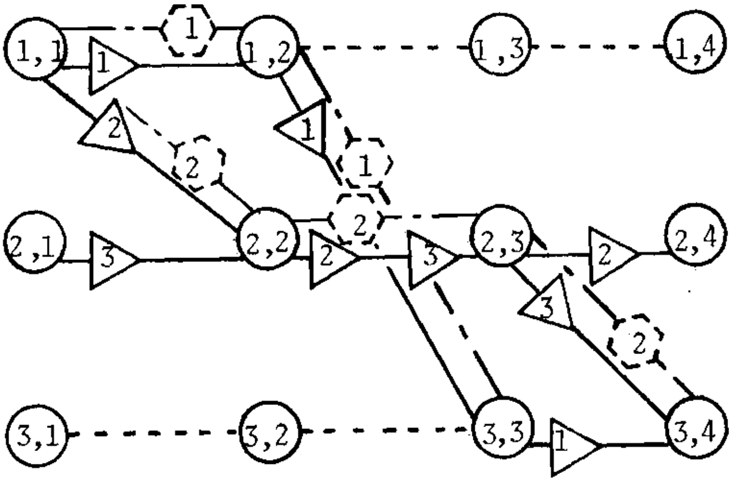
$$\sum_{p\varepsilon\Omega^{p}_{i,t}} w_p \qquad \Delta'_{k,r} \quad + g_{(i,t)(k,r)} = q_{(i,t),(k,r)} \ \forall(i,t)\ \forall(k,r),$$

$$\sum_{p\varepsilon\Omega^{p}_{i,t}} w_p \qquad (\underset{t<r\leq\min\{v,W+t\}}{\cup\Delta_{k,r}} \qquad ) g_{(i,t)_u} = Q'_{(i,t)_u}\ \forall(i,t)\ \forall u,$$

where $\gamma_{(i,t)}$ is the set of flight arcs fixed across the window boundary, similar to $\Delta'_{k,r}$, and $Q'_{(i,t)u}$ is the amount of commodity originating at (i,t) destined for u that cannot use a path that crosses the window boundary.

In the format $n_{a,(i,t)}$ represents the aircraft type artificial variables, $x_{(f,i,t,j,s)}$ represents the arc slack variables, and $g_{(i,t)(k,r)}$ and $g_{(i,t)u}$ represent the commodity artificial variables. A row exists for every arc, each boundary node of each aircraft type, and each commodity demand. Thus, a column is essentially the column of the arc-path matrix for the underlying network, with artificial arcs enforcing boundary and demand constraints.

An example is shown in Figure 4. One commodity is introduced at node (1,1) destined for location 3. Two aircraft types are used and 3 flights are flown across the window. Flights 1 and 2 are the same type of aircraft, while flight 3 is of a different type. There are 12 arcs shown on the network. As stated the matrix has sixteen rows, 12 arc capacities plus 3 aircraft type boundary nodes, plus 1



| | Flight | | | Path | | Arc | | | | | | | | | | | | Artificial Variables | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 1 | 2 | Slack Variables | | | | | | | | | | | | Commodity | Flight | | | |
| (S,1,1) | | | | 1 | 1 | | | | | | | | | | | | | -1 | | | | Q(1,1)3 |
| (1,1,1,2) | 1 | | | 1 | | 1 | | | | | | | | | | | | | | | | ∞ |
| (1,1,2,2) | | 1 | | | 1 | | 1 | | | | | | | | | | | | | | | $A_1$ |
| (2,1,2,2) | | | 1 | | | | | 1 | | | | | | | | | | | | | | ∞ |
| (3,1,3,2) | | | | | | | | | 1 | | | | | | | | | | | | | ∞ |
| (1,2,1,3) | | | | | | | | | | 1 | | | | | | | | | | | | ∞ |
| (1,2,3,3) | 1 | | | 1 | | | | | | | 1 | | | | | | | | | | | $A_1$ |
| (2,2,2,3) | | 1 | 1 | | 1 | | | | | | | 1 | | | | | | | | | | ∞ |
| (3,2,3,3) | | | | | | | | | | | | | 1 | | | | | | | | | ∞ |
| (1,3,1,4) | | | | | | | | | | | | | | 1 | | | | | | | | ∞ |
| (2,3,2,4) | | 1 | | | | | | | | | | | | | 1 | | | | | | | ∞ |
| (2,3,3,4) | | | 1 | | 1 | | | | | | | | | | | 1 | | | | | | $A_2$ |
| (3,3,3,4) | 1 | | | | | | | | | | | | | | | | 1 | | | | | ∞ |
| (2,4,$S_1$) | | 1 | | | | | | | | | | | | | | | | | -1 | | | 1 |
| (3,4,$S_1$) | 1 | | | | | | | | | | | | | | | | | | | -1 | | 1 |
| (3,4,$S_2$) | | | 1 | | | | | | | | | | | | | | | | | | -1 | 1 |
| $c_j - z_j$ | $C_f$ | | | $R_p$ | | $\pi$ | | | | | | | | | | | | $\alpha$ | $\eta$ | | | b |

Figure 4. Linear Relaxation Basis

commodity constraint. The objective function remains separate. Note that with 3 flights and 2 commodity paths, the number of columns is 21. For each commodity path column, the arcs along which the commodity travels are indicated by placing the value 1 in that row, including the artificial arc from the source to the origin. Path 1 provides an example. The arcs in the network used by path 1 are noted in the matrix by 1's in rows (S,1,1),(1,1,1,2),(1,2,3,3). Flights are similarly noted as in flight 1 whose arcs and rows are indicated as (1,1,1,2),(1,2,3,3),(3,3,3,4),(3,4,$S_1$). For flights, the destination nodes are grouped by similar aircraft type. Thus the arc (3,4,$S_1$) is for flights ending at node (3,4) and of type 1.

Though the full constraint matrix is easily visualized for a small problem like that of Figure 4, explicit statement of the matrix for even a moderate size problem would be an impossibly large task. However, in the simplex procedure only variables in the basis need be identified explicitly. These variables define the basis inverse illustrated in Figure 4 by the columns of the slack and artificial variables. This means that the use of a method of implicitly evaluating the flights and commodity paths as incoming nonbasic variables can eliminate the need to explicitly enumerate the full constraint matrix.

The second basic element in a column generation approach is the rule by which a candidate to enter the basis is selected. The normal simplex procedure identifies a candidate to enter as one that possesses a negative Cj-Zj in the objective function row. Those factors comprising the Cj-Zj for a flight or commodity path are related to values on arcs. The Zj portion of this term is comprised of a sum of simplex multipliers over constraints in which the variable appears. Designate by $\pi_{(i,t,js)}$ those multipliers for the arcs in the network; by $\eta_{a_f(j,s)}$ the simplex multipliers for the artificial variables and arcs of the boundary node for each aircraft; and by $\alpha_{(i,t)u}$ the simplex multipliers for the artificial variables and arcs of the commodity constraints. The multipliers $\eta_{(a_f(j,s)}$ and $\alpha_{(i,t)u}$ are quite distinct from each other. The aircraft type simplex multiplier is involved in flight path Cj-Zj as the principle negative value. The commodity path multiplier plays the same role in the Cj-Zj term for commodity paths.

A simplex multiplier has a value of zero until its row is capacitated. In the case of a diagonal flight arc this capacity is the capacity of the aircraft. Artificial arcs for commodities are capacitated when all the cargo has been assigned to paths connecting source and sink nodes. Artificial arcs for aircraft types reach capacity when the correct number of flights exist. Note, however, that horizontal arcs are assigned an infinite capacity and thus will never have a simplex multiplier value other than zero and will never be a candidate to enter the basis. Thus, these arcs can be removed from the basis leaving a smaller problem. Figure 5 shows an example of this reduction for the example in Figure 4. Only 7 rows and 7 corresponding simplex multipliers remain from the 16 used before.

The $c_j$ values of the $c_j - z_j$ term are basically the arc costs. For example, the $c_j$ value of flight 1 is the sum of the arc costs along arcs (1,1,1,2), (1,2,3,3), (3,3,3,4). Thus, the $c_j - z_j$ for any path is the sum of the difference between the cost and the simplex multiplier on each arc along the path. For flight 1 above, this is

$$(c_{(1,1,1,2)} - \pi_{(1,1,1,2)}) + (c_{(1,2,3,3)} - \pi_{(1,2,3,3)}) +$$

$$(c_{(3,3,3,4)} - \pi_{(3,3,3,4)}) = c_1 - z_1.$$

One method of treating arcs separately and a group of feasible paths implicitly is by using a shortest path algorithm. The algorithm identifies the shortest permissible path from every node to the source node. The length of a path is $\Sigma(c_{(i,t,j,s)} - \pi_{(i,t,j,s)})$. Implicit evaluation of all paths is accomplished by identifying only the shortest path as the incoming nonbasic column.

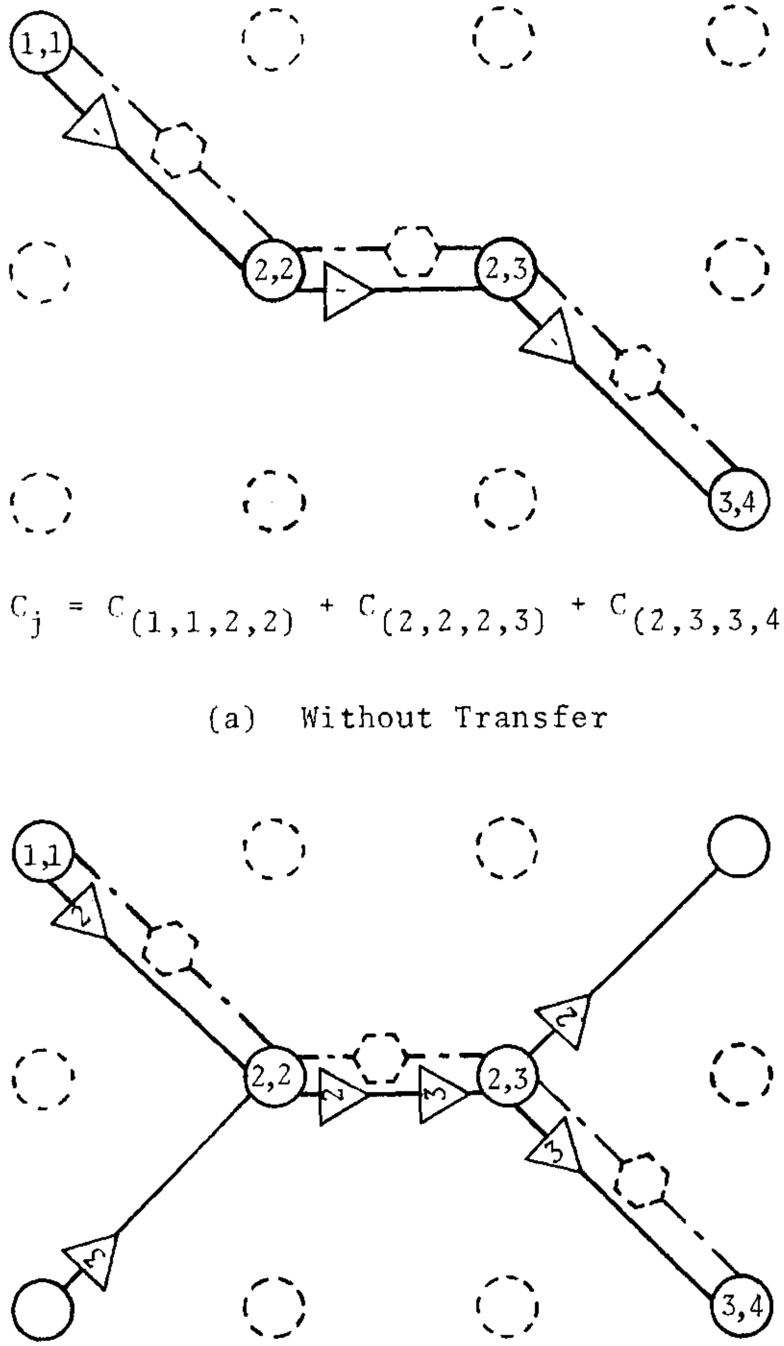
There exist some complications in the air freight model that prevent the usual shortest path procedure from being accepted as just outlined. First of all every path must be a column that can enter the basis if the shortest

| | | | | | | Slack Variable | | | Artificial Variable | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Flight | | | Path | | | | | Commodity | Flight | | | |
| | 1 | 2 | 3 | 1 | 2 | Arc | | | | | | | |
| (S,1,1) | | | | 1 | 1 | | | | -1 | | | | 12 |
| (1,1,2,2) | | 1 | | | 1 | 1 | | | | | | | 10 |
| (1,2,3,3) | 1 | | | 1 | | | 1 | | | | | | 10 |
| (2,3,3,4) | | | 1 | | 1 | | | 1 | | | | | 10 |
| $(2,4,S_{u_1})$ | | 1 | | | | | | | | -1 | | | 1 |
| $(3,4,S_{u_1})$ | 1 | | | | | | | | | | -1 | | 1 |
| $(3,4,S_{u_2})$ | | | 1 | | | | | | | | | -1 | 1 |
| $C_j - Z_j$ | $C_f$ | | | $R_p$ | | $\pi$ | | | $\alpha$ | $\eta$ | | | b |

Figure 5. Reduced Basis

path procedure is always to yield an entering column. All paths are not feasible flights or commodity paths in the air freight model. The model requires a horizontal arc between two diagonal arcs in any flight or commodity path. This prevents two consecutive diagonal arcs from constituting a feasible path. This is illustrated in Figure 2 of Chapter II in which the path consisting of arcs (1,1,2,2), (2,2,3,3) is not permissible, but path (1,1,2,2), (2,2,2,3), (2,3,3,4) is feasible. Thus, modifications of the general algorithm must be made.

One of the basic concepts for this column generation approach is the ability to assign all variable costs to the arcs independently. The air freight model introduces a contradiction to this that transfer costs cannot be associated with arcs. As long as transfer costs have positive value, this cannot be accomplished. As illustrated in Figure 6, the same commodity path through the same sequence of nodes can have different costs depending on transfers. In part (a) no transfer occurs and all is in order. Part (b), however, requires a transfer between flights and thus, incurs a transfer cost. The transfer cost cannot be assigned to either the diagonal or horizontal arcs involved. Because of this complication, two cases will be discussed. Transfer costs will be considered to be equal to zero, thus, avoiding any unassignable cost and retaining the ability to transfer commodities. Also, an approach allowing transfer costs to



$C_j = C_{(1,1,2,2)} + C_{(2,2,2,3)} + C_{(2,3,3,4)}$

(a) Without Transfer

$C_j = C_{(1,1,2,2)} + C_{(2,2,2,3)} + h_2 + C_{(2,3,3,4)}$

(b) With Transfer

Figure 6. Transfer Cost not Associated with Arcs

be positive is presented.

Besides the commodity path aspect of the column generation approach, the plane flight columns must be treated separately. Separate in the fact that the source and sink nodes of plane flights differ fundamentally from commodity path nodes. A flight has one sink node and many possible origin nodes while a commodity path has a single origin node and many possible sink nodes. Since flight costs can be very easily assigned to arcs, a modified shortest path algorithm is applicable in identifying non-basic candidates to enter the basis. The modification is similar to that for commodity paths.

Separate treatment, however, does not limit the advantage gained by the fact that both types of paths share the same arc simplex multiplier values. Thus, a change in a flight can change $\pi_{(i,t,j,s)}$ values in such a manner as to make several commodity path changes. The reverse effect also holds true. If this complimentary effect could be manipulated, an optimal solution to the LP relaxation approach could perhaps be more quickly reached. The ability to iterate between flights and commodity paths can be developed in many ways, but pursuit at this time is beyond the scope of this thesis.

Another basic point in this iterative process, is that without commodities to impose demand on the system, the airplane flights would consist of horizontal lines across the network and the trivial optimum fleet size would be zero. Thus, commodities generate the need for flights. In fact, the $c_j - z_j$ for a flight will be nonnegative until some path using the flight is basic. This means that the network used to generate incoming commodity paths must be allowed to extend beyond the one of existing flight arcs upon which to place demand. The open graph concept provides the extension required.

An open graph is defined as the network in which all feasible arcs between nodes are included in the network. An example is shown in Figure 7. Note that it requires two epochs to travel from location 1 to 2 and an arc at every epoch node of location 1 connects it to location 2. The reverse applies in travelling from location 2 to 3. It takes three epochs to travel from 2 to 3. No direct flight arc exists between locations 1 and 3 in the example. This can occur when the range of the aircraft is inadequate to connect the two locations. Conversely a closed graph would contain only selected arcs from the open graph. For instance, in travelling from location 1 to 2 only arcs (1,2,2,4) and (1,4,2,6) are permitted while from 2 to 1 only arc (2,3,1,5) is permitted. All of the previous networks illustrated were closed graph systems. Both the open and closed graph concepts for the air freight model include horizontal arcs between the nodes.

The open graph in Figure 7 is an illustration for one

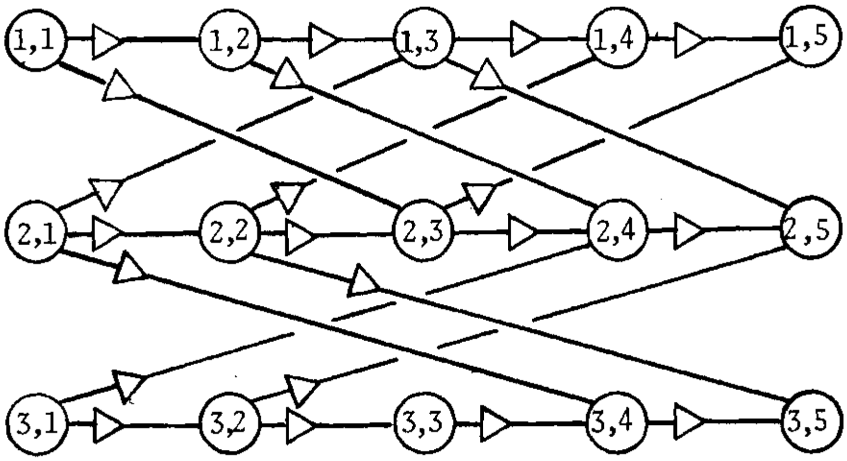


Figure 7. Open Graph, Single Aircraft Type

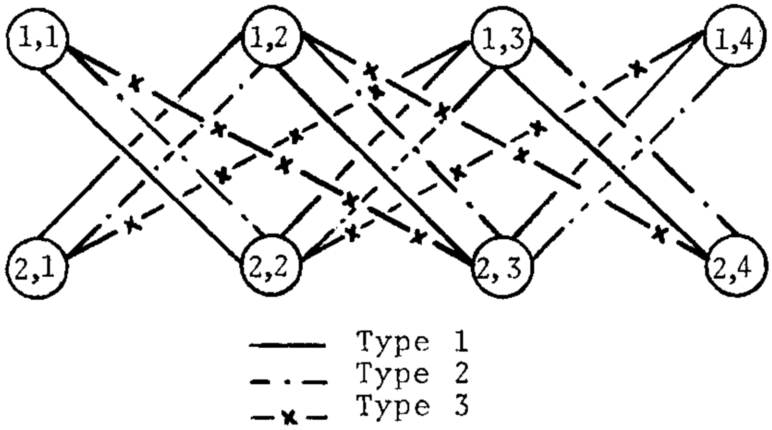


Figure 8. Open Graph, Multiple Aircraft Type

type of aircraft only. When, in the air freight model, the fleet is comprised of several aircraft types,an arc is required for each type. This is because each aircraft type has different capacity and even perhaps a different flight time requirement between locations. Figure 8 shows the simplest two location network with three aircraft types. Two aircraft have the same speed and range but different cargo capacities. The third aircraft has a different capacity as well as flying speed. The number of arcs has been tripled in this example from the same open graph with only one aircraft type. In order to simplify the examples illustrating each of the algorithms developed in the column generation approach only one aircraft type is assumed in the fleet. In reality there could be several more but the algorithm will probably be as effective on the larger network of arcs as it is on the smaller.

Summarizing,the linear programming relaxation approach proposed will include a column generation approach in which the nonbasic variables are implicitly evaluated over an open graph. First, commodity paths are evaluated to identify those flight arcs of greatest benefit. Flight algorithms then are applied introducing new flow capabilities for the commodities. This system iterates between the commodity paths and airplane flights until no further changes can be made. Finally the round-off procedure insures that integral valued flights exist at the boundary points of the time window. The central problem of this approach, i.e. efficient implicit generation of columns for flights and commodity paths, is the subject of the next chapter.

# CHAPTER IV

## IMPLICIT APPROACHES FOR SELECTING ENTERING PATHS

One of the elements of the column generating approach discussed in the last chapter, was the need to implicitly determine a nonbasic column to enter the basis. In the air freight model, an airplane flight and a commodity path can both be considered columns. Simple, efficient shortest path approaches must be developed before any implicit investigations of flights or commodity paths can be done. The problem of transfer costs not being associated with arcs, forces the examination of two cases; one case without transfer costs and the second with positive transfer costs. But, first of all, a basic understanding of shortest path procedures is required before specific algorithms can be proposed.

The general structure for the shortest path algorithm on the air freight time-space network begins by selecting a sink as a starting point and labeling it zero. The remaining nodes are labeled at positive infinity. The main process then begins, changing labels so that if

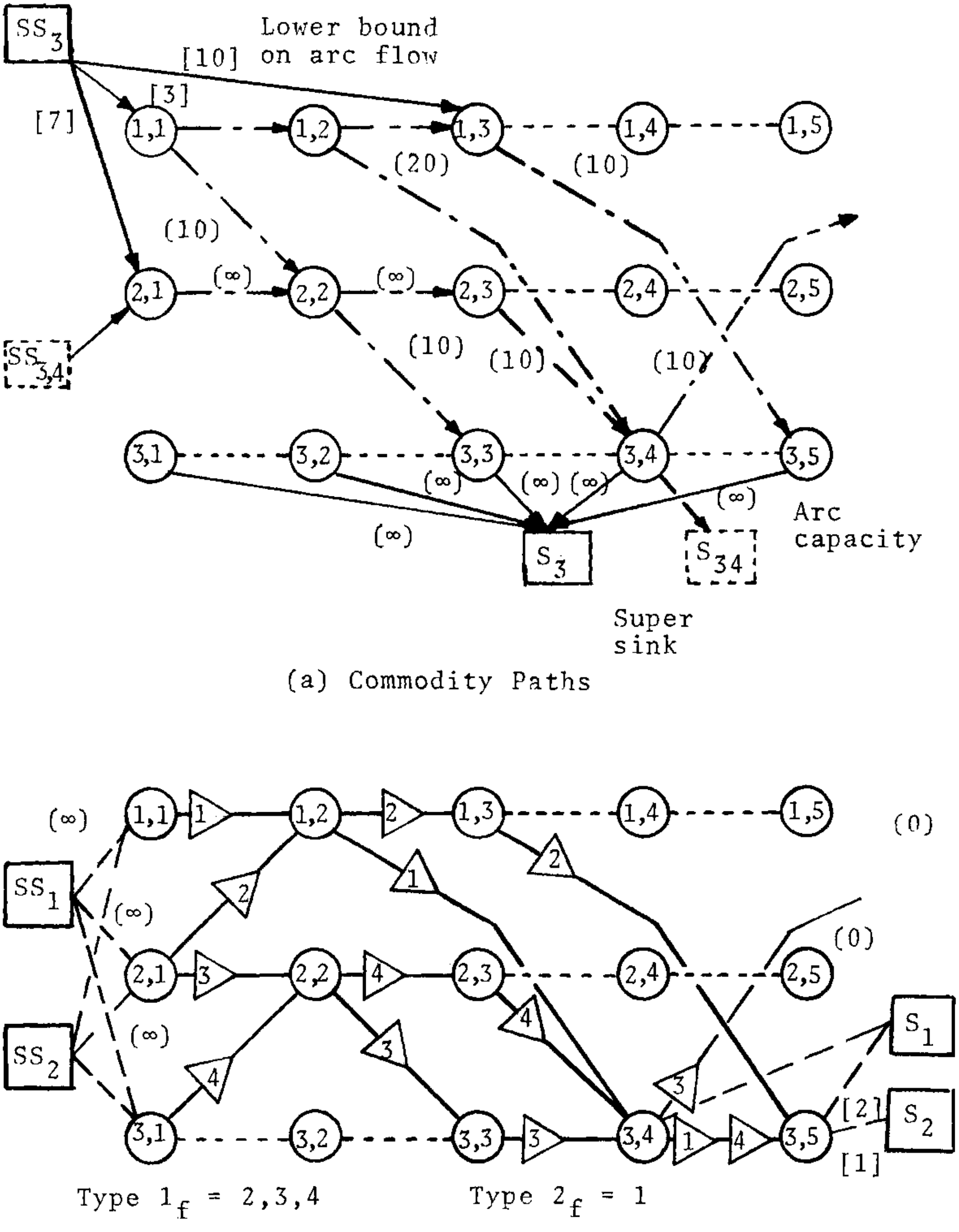
$$\delta_{(i,t)} + \ell_{(i,t,j,s)} < \delta_{(j,s)}$$

then $\delta_{(j,s)}$ is replaced by $\delta_{(i,t)} + \ell_{(i,t,j,s)}$. Here $\delta_{(i,t)}$ is the label showing the shortest path to the sink from node (i,t) and $\ell_{(i,t,j,s)}$ is the length of the arc connecting (i,t) to (j,s). Continuing this process until no more changes can be made, the resulting labels indicate the shortest path from each node to the sink. Thus, by noting labels and arc lengths, a shortest path from source to sink can be found.

Each of the approaches to be developed below is a modification of this general shortest path algorithm. Several fundamental concepts apply to all the algorithms. First of all, the air freight network is acyclic. The acyclic structure eliminates any concern over the formation of negative cycles since no cycle can be formed. It also enables the evaluation procedure at each node to be accomplished in a one-pass, dynamic programming manner. The procedure progresses through the network evaluating a node once, and once evaluated the values established for that node never change.

Second, not all directed arc combinations are permitted in constructing a path. Commodities and flights must remain in the same location at least one time epoch before moving to a different location. Thus, a special labeling at each node must be developed to keep two consecutive diagonal arcs from being considered without a horizontal arc in between. This double label will be represented as an ordered pair. The first value indicates the shortest path to the super sink from node (i,t) leaving along a diagonal arc: $\delta_{(i,t)1}$. The second of the ordered pair values is the label of the shortest path to the super sink from node (i,t) leaving along a horizontal arc: $\delta_{(i,t)2}$. The use of these labels is explained more fully in each of the algorithms.

Third, and most complicating, the network can have multiple sources and sinks. Commodities destined for a common location can be considered as a single commodity originating at several nodes at different locations and ending at several nodes at the same location. General shortest path algorithms are designed to find the shortest distance to a single node from every other node. In order to create a single sink situation, a super sink, $S_u$, that has arcs to it from each of the multiple sinks is introduced. A super source, $SS_u$, is also introduced having arcs from it to each origin node. In the case of commodity paths, the sink to super sink arcs have infinite capacity and the super source to origin nodes are capacitated below by the flow requirements of that commodity. The opposite is the case in the plane flight alignment. Here, the source arcs have infinite capacity and the sink to super sink arcs are capacitated by flow of flights through the sink node. In Figure 9, a simple illustration of each of these formats is provided. Part (a) of the figure shows a commodity path construction where the super sink $S_3$ has an arc to it from each sink node in the destination group at location 3. Each



(a) Commodity Paths

(b) Airplane Flights

Figure 9. Super Sources and Sinks

of these arcs has infinite capacity. The corresponding super source, $SS_3$, is connected to those origin nodes with destination at location 3. The lower bound on these arcs are the flow requirements of the commodity, such as the 7 units of the commodity originating at node (2,1) indicated on arc $(SS_3,2,1)$. In the flight case, part b of the figure, the arcs from $(SS_1,1,1)$, $(SS_1,2,1)$, $(SS_1,3,1)$ have infinite upper bounds. The sink arcs to the super sink, $S_1$, have fixed lower bounds as shown for arc $(3,5,S_1)$ whose lower bound is two flights of type 1 aircraft.

Note that the super source and sink are not connected to every origin node or sink node in the network at the same time. This is because of the multicommodity property of the model. What the super sink can do is connect those nodes that serve as a common sink for a commodity. And, since commodities destined for the same location actually share the same sink nodes, several commodities can be evaluated at the same time. Thus, in Figure 9 each of the commodities destined for location 3, namely (1,1),3; (1,3),3; and (2,1),3 could be evaluated implicitly, all at the same time. No commodities destined for another location could be examined, but must wait their group of nodes turn to be attached to the super sink.

The boundary constraint imposed upon commodity flow that requires feasibility within the window creates still another type of commodity condition. By adopting intermediate destinations, flow is required to reach the intermediate node by the best feasible path. Unlike the original destination conditions of having several possible destination nodes, the intermediate destination is the only node permitted. Thus, intermediate destination nodes constitute a group of one node attached to the super sink and the only commodities evaluated are those whose destination is that node only. This is shown in the example of Figure 9(a) as intermediate node (3,4). The super sink is attached to it alone, thus, the designation $S_{(3,4)}$. As well, the only super source arc is to the commodity destined for the intermediate node ($SS_{(3,4)}$, (2,1)) since (2,1),(3,4) is the only commodity involved.

Flights can be grouped together over a single aircraft type as seen in the example. This is possible because an aircraft type can be considered as a single commodity. A different type of aircraft constitutes a different commodity, thereby, probibiting the evaluation of more than one type at a time. As above, a single type of aircraft can group the boundary nodes for that type of aircraft together and evaluate them all at the same time. This is shown in Figure 9 by having two aircraft type, type 1 and type 2 and establishing separate super sources $SS_1$ and $SS_2$, and separate super sinks, $S_1$, $S_2$, to evaluate the network. Separate evaluations are required, one for type 1 and one for type 2 aircraft. With these clarifications made, the development of each type of algorithm required can be begun.

## Shortest Path Approach Without Transfer Costs

As long as all costs can be assigned to arcs independently the implicit search for entering paths by a modification of the shortest path algorithm can be used in a column generation approach. When transfer costs are set at zero, all remaining costs can be independently allocated to arcs. A zero transfer cost assumption is problably not that difficult to make, because transfer costs are likely to be small relative to others and most of the costs associated with transfers are fixed parts of ground operations. Ground crew size and equipment investment could be assigned as an overall inventory cost. Intermediate commodity storage space can be included in general overhead. Even loading and unloading activities can be balanced out over the costs assigned at entering and destination locations.

A change in the arc labeling system for a commodity path in the original air freight model from Chapter II must occur. First of all, summing over the zero flight arcs where transfers occur can be discontinued. Transfers can still occur, but the concern over which arc or how often they occur is no longer necessary.

Second, instead of keeping track of the exact flight arc being used between locations, the only requirement now is to be able to distinguish airplane types. Thus, the arcs in this algorithm will be written as (a,i,t,j,s) where a is the plane type of the arc. This simplicity is essential, since the commodity path algorithm will be using an open graph network; specific flights are not recorded.

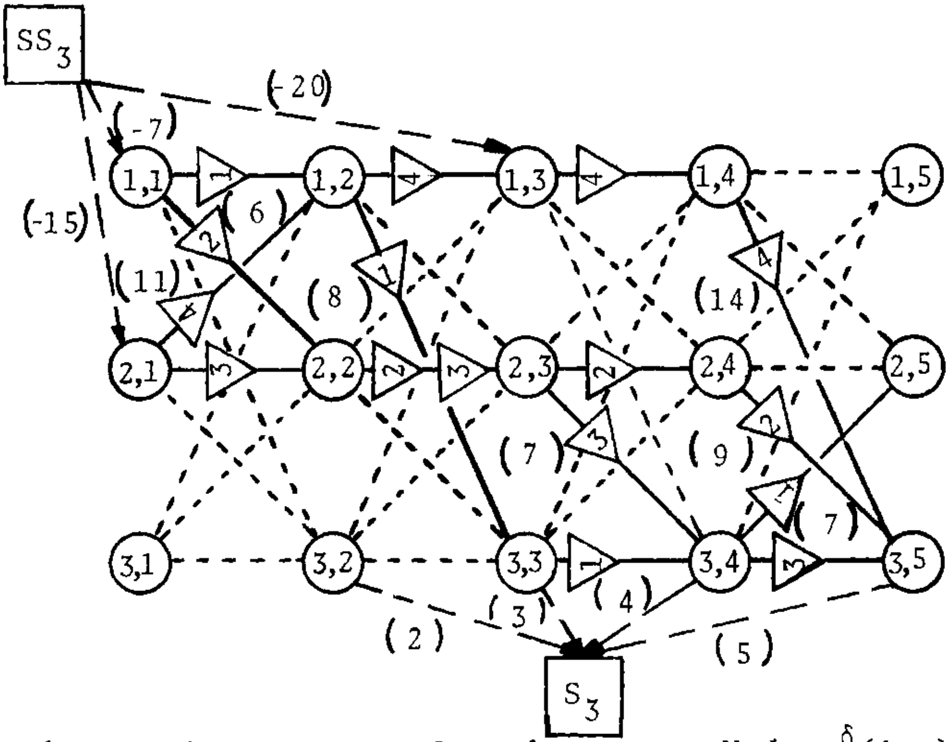
The next major change is the commodity inventory cost allocation. This cost depends on the length of time it takes for a commodity to reach its destination and not on the specific path. Thus, inventory cost can be collected at one arc leaving the others without a cost. The only arc with a cost will be between the destination node and the super sink. This is illustrated in the example to follow in Figure 10.

Once the group of nodes (or single node in the case of an intermediate destination), u, is selected; the appropriate super source, $SS_u$, and super sink, $S_u$, arcs can be connected. These arcs are considered as horizontal arcs in the labeling and evaluation rules of the algorithm. This means that at the sink nodes, the evaluated double label will be $(\infty, c_{(u,s),S_u})$, and the super source label can select the minimum of either the diagonal or horizontal label at the origin node.

The algorithm's rules are now presented.

## Commodity Path Algorithm

Step 0. Initialization: Set all label values at every node, except the super sink, at infinity. The



| Flight | Arc (1,i,t,j,s) | Length (c-n)(1,itjs) |
|---|---|---|
| 1 | (1,1,1,2) | 0 |
| | (1,2,3,3) | 8 |
| | (3,3,3,4) | 0 |
| | (3,4,2,5) | 7 |
| 2 | (1,1,2,2) | 6 |
| | (2,2,2,3) | 0 |
| | (2,3,2,4) | 0 |
| | (2,4,3,5) | 9 |
| 3 | (2,1,2,2) | 0 |
| | (2,2,2,3) | 0 |
| | (2,3,3,4) | 7 |
| | (3,4,3,5) | 0 |
| 4 | (2,1,1,2) | 11 |
| | (1,2,1,3) | 0 |
| | (1,3,1,4) | 0 |
| | (1,4,3,5) | 14 |

| Node | $^{\delta}$(i,t)1 | $^{\delta}$(i,t)2 |
|---|---|---|
| $S_3$ | 0 | 0 |
| (3,5) | ∞ | 5 |
| (2,5) | ∞ | ∞ |
| (1,5) | ∞ | ∞ |
| (3,4) | ∞ | 4 |
| (2,4) | 14 | ∞ |
| (1,4) | 19 | ∞ |
| (3,3) | ∞ | 3 |
| (2,3) | 11 | 14 |
| (1,3) | 24 | 19 |
| (3,2) | 34 | 2 |
| (2,2) | 23 | 11 |
| (1,2) | 11 | 19 |
| (3,1) | 31 | 2 |
| (2,1) | 22 | 11 |
| (1,1) | 17 | 11 |
| $SS_3$ | -4 node (2,1) | |

Figure 10. Commodity Path Algorithm

super sink label value is zero. The first node to be evaluated is the sink node with the largest epoch time and then the largest location value.

$$(i,t) = \underset{i}{\text{Maximum}} \{\underset{t \varepsilon u}{\text{Maximum}} (i,t)\},$$

where u is the group of sink nodes.

Step 1. Node Evaluation: Evaluate both labels at the node as follows:

a. Diagonal Arc Label, $\delta_{(i,t)1}$: From all diagonal arcs the smallest sum of the length of the arc plus the horizontal label at the destination node of the arc.

$$\delta_{(i,t)1} = \underset{(j,s)\varepsilon\beta_{(i,t)}}{\text{Min}} \{(c_{(a,i,t,j,s)} - \pi_{(a,i,t,j,s)}) + \delta_{(j,s)2}\},$$

where $\beta_{(i,t)}$ is the set of diagonal arcs departing from the node (i,t).

b. Horizontal Arc Label, $\delta_{(i,t)2}$: From the horizontal arc leaving the node (i,t), select the minimum of the ordered pair of labels at node (i,t+1).

$$\delta_{(i,t)2} = \text{Minimum } (\delta_{(i,t+1)1}, \delta_{(i,t+1)2})$$

For the labels where i = u, a sink node, the label is

$$\delta(u,t)2 \equiv c_{(u,t)}S_u$$

where $c_{(u,t)S_u}$ is the inventory cost over the arc to the super sink, $S_u$.

Step 2. Node Sequence: Repeat Step 1 for another node (i,t)' according to the following priorities:

a. The next node in the same epoch:

$$(i,t)' = (i-1,t); \text{ if } i-1 = 0 \text{ check rule b.}$$

b. The largest location in the next epoch:

$$(i,t)' = (\text{Max } i,t-1); \text{ if } t-1 = 0 \text{ check rule c.}$$

c. Since both i and t = 0, all nodes have been evaluated. Continue on to Step 3.

Step 3. Shortest Path Determination: Once all nodes have been examined, the super source is all that remains. Find the shortest path to the super source:

$$\delta_{SS_u} = \underset{(i,t)\varepsilon\sigma_u}{\text{Min}} \{\text{Min}(\delta_{(i,t)1}, \delta_{(i,t)2}) + \alpha_{(i,t)u}\}$$

where $\sigma_u$ is the set of origin nodes with destination within group u, and $\alpha_{(i,t)u}$ is the dual variable of the commodity (i,t)u.

Step 4. Nonbasic Candidate:

a. If the shortest path in Step 3 is negative, i.e. $\delta_{SS_u} < 0$, then by retracing the labels the candidate commodity path is found.

b. If the shortest path in Step 3 is equal to or greater than zero, i.e. $\delta_{SS_u} \geq 0$, then stop. No nonbasic commodity path to this sink group can enter the basis.

If a candidate has been found by the algorithm, then a simplex procedure can be followed to enter the path. If not the algorithm stops. Every group must be evaluated including intermediate destination nodes. The procedure for selecting groups is arbitrary.

A simple example problem is presented in Figure 10. For ease of conception, a single aircraft type is shown. The only change in adding aircraft types would be to add more arcs to the open graph. Basic flight arcs are represented with solid lines while the non-basic flight arcs are noted by dotted lines. All arc lengths are shown in parenthesis and have been computed as $(c_{(1,i,t,j,s)} - \pi_{(1,i,t,j,s)})$. The length of the dotted diagonal arcs are fixed at 20. A separate table is included for the final label values at each node. The entire example will not be explained but key points in the algorithm will be pointed out.

Once the group of nodes at location 3 have been selected, the super sink and super source are connected to the network. Arcs $(3,2,S_3)$,$(3,3,S_3)$,$(3,4,S_3)$ and $(3,5,S_3)$ are added at commodity cost 2,3,4,5, respectively. This corresponds to increasing inventory cost the longer the commodity stays in the network. Next, the super source connects to origin nodes (1,1), (1,3) and (2,1) with the associated dual multiplier costs, $\alpha_{(i,t)3}$, of -7, -20, -15. Initialization of the network sets all label valves at infinity, except the super source which is set at zero. Beginning at the first evaluation node (3,5), it is recognized as a sink node and Step 1b assigns the labels $(\infty,5)$. Step 2 next selects node (2,5). Applying Step 1 on node (2,5) causes no changes and the labels remain at $(\infty,\infty)$. The same thing occurs at the next node (1,5). Step 2 now designates node (3,4) as the next to be evaluated. This process continues. At node (1,2) for instance, the diagonal label evaluation in Step 1a compares the diagonal arcs (1,2,2,3) and (1,2,3,3). The equations show:

$$\delta_{(1,2)1} = \text{Minimum} \begin{array}{l} (c_{(1,1,2,2,3)} - \pi_{(1,1,2,2,3)}) + \delta_{(2,3)2} = 20+14=34 \\ (c_{(1,1,2,3,3)} - \pi_{(1,1,2,3,3)}) + \delta_{(3,3)2} = 8+3=11 \end{array}$$

Thus, the value of the label $\delta_{(1,2)1}$ is 11. Note that the double diagonal arc (1,2,2,3),(2,3,3,4) was not considered because only the horizontal label at node (2,3) was utilized. Thus, only proper air freight commodity paths are evaluated. Finally, when the super source is evaluated the equations show:

$$\delta_{SS_3} = \text{Minimum}\left\{\begin{array}{l} \text{Min}(\delta_{(1,1)1}, \delta_{(1,1)2}) + \alpha_{(1,1)3} = 11\text{-}7 = 4 \\ \text{Min}(\delta_{(1,3)1}, \delta_{(1,3)2}) + \alpha_{(1,3)3} = 19\text{-}20 = -1 \\ \text{Min}(\delta_{(2,1)1}, \delta_{(2,1)2}) + \alpha_{(2,1)3} = 11\text{-}15 = -4 \end{array}\right\}$$

The shortest path to the super source is a -4 and thus, commodity (2,1),3 will provide a new non-basic path to enter the basis. The path is simply identified by retracing through the network using labels and arc lengths.

To prove that the algorithm actually finds the shortest path through the network to the super sink, it must be shown that the final double labels at each node do indeed correspond to the shortest path from that node. First of all, the nodes directly connected to the super sink are obviously labeled correctly. Now assume that the double labels are correct for all nodes with time components greater than that for a given node (i,t), (i.e.: all nodes to which this node can lead). A path or flight in the air freight network that departs node (i,t) along a diagonal arc must next follow a horizontal arc at its next stop. Thus, clearly the minimum sum of the length of all possible diagonal arcs leaving node (i,t) plus the horizontal arc label at the destination of those arcs must be the length of the shortest path via a departing diagonal arc. This is exactly the rule for selecting the diagonal arc label at node (i,t),

$$\delta_{(i,t)1} = \underset{(j,s)\varepsilon\beta_{(i,t)}}{\text{Minimum}} \{\ell_{(i,t,j,s)} + \delta_{(j,s)2}\},$$

where $\beta_{(i,t)}$ is the set of destination nodes for diagonal arcs leaving (i,t).

At the destination end of a horizontal arc leaving node (i,t) either a diagonal or horizontal arc may be taken. Thus, the shortest path to (i,t) is clearly the minimum of the diagonal and horizontal arc labels at the destination end of the horizontal arc. Once again, this is the exact rule for selecting the horizontal arc label at node (i,t)

$$\delta_{(i,t)2} = \text{Minimum } (\delta_{(i,t+1)1}, \delta_{(i,t+1)2})$$

Thus, both labels at (i,t) have set at the shortest path to the super sink, leaving node (i,t) on the diagonal and horizontal arcs.

In assuring the shortest path from the super sink, the network nodes have been shown to be correct and the arc length from the super source to each origin node (i,t) is known $\alpha_{(i,t)u}$. Since a commodity entering the system at a node can depart, either on a diagonal or horizontal arc, the shortest path is simply the minimum sum of all the arc lengths from the super source to the origin nodes plus the minimum of both labels at the origin node reached. This is the rule applied in the application:

$$\delta_{SS_u} = \underset{(i,t)\varepsilon\sigma_u}{\text{Minimum}} \{\text{Minimum}(\delta_{(i,t)1}, \delta_{(i,t)2}) + \alpha_{(i,t)u}\},$$

where $\sigma_u$ is the set of all origin nodes with destination u. The shortest path from super source to super sink is in fact the one found by applying the rules in the algorithm.

## Shortest Path Approach for Airplane Flights

Airplane flights are the only means of moving commodities through the network. And, as discussed before, flights would tend to be horizontal lines if the commodity path algorithm were not solved on an open graph in order to introduce new diagonal flight arcs into the basis. Thus, the network for this algorithm is also an open graph. The arc lengths will be different than in the commodity path algorithm. Variable flight costs do depend on the route flown, causing costs to be associated with individual arcs instead of lumped together as in commodities. This means that the $c_{(a,i,t,js)}$ will have a value greater than zero on all diagonal arcs. Simplex variables $\pi_{(a,i,t,j,s)}$ are the same as for the commodity path algorithm since the same rows are involved.

The major difference between the two algorithms is the source and sink node structure. As illustrated in Figure 9 and discussed in that section, flights have a single sink node and many possible origin nodes. This is the exact opposite of commodities. Sink nodes are the boundary condition nodes for a particular flight and source nodes are found in the first epoch of the window at every location. In the case of flights only one aircraft type can be evaluated at a time over all the boundary nodes where aircraft of that type terminate. This is similar to commodity destination groups, but more flexible in that all locations can be grouped together. Note, also, that no intermediate destination nodes are required because the flight extends entirely across the window. The number of arcs in the open graph is also reduced to only those of the aircraft type being considered.

Once the aircraft type has been selected and the relevant boundary nodes identified, the super sink can be connected. The arcs are directed from each sink node to the super sink. The arc cost is zero, since the costs cannot be bundled as in the commodity case. The sink arcs correspond to commodity source arcs in that lower bounds on flow are placed on the arc to force flow in the system. Integer valued, these lower bounds represent the number of aircraft, of the type being considered, that are supposed to terminate at the node within the window. The fact that a lower capacity is forced also means that the dual variable for the flights terminating at this node is applied to the sink arc. On the commodity case the dual variable was applied at the source arcs. Thus, the arc length $c_{(a,(u,t)S_u)}-\eta_{a(i,t)}$ is simply the negative dual variable $-\eta_{a(i,t)}$ since the cost is zero.

At the other end of the window, all of the location nodes in the first time epoch are now source nodes. The connection to the super source is across a zero cost arc with infinite capacity. Thus, no arc length is assignable to the source arcs.

Once the super source and sink nodes are connected to the network, only minor changes in the rules that applied in the commodity path algorithm need be made in formulating the airplane flight algorithm. These are easily understood as occurring at the sink and source nodes.

Airplane Flight Algorithm

Step 0. Initialization: Set all label values at every node, except the super sink, at infinity. The super sink label value is zero. The first node to be evaluated is the sink node with the largest epoch time and then the largest location node within that epoch.

$$(i,t) = \underset{i}{\text{Maximum}} \{\underset{(i,t)\varepsilon u}{\text{Maximum}} (i,t)\}$$

where u is the group of sink nodes.

Step 1. Node evaluation: Evaluate both labels at the node as follows:

a. Diagonal Arc Label, $\delta_{(i,t)1}$: From all diagonal arcs leaving node (i,t), select the smallest sum of the length of the arc plus the horizontal label at the destination node of the arc.

$$\delta_{(i,t)1} = \underset{(j,s)\varepsilon\beta_{(i,t)}}{\text{Minimum}} \{(c_{(a,i,t,j,s)} - \pi_{(a,i,t,js)}) + \delta_{(j,s)2}\},$$

where $\beta_{(i,t)}$ is the set of diagonal arcs departing from the node (i,t).

b. Horizontal Arc Label, $\delta_{(i,t)2}$: From the horizontal arc leaving node (i,t), select the minimum of the ordered pair of labels at node (i,t+1)

$$\delta_{(i,t)2} = \text{Minimum } (\delta_{(i,t+1)1}, \delta_{(i,t+1)2})$$

For the labels where (i,t)εu, a sink node is evaluated, the label is

$$\delta_{(i,t)2} \equiv -\eta_{a(i,t)}$$

where $\eta_{a(i,t)}$ is the dual variable for the node (i,t) and aircraft type a.

Step 2. Node Evaluation Sequence: Repeat Step 1 for another node (i,t) according to the following priorities:

a. The next node in the same epoch:

$(i,t)' = (i-1,t)$; if $i-1=0$ check rule b.

b. The largest location in the next epoch:

$(i,t)' =$ (Maximum $i,t-1$); if $t-1=0$ check rule c.

c. Since both i and t are equal to zero, all nodes have been evaluated. Proceed to Step 3.

Step 3. Shortest Path Determination: Once all the nodes have been examined, the shortest path to the source is easily found. Select the minimum of all the labels in the source node epoch:

$$\delta_{SS_a} = \underset{i}{\text{Minimum}} \{\text{Minimum } (\delta_{(i,1)1}, \delta_{(i,1)2})\}$$

where the first epoch of the window equals 1.

Step 4. Non-basic Candidate:

a. If the shortest path in Step 3 is negative, i.e. $\delta_{SS_a} < 0$, then by retracing the labels, the candidate airplane flight is found.

b. If the shortest path in Step 3 is equal to or greater than zero, i.e. $\delta_{SS_a} \geq 0$, then stop. No nonbasic flight for this aircraft type can enter the basis.

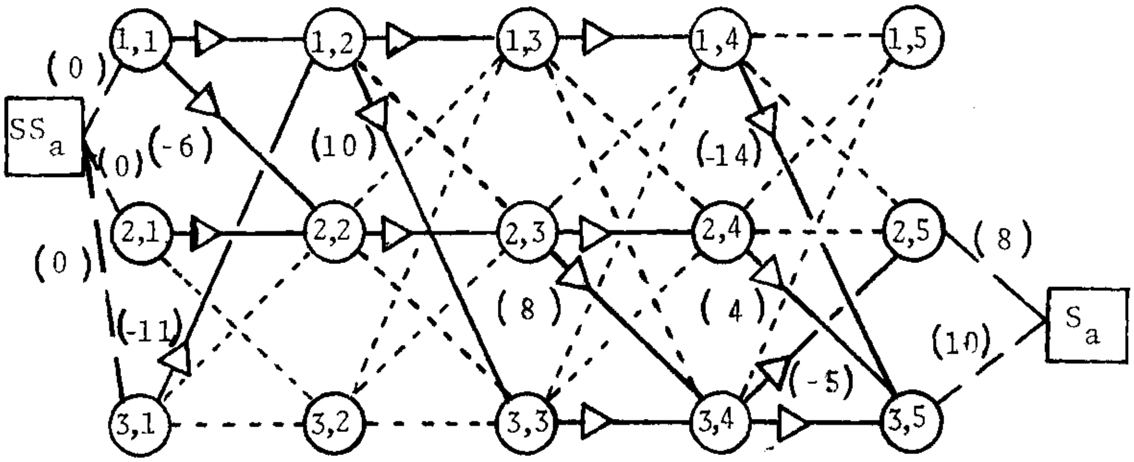
If a candidate is found, it can be entered into the basis. All of the aircraft types must eventually be evaluated. Once no more candidates can be found for each of the aircraft types, the flight network is optimal. The procedure can now iterate back to the commodity path algorithm or if that is optimal as well, the round off procedure would be applied next. Note that fewer arcs are included in this network than in the complete open graph of the commodity path algorithm. Also, the flight algorithm need be applied only once for each type of aircraft instead of several times over different sink groups. Thus, the iteration process between the algorithms would probably spend more time in flight algorithm as it would probably generate optimal solutions more quickly than the commodity path algorithm.

As the window reaches the true time 1 in the overall network, the source node conditions change. Similar type aircraft flights are required to repeat. This actually reduces the possible source node locations to those nodes that were sink nodes when the window included time T as its boundary condition epoch. The source arcs also now have lower bound capacities and dual variable values that must be considered in the algorithm. These additional conditions can be easily implemented. The rule for determining the shortest path to the super source, Step 3, simply adds the fact that the source arcs have a length. The minimum of the smallest label at a node plus the arc length to the super source still can be bound and identified as the overall shortest path. Thus, the repeating flight aspects of the model can be readily handled.

An example network is illustrated in Figure 11. Only those basic flight paths and arcs with negative length are drawn as solid lines. Dotted lines are used to denote the other arcs in the open graph for this aircraft type. For simplicity, the lengths of the dotted arcs are assumed to be 15. The same procedure as in the commodity path algorithm with the modifications made for flights results in the labels and shortest path as seen in the figure. Infeasible paths are still not permitted by the algorithm. The shortest path characteristics and evaluative steps for the flight shortest path are the same as in the commodity path algorithm. Differing sink and source arc costs between the two approaches does not effect the procedural similarities. Thus, the proof of convergence for the commodity path algorithm applies equally well for the airplane flight algorithm.

### Shortest Path Approach with Transfer Costs

Sometimes the circumstances surrounding commodity handling will not permit the cost of transferring commodities at an intermediate location to be considered zero. Exceptional labor cost, additional administrative attention and a premium placed on storage space could easily force a positive transfer cost to be assigned at each location. Thus, the assumption



| Arc | Cost |
|---|---|
| (1,1,2,2) | -6 |
| (3,1,1,2) | -11 |
| (1,2,3,3) | 10 |
| (2,3,3,4) | 8 |
| (1,4,3,5) | -14 |
| (2,4,3,5) | 4 |
| (3,4,2,5) | -5 |
| (2,5,S$_a$) | 8 |
| (3,5,S$_a$) | 10 |
| (SS$_a$1,1) | 0 |
| (SS$_a$2,1) | 0 |
| (SS$_a$3,1) | 0 |

All other = 15
Diagonal Arcs

All Horizontal = 0
Arcs

| Node | $\delta_{(i,t)1}$ | $\delta_{(i,t)2}$ |
|---|---|---|
| S$_a$ | 0 | 0 |
| (3,5) | ∞ | -10 |
| (2,5) | ∞ | -8 |
| (1,5) | ∞ | ∞ |
| (3,4) | -13 | -10 |
| (2,4) | -6 | -8 |
| (1,4) | -24 | ∞ |
| (3,3) | 7 | -13 |
| (2,3) | 2 | -8 |
| (1,3) | 5 | -24 |
| (3,2) | -9 | -13 |
| (2,2) | -9 | -8 |
| (1,2) | -3 | -24 |
| (3,1) | -35 | -13 |
| (2,1) | -9 | -9 |
| (1,1) | -14 | -24 |
| S$_a$ | -35 | node (3,1) |

Figure 11. Airplane Flight Algorithm

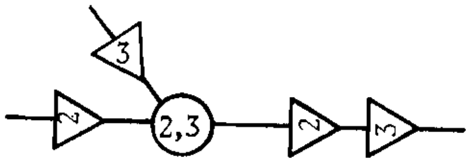
of the previous section is not permitted.

Once again, the fact that transfer costs are not assignable to arcs prevents us from utilizing the previously developed algorithm. The solution can be found in being able to remember the costs for certain commodity path-flight matchings at a location or node. This was illustrated in Figure 6. If at node (2,3) the path length to the super sink was the same for both flights 1 and 3, but longer for flight 2, then at node (2,2) with zero transfer cost either flight 1 or 2 could be chosen and the optimal guaranteed. However, with positive transfer costs, this is no longer true. To select flight 2 incurs a transfer cost and thus, is longer than the path when flight 1 is chosen. A way of remembering what the cost on all three flights from location 2 is needed. In other words, labels are required that show what the length of the path from node (2,2) to the super sink is, given cargo leave that node on flights 1 or 2 or 3. Thus, when a decision comes to choose the diagonal arc for flight 1 or 2, the paths are easily distinguished. Either stay on flight 1 or 2 or transfer from 3 to 2. To stay on 1 is less costly than staying on 2. To pay and transfer from flight 3 to 2 is more than staying on flight 1. Thus, flight 1 is chosen.

The label system required can be developed around a table. The axis of the table are locations along one side and flight along the other. The shortest path for a commodity from location 2 to the super sink at the present time, given that it arrives on flight 3, the next time it arrives at location 2, is placed at the intersection of flight 3 and location 2. An example is presented in Figure 12. On the table, at the intersection of flight 3 and location 2, i.e. $e_{32}$, the quantity 10 is given. This means that at node (2,t) if flight 3 were arriving the shortest path on to the super sink is 10. No matter what flight the commodity takes leaving location 2, as long as it arrives on flight 3, the shortest path remains 10. Actually to arrive on flight 2 is the shortest path with a value of 9. However, because of path costs and transfer costs, the cost of arriving on flight 3 is higher.

When this table is used in conjunction with the double labeling procedure developed earlier an optimal path can be determined. Some rule changes are required, however. First of all, fix the double labels at the node being examined and change other node labels based on it rather than labeling it based on other node labels. The label values may change several times before being evaluated, but during the evaluation, and thereafter, the labels are fixed.

Another change is that diagonal labels are no longer changed without regard to the flight across that diagonal arc. The path across that arc is the sum of the arc length and the table entry of the particular flight and location pair considered. Major table changes occur at horizontal



Locations = i

Flights = f

| | 1 | 2 | 3 | M |
|---|---|---|---|---|
| 1 | 16 | 15 | $e_{13}$ | ... |
| 2 | 14 | 9 | $e_{23}$ | ... |
| 3 | 12 | 10 | $e_{33}$ | ... |
| ⋮ | ⋮ | ⋮ | ⋮ | |
| N | | | | |

Figure 12. Flight-Location Table

label evaluations. This is natural since transfer costs are incurred during horizontal arcs. The shortest path of the table entries for that location is chosen as the horizontal label for the node at the next epoch, t-1. The transfer cost is added to this value and then compared to the table entries. If the existing entries at that location for each flight are less than the shortest path plus transfer cost then no changes are made. For every entry at that location larger than this sum, it is replaced by the smaller of the two. Thus, if it is cheaper not to transfer at this location, none is made but, if a transfer can be made at a lesser expense, it is done.

Because of fact that shortest paths must be remembered in the context of a particular airplane flight, the concept of an open graph can no longer be used. The open graph dealt only with feasible aircraft type arcs that were blended into flights by the airplane flight algorithm. Since a closed graph is the only solution, the next question is that of selecting "good" flights for the graph. A "good" set of flights would be those already close to optimal in feasibility and requiring minor commodity changes due to transfer cost influences. Here an earlier application of the algorithm without transfer costs could provide the needed "good" flights. How many to select or when is a question beyond the scope of this thesis.

Once the closed graph has been determined the procedures become similar to the commodity path procedure previously developed. First a group of sink nodes are selected (including the intermediate destination nodes, as before). The super source and super sink nodes are connected next. The sink node labels are determined as before.

The transfer cost algorithm rules can be stated as follows.

## Transfer Cost Algorithm

Step 0. Initialization: Set all labels values and table values, except the sink nodes and super sink at infinity. The super sink value is zero. The sink node labels will both be set at the cost of the arc to the super sink.

$$^{\delta}(u,s)1 = {}^{\delta}(u,s)2 \equiv {}^{c}(u,s,S_u) \quad (u,s)\epsilon u'$$

where u is again the group of sink nodes. The first node to be evaluated is the largest location node in the largest epoch of the group of destination nodes.

$$(j,s) = \underset{j}{\text{Maximum}} \{\underset{(j,s)\epsilon u}{\text{Maximum}} (j,s)\}.$$

Step 1. Diagonal Arc Evaluation: If no diagonal arcs enter the node being evaluated, (j,s), then proceed to Step 2. Otherwise, for each diagonal arc (f,i,t,j,s) arriving at node (j,s):

a. Change the table entry at location i and flight f to the minimum of either the present entry value or the sum of the table entry at location j for flight f plus the length of the arc (f,i,t,j,s).

$$e_{fi} = \underset{(i,t)\varepsilon\phi(j,s)}{\mathrm{Min}} \{e_{fi}, e_{fj} + c_{(f,i,t,js)} - \pi_{(f,i,t,js)}\}$$

where $\phi_{(j,s)}$ is the set of departure nodes for all diagonal arcs arriving at node (j,s).

b. Change the diagonal arc label at each node (i,t) to the minimum of either the current label value or the table entry value just computed in Step 1a.

$$\delta_{(i,t)1} = \mathrm{Min}\ (\delta_{(i,t)1}, e_{fi})$$

Step 2. Horizontal Arc Evaluation: If the node is from the group of destination nodes proceed to Step 3. Otherwise:

a. Set the horizontal arc label for the node (j,s-1) at the minimum of the horizontal and diagonal labels of the node being examined, (j,s).

b. Reevaluate the entire column of table entries for location j. For each flight entry f in column j select the minimum of the present table value and the horizontal label determined in Step 2a, plus the transfer cost at location j,Hj.

$$e_{fj} = \underset{f}{\text{Minimum}} \; (e_{fj}, \delta_{(j,s-1)2} + H_j)$$

Step 3. Node Examination Sequence: Repeat Steps 1 and 2 for another node (i,t)' according to the following Priorities:

a. The next node in the same epoch:

(j,s)' = (j-1,s) if j-1=0, check rule b.

b. The maximum node in the next epoch:

(j,s)' = (Maximum j,s-1); if s-1=0, check rule c.

c. Since both j and s are equal to zero, all nodes have been evaluated. Proceed to Step 4.

Step 4. Shortest Path Determination: Once all nodes have been examined the super source is all that remains. Find the shortest path to the super source:

$$\delta_{SS_u} = \underset{(j,s)\varepsilon\sigma_u}{\text{Minimum}} \; \{\text{Minimum} \; (\delta_{(j,s)1}, \delta_{(j,s)2}) + \alpha_{(j,s)u}\}$$

where $\sigma_u$ is the set of origin nodes with destination within the group u, and $\alpha_{(j,s)u}$ is the dual variable of the commodity (j,s)u.

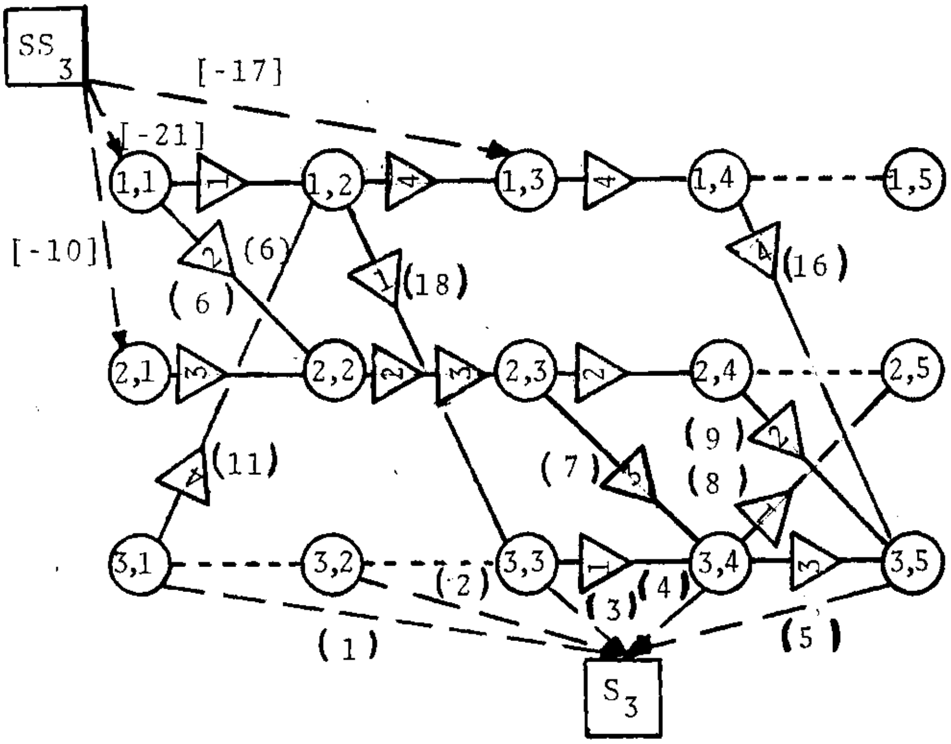
Step 5. Non-basic Candidate: In determining a nonbasic candidate the length of the shortest path to the super source in Step 4 must be known.

a. If the shortest path in Step 4 is negative, i.e. $\delta_{SS_u} < 0$, then by retracing the labels, the candidate commodity path is found.

b. If the shortest path in Step 4 is equal to or greater than zero, i.e. $\delta_{SS_u} \geq 0$, then stop. No candidate exists.

Once the algorithm finishes with the group of destination nodes u, then the same procedure as before is implemented. Either a candidate is found and entered into the basis, or no candidate exists and a new group of sink nodes must be chosen. Once all of the groups fail to find a candidate then the network is optimal over the closed graph. No iteration with the flight algorithm is possible since no nonbasic flight arcs exist in the closed graph. The rounding procedure can now be applied and the window moved another epoch toward time 1.

Figure 13 and Table 1 show how the same example problem as presented for the commodity path algorithm is solved using the transfer cost algorithm. One of the first things noticeably different is the number of arcs in the network. As stated, the closed graph uses only "good" flights generated in the first phase of the procedure.



| $H_j$ = | 4 | 5 |
|---|---|---|
| j = | 1 | 2 |
| f = 1 | 21 | 16 |
| 2 | 20 | 14 |
| 3 | 24 | 11 |
| 4 | 21 | 16 |
| 0 | | |

| Node | $\delta(j,s)1$ | $\delta(j,s)2$ |
|---|---|---|
| $S_3$ | 0 | 0 |
| 3,5 | 5 | 5 |
| 2,5 | ∞ | ∞ |
| 1,5 | ∞ | ∞ |
| 3,4 | 4 | 4 |
| 2,4 | 14 | ∞ |
| 1,4 | 21 | ∞ |
| 3,3 | 3 | 3 |
| 2,3 | 11 | 14 |
| 1,3 | ∞ | 21 |
| 3,2 | 2 | 2 |
| 2,2 | ∞ | 11 |
| 1,2 | 21 | 21 |
| 3,1 | 1 | 1 |
| 2,1 | ∞ | 11 |
| 1,1 | 20 | 21 |
| $SS_3$ | -1 | Node (1,1) |

Figure 13. Transfer Cost Algorithm

Table 1. Transfer Cost Algorithm

| Algorithm Step | Node Evaluated | Node Changed | Label Changes | Table Changes |
|---|---|---|---|---|
| 0 | | All | Set at Infinity | Set at Infinity |
| | Su | (3,5) | (5,5) | None |
| | | (3,4) | (4,4) | None |
| | | (3,3) | (3,3) | None |
| | | (3,2) | (2,2) | None |
| | | (3,1) | (1,1) | None |
| 1 | (3,5) | (1,4) | $\delta_{(1,4)1}=5+16=21$ | $e_{41}=21$ |
| | (3,5) | (2,4) | $\delta_{(2,4)1}=5+9=14$ | $e_{22}=14$ |
| 1 | (2,5) | (3,4) | No Change | |
| 2 | (2,5) | (2,4) | No Change | $e_{12}=e_{32}=e_{42}=19$ |
| 2 | (1,5) | (1,4) | No Change | $e_{11}=e_{21}=e_{31}=25$ |
| 1 | (3,4) | (2,3) | $\delta_{(2,3),1}=4+7=11$ | $e_{32}=11$ |
| 2 | (2,4) | (2,3) | $\delta_{(2,3)2}=14$ | $e_{12}=e_{42}=16$ |
| 2 | (1,4) | (1,3) | $\delta_{(1,3)2}=21$ | No Change |
| 1 | (3,3) | (1,2) | $\delta_{(1,2)1}=3+18=21$ | $e_{11}=21$ |
| 2 | (2,3) | (2,2) | $\delta_{(2,2)2}=11$ | No Change |
| 2 | (1,3) | (1,2) | $\delta_{(1,2)2}=21$ | No Change |
| 1 | (2,2) | (1,1) | $\delta_{(1,1)1}=14+6=20$ | $e_{21}=20$ |
| 2 | (2,2) | (2,1) | $\delta_{(2,1)2}=11$ | No Change |
| 1 | (1,2) | (3,1) | No Change | No Change |
| 2 | (1,2) | (1,1) | $\delta_{(1,1)2}=19$ | $e_{31}=24$ |

Table 1 (cont.)

| Algorithm Step | Change or Action |
|---|---|
| 4a | $\delta_{s(1,3)} = 21 + (-17) = 4$<br>$\delta_{s(1,1)} = 20 + (-20) = -1 =$ Minimum<br>$\delta_{s(2,1)} = 1 + (-10) = 1$ |
| 4b | $\delta_{s(1,1)}$ is a candidate |
| 5 | (2,1,1,2,)-(2,2,2,2,3)-(2,2,3,2,4)-(2,2,4,3,5) is the candidate path. |

Transfer costs at each location are also introduced at this time. Construction of the flight vs. location table as seen in the figure is also required. Once all of these ingredients are ready and the sink nodes selected, the application of the algorithm can begin.

Initialization, as in all the others, begins by setting all node label values at infinity. The super sink is initialized at zero. Next, the sink node labels are set and are never changed but must still be evaluated. Both labels of each node are set at the same value, that of the arc length from that sink node to the super sink. Once this has been done, the first node evaluated is determined as in the other algorithms. Beginning at node (3,5), the first node changed is node (1,4). The diagonal arc label at node (1,4) is now set at $\delta_{(1,4)1} = 21$, and the table entry for flight 4 from location 1 is set at $e_{41} = 21$, also. The next diagonal arc into node (3,5) originates at node (2,4) and is flight 2. Label changes are made $\delta_{(2,4)1} = 5+9 = 14$, and the corresponding table change made $e_{22} = 14$. No horizontal label or table changes are made since horizontal arcs have no significance at the sink nodes. These steps completed, the next node can be evaluated.

Node (2,5) has one diagonal arc along flight 1 from the sink node (3,4). Applying Step 1, no change is made since the labels are already at lower levels. At this point, note that no table entries exist for the destination location of the problem. The horizontal label remains at infinity since no shortest path yet exists. A major table column change is, however, in order. The minimum entry is $e_{22} = 14$. The others are thus changed to this minimum plus a transfer cost of 5. For this location column in the table, the entries now have values of $e_{12} = 19$, $e_{22} = 14$, $e_{32} = 19 = e_{42}$. Note, also, that by basing the evaluation system on flights only feasible paths are investigated.

After evaluating node (1,2) the evaluation of nodes can be terminated since no forward looking changes can be made from the first epoch of the window. The implementation of Step 4 is exactly comparable to Step 3 in the commodity path algorithm. This time the equations for the path to the super source look like this:

$$\delta_{SS_3} = \text{Minimum}\left\{\begin{array}{l} \text{Min}(\delta_{(1,1)1}, \delta_{(1,1)2}) + \alpha_{(1,1)3} = 20 + (-21) = -1 \\ \text{Min}(\delta_{(1,3)1}, \delta_{(1,3)2}) + \alpha_{(1,3)3} = 21 + (-17) = 4 \\ \text{Min}(\delta_{(2,1)1}, \delta_{(2,1)2}) + \alpha_{(2,1)3} = 11 + (-10) = 1 \end{array}\right\}$$

Thus, a candidate non basic path has been identified and by retracing the path through the network from node (1,1) it can be found.

The effect of transfer costs can be seen at node (2,3). The labels identify flight 3 as the shortest path leaving location 2, $\delta_{(2,3)1} = 11$. However, the table shows that to say on flight 2 is cheaper than incurring a transfer cost of 5, $e_{22} = 14$. The shortest path thus, stays on flight 2 instead of transferring to flight 3 as was done in the example of the commodity path algorithm without transfer costs.

To prove that the transfer cost algorithm also finds the shortest path through the network, it must be shown that the double labels at each node do indeed correspond to the shortest path to the super sink from that node. Those nodes connected directly to the super sink are certainly correctly labelled. Now assume that the double labels are correct for all nodes with time epoch components greater than that for a given node (i,t) (i.e. all nodes to which this node can lead). Also, the flight-location table entries are assumed correct. After traversing the horizontal arc leaving node (i,t), a path can select either a diagonal or horizontal arc from node (i,t+1). Clearly, then, the shortest path along this horizontal must be the minimum of both labels at node (i,t+1). This is exactly the same rule as used in the algorithm only viewed in the opposite order,

$$\delta_{(i,t)2} = \text{Minimum } (\delta_{(i,t+1)1}, \delta_{(i,t+1)2}).$$

With the horizontal label as now set, table entry changes must be evaluated. The entire column at location i must be checked to insure that no shortest path along a flight arriving at location i is longer than the absolute shortest path just set plus the transfer cost at location i. This insures that the diagonal arc labels will be shortest paths also.

Paths along diagonal arcs must be associated with flights when transfer costs are positive. This means that the arc length plus shortest path to the super sink, based on the table entry values at the time the destination node was evaluated, is the shortest path from node (i,t) along that flight arc. To find the diagonal arc label at (i,t), simply select the minimum of the diagonal arc shortest paths

$$\delta_{(i,t)1} = \underset{(j,s)\varepsilon\beta(i,t)}{\text{Minimum}} \{\ell_{(f,i,t,j,s)} + \delta_{(j,s)f}\},$$

where $\delta_{(j,s)f}$ is the shortest path table entry value at location j for arriving flight f at time epoch s. The table value for this equation is written in this form because the table value presently may be different. Only at node (j,s) could the table entry be guaranteed as the shortest path to the super sink. As the result of selecting the minimum of these shortest paths, the diagonal arc label, is the shortest path from (i,t) to the super sink.

Once each node has been evaluated, the path from super source to super sink is obtained in the same manner as in the commodity path algorithm. Select the minimum sum of the dual variable for the arc from super source to origin node plus the minimum of both labels at the origin node

$$\delta_{SS} = \underset{(i,t)\varepsilon\sigma_u}{\text{Minimum}} \{\alpha_{(i,t)u} + \text{Minimum}(\delta_{(i,t)1}, \delta_{(i,t)2})\},$$

where $\sigma_u$ is the set of origin nodes with destination u. The shortest path from super source to super sink is definitely obtained by applying the transfer cost algorithm.

# CHAPTER V

## CONCLUSIONS AND EXTENSIONS

The objective of this thesis is to formulate an exact model of the air freight problem and then to propose solution schemes, based on this model, for approaching the problem at a reduced, more manageable level. Of the heuristic procedures proposed, none introduces a major obstacle in efforts to maintain an optimality based solution procedure. The next step to be taken is the actual solution of the air freight problem. Further development of the solution approach proposed in this thesis is required.

The model formulated in Chapter II, while not being the only approach, does present a logical, intuitive unification of all the major components of the air freight problem. The arc-path formulation permits the scheduling and routing components to be combined into a single problem of selecting a single airplane flight from among all of the feasible combinations. Multi-commodity flow is also easily conceptualized as moving along paths from origin to destination. Most importantly, the construction of these airplane flights and commodity paths in a manner permitting commodity transfer and the resulting path structure complications is possible in the air freight model presented.

Inclusion of commodity transfer in the model formulation is a major addition to this area. Neither tanker scheduling, truck routing, nor air passenger routing literature had included this aspect. Complications do occur, as explained in Chapter III, when commodity transfer is permitted and continue to increase if transfer costs are considered positive.

A reduction of the problem solution approach to a more manageable size is proposed in several logical steps. First, the ability to fix the fleet size and aircraft mix is assumed to be a reasonable approach. Knowledgeable sources and other operational constraints can limit the feasible combinations to the point that within a few iterations of the fixed fleet concept, a best solution can be selected.

Once the fleet is fixed, the second step looks at smaller pieces of the time space network. Here, the time window concept is introduced and becomes the lower limit on problem size since commodity flow feasibility within time restrictions must be maintained. A repetitive procedure for utilizing the time window is presented that hopefully loses very little to an overall optimal view of the system, since the commodity time in the system is the binding constraint in the model in this approach.

Finally, a linear relaxation approach to the mixed integer problem at each time window is proposed to allow further simplification of the still complex problem. This relaxation approach utilizes an optimal basis. Since it is a linear programming method, round off procedures are used to return to the mixed integer solution required.

A column generation approach to the linear relaxation concept allows the implicit evaluation of all the nonbasic columns as long as the column costs can be associated with the individual arcs in the column. If this is possible, then a criteria for an entering column into the basis can be determined and an optimal solution achieved. Shortest path procedures provide efficient methods of identifying candidate columns to enter the basis.

The major complication to applying this in the air freight problem appears in the commodity transfer cost. It is not allocable to arcs. When transfer costs are combined with the other commodity costs and no longer considered separate, the commodity path with transfers can be adapted to the column generation approach.

Another factor that must be recognized in the same column generation approach is that even though plane flights and commodity paths are included in the same problem, they must be handled separately. In addition, the only way new flights are generated is by commodity flow changes and demands. This required the problem to be solved over an open graph in order to insure that commodities are not limited to existing basic plane flight arcs.

Finally, modifications to the general shortest path procedure for use in the column generation approach are presented. An algorithm for commodity paths, in which transfer costs are not separate, and an algorithm for airplane flights are developed. In addition, an algorithm to handle commodity paths with identifiable transfer costs illustrates the complications involved as a result of transfer costs not being able to be associated with arc costs. The bookkeeping aspect of the algorithm expands and the number of feasible flights in the network contracts.

Pursuing the solution procedure proposed to a final answer will require additional work. Only approaches to various parts of the solution procedure are presented. Deeper involvement with the algorithms, column generation approach, and time window concept is required. For example, if the same destination node group is to be reused in the algorithm after finding a candidate path, a method of updating the network might be developed instead of starting over from Step 0. Even the iterative process of selecting the destination node group must be investigated. Rules for switching between a commodity path algorithm and a plane flight algorithm are needed to take advantage of the characteristics of each process. If transfer costs must be considered, the decision of how many and which flights to be included in the closed graph can have a definite affect on the answer and length of time to reach that answer.

The time window problems must be answered as well.

Retaining as much information as possible between windows might reduce start up time in computations. The time spent at any one window might be limited to prevent getting delayed at any one point in the solution. In the same context, the accuracy of the solution is a question that must be answered early in the design of the solution procedure. Time trade-offs might generate a good solution in less time.

Even once an answer for a fixed fleet is obtained, a form of gradient search approach locally applied might be developed in looking for the final fleet size and mix of aircraft types.

# APPENDIX

## NOMENCLATURE

$A_a$ cargo capacity of aircraft type a

$C_f$ total cost of flight f

$F_{(a_f)}$ fix acquisition cost of aircraft type a used on flight f

$H_i$ unit transfer cost at location i

$I_p$ unit inventory holding cost of commodity path p:

$$\underset{(f,i,t,j,s)\varepsilon\Gamma_p}{\text{Maximum}} \{s\} - \underset{(f,i,t,j,s)\varepsilon\Gamma_p}{\text{Minimum}} \{t\}$$

$K_{(a_f)}$ unit operations cost of aircraft type a used on flight f

$L_{j(a_f)}$ landing cost at location j for aircraft type a used on flight f

$M$ number of locations in the system

$N_a$ number of aircraft of type a operating flights

$N_{a(i,t)}$ number of aircraft of type a operating flights that include node (i,t)

$Q_{(i,t)u}$ amount of commodity with o-d designation (i,t)u

$Q'_{(i,t)u}$ amount of commodity with o-d designation (i,t)u that cannot use a path crossing the window boundary

$R_p$ total unit cost of commodity path p:

$$I_p + \sum_{\substack{\{(o,i,t,i,t+1)\varepsilon\Gamma_p:\text{ there exists} \\ (f,j,s,i,t)\varepsilon\Gamma_p \text{ with } f > 0\}}} H_i$$

$S_u$ super sink connected by arcs from sink nodes u

$SS_u$ super source connected by arcs from it to origin nodes of commodities with destination u

T period length of repeating flight cycles

W width in time epochs of time window

a aircraft type used in the fleet

$a_f$ aircraft type used on flight f

b right hand side values in the linear relaxation basis

$c_{(f,i,t,j,s)}$ or $c_{(a,i,t,j,s)}$ variable cost associated with the arc on flight f or aircraft type a from node (i,t) to node (j,s)

$c_{(u,t)S_u}$ arc cost associated with the sink arc from sink node (u,t) to super sink $S_u$

$e_{fi}$ table entry for flight f at location i: The shortest path to the super sink for flight f arriving at location i

f a flight of a single aircraft type connecting locations in a continuous path from time 1 to T

$g_{(i,t),(k,r)}$ or $g_{(i,t)u}$ an artificial variable for the commodity flow

(i,t) node at location i at epoch t

(i,t,j,s) arc from node (i,t) to node (j,s)

(f,i,t,j,s) flight f on arc from node (i,t) to (j,s)

$\ell_{(f,i,t,j,s)}$ or $\ell_{(a,i,t,j,s)}$ length of the arc on flight f or aircraft type a from node (i,t) to node (j,s)

$n_{a,(i,t)}$ an artificial variable for the airplane flight flow constraint in the linear relaxation formulation

$p$ — a commodity path connecting an origin node $(i,t)\varepsilon\Omega$ with a destination $u\varepsilon\Delta$, with a set of arcs defined in $\Gamma_p$ that does not exceed the time constraint on commodity movement

$q_{(i,t),S_u}$ — amount of commodity carried from the sink node $(i,t)$ to the super sink $S_u$

$u$ — group of destination or sink nodes

$v$ — time window boundary time epoch value

$w_p$ — amount of commodity carried on commodity path P

$x_{(f,i,t,j,s)}$ — a slack variable for the diagonal arc capacity constraint in the linear relaxation formulation

$y_f$ — 1 if flight f, consisting of the set $\Lambda_f$, is used

0 otherwise

$z$ — objective function value

$\Delta_{(j,s)}$ — set of all commodity paths ending at node $(j,s)$

$\Delta'(k,r)$ — set of all commodity paths ending at intermediate node $(k,r)$:

$$p\text{: } (o,k,r,k,r+1)\varepsilon\Gamma_p \text{ and } r = v$$
$$\text{or } (f,k,r+1,j,s)\varepsilon\Gamma_p \text{ with } f \neq 0,\ k \neq j,$$
$$\text{and } r+1 \leq v < s$$

$\Gamma_p$ — set of arcs comprising the commodity path p:

$\{(f,i,t,j,s)$: if $f > 0$, path p takes flight f along arc $(i,t,js)$;

if $f = 0$, then $i = j$ and path p holds at location i in ground storage.$\}$

$\Lambda_f$ set of arcs comprising the flight path for flight f:

$$\{(i,t,j,s): (i,t,j,s) \text{ is part of flight } f\}$$

$\tau_{(i,t)}$ set of flight arcs crossing the window boundary such that

$$f:\ (i,v,i,v+1)\varepsilon\Lambda_f \text{ or}$$
$$(i,t+1,j,s)\varepsilon\Lambda_f \text{ with } i \neq j$$
$$\text{and } t+1 \leq v < s$$

$\Omega_{(i,t)}$ set of all commodity paths originating at node (i,t)

$\alpha_{(i,t)u}$ longest basic path for commodity with o-d designation ((i,t),u)

$\beta_{(i,t)}$ set of all destination nodes for diagonal arcs departing node (i,t)

$\delta_{(i,t)1}$ shortest path to the super sink from node (i,t) by departing on a diagonal arc

$\delta_{(i,t)2}$ shortest path to the super sink from the node (i,t) by departing on a horizontal arc

$\delta_{SS_u}$ or $\delta_{SS_a}$ shortest path from the super source to the super sink using destination nodes from group u or aircraft type a

$\eta_{a,(i,t)}$ length of basic flight f terminating at node (i,t)

$\pi_{(i,t,j,s)}$ simplex multiplier for arc (i,t,j,s)

$\phi_{(j,s)}$ set of all departure nodes for diagonal arcs arriving at node (j,s)

$\sigma_u$ set of all origin nodes for commodities destined for location u

# BIBLIOGRAPHY

1. Applegren, L. H., "A Column Generating Algorithm for a Ship Scheduling Problem," Transportation Science 3(1), 53-68, 1969.

2. Bellmore, M., G. Bennington and S. Lubore, "A Maximum Utility Solution to a Vehicle Constrained Tanker Scheduling Problem," Naval Research Logistics Quarterly 15(3), 403-411, 1968.

3. __________, "Further Extensions of the Tanker Scheduling Problem," The Mitre Corporation, MTP-339, October, 1969.

4. __________, "A Multivehicle Tanker Scheduling Problem," Transportation Science 5(1), 36-47, 1971.

5. Bellmore, M., J. C. Liebman, and D. H. Marks, "An Extension of the (Swarc) Truck Assignment Problem," Naval Research Logistics Quarterly 19(1), 91-99, 1972.

6. Briskin, L. F., "Selecting Delivery Dates in the Tanker Scheduling Problem," Management Science 12(6), B224-B235, 1966.

7. Burger, P. and D. B. Rice, "Integer Programming Models of Transportation Systems: An Airline Systems Example," Krannert Graduate School, Purdue University, Institute Paper No. 133, Lafayette, Indiana, April, 1966.

8. Clarke, G. and J. W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," Operations Research 12(4), 568-581, 1964.

9. Dantzig, G. B. and A. R. Ferguson, "The Allocation of Aircraft to Routes--An Example of Linear Programming under Uncertain Demand," The Rand Corporation, P-727, December, 1956.

10. Dantzig, G. B. and D. R. Fulkerson, "Minimizing the Number of Tankers to Meet a Fixed Schedule," Naval Logistics Research Quarterly 1, 217-222, 1954.

11. Danzig, G. B. and D. L. Johnson, "Maximum Payloads per Unit Time Delivered through an Air Network," Operations Research 12(2), 230-236, 1964.

12. Demmy, W. S. and Maj. K. E. Brant, U. S. Air Force, "Design of a Military Air Cargo Transportation System by Use of a Large Scale Mathematical Programming Model," presented at 41st ORSA Meeting, New Orleans, April, 1972.

13. Etschmaier, M. M. and M. Rothstein, "Operations Research in the Management of Airlines," Omega 2(2), 157-159, 1974.

14. Ferland, J. A., "Minimum Cost Multicommodity Circulation Problem with Convex Arc Costs," Transportation Science 8(4), 355-360, 1974.

15. Fetter, R. B. and R. C. Steorts, "A Model for the Design and Evaluation of Air Cargo Systems," The Rand Corporation, RM-4801-PR, October, 1966.

16. Ford, L. R. and D. R. Fulkerson, "A Suggested Computation for Maximal Multicommodity Network Flows," Management Science 5, 97-101, 1958.

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

18. Garfinkel, R. S. and G. L. Nemhauser, Integer Programming, John Wiley and Sons, 1972.

19. Gavish, B. and P. Schweitzer, "An Algorithm for Combining Truck Trips," Transportation Science 8(1), 13-23, 1974.

20. Geoffrion, A. M., "An Improved Implicit Enumeration Approach for Integer Programming," Operations Research 17, 437-454, 1969.

21. Gillett, B. E. and Leland R. Miller, "A Heuristic Algorithm for the Vehicle-Dispatch Problem," Operations Research 22(2), 340-349, 1974.

22. Gould, F. J., "A Linear Programming Model for Cargo Movement Evaluation," Transportation Science 5(4), 344-365, 1971.

23. Jarvis, J. J., "On the Equivalence Between the Node Arc and Arc Chain Formulations for the Multicommodity Maximal Flow Problem," Naval Research Logistics Quarterly 16(4), 525-529, 1969.

24. Krolack, P., W. Felts, and J. Nelson, "A Man-Machine Approach Toward Solving the Generalized Truck Dispatching Problem," Transportation Science 6(2), 149-170, 1962.

25. Lam, T. N., "Comments on a Heuristic Algorithm for the Multiple Terminal Delivery Problem," Transportation Science 4(4), 403-405, 1970.

26. Leblanc, L. J., "An Algorithm for the Discrete Network Design Problem," Transportation Science 9(3), 183-199, 1975.

27. Levin, A., "Some Fleet Routing and Scheduling Problems for Air Transportation Systems," Flight Transportation Laboratory Report R-68-5, M.I.T., 1969.

28. _________, "Scheduling and Fleet Routing Models for Transportation Systems," Transportation Science 3(5), 232-255, 1971.

29. Martin-Lof, A., "A Branch-and-Bound Algorithm for Determining the Minimal Fleet Size of a Transportation System," Transportation Science 4(2), 159-163, 1970.

30. McKay, M. D. and H. O. Hartley, "Computerized Scheduling of Seagoing Tankers," Naval Research Logistics Quarterly 21(2), 255-264, 1974.

31. Midler, J. L., "A Stochastic Multiperiod Multimode Transportation Model," Transportation Science 3(1), 8-29, 1969.

32. Mole, R. H., "Dynamic Optimization of Vehicle Fleet Size," Operations Research Quarterly 26(1), 25-34, 1975.

33. Peters, H. J., "Routing of Aircraft and Economics of Flight Operations," Ph.D. Thesis, Technical University of Stuttgart, 1974.

34. Rao, M. R. and S. Zionts, "Allocation of Transportation Units to Alternative Trips--A Column Generation Scheme with Out of Kilter Subproblems, Operations Research 16(1), 52-63, 1968.

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

36. Salzborn, F. J., "A Note on Fleet Routing Models for Transportation Systems," Transportation Science 6(3), 335-337, 1972.

37. Schwartz, N. L., "Discrete Programs for Moving Known Cargos from Origins to Destinations on Time at Minimum Bargeline Fleet Cost," Transportation Science 2(2), 134-145, 1968.

38. Simpson, R. W., "Computerized Schedule Construction for an Airline Transportation System," Flight Transportation Laboratory Report FT-66-3, M.I.T., December, 1966.

39. Swarc, W., "The Truck Assignment Problem," Naval Research Logistics Quarterly 14(4), 529-557, 1967.

40. Tapiero, C. S. and M. A. Soliman, "Multi-Commodities Transportation Schedules Over Time," Networks 2, 311-327, 1972.

41. Tillman, F. A., "The Multiple Terminal Delivery Problem with Probabilistic Demands," Transportation Science 3(3), 192-204, 1969.

42. Tomlin, J. A., "Minimum-Cost Multicommodity Network Flows," Operations Research 14, 45-51, 1966.

43. Turner, W., P. Ghare, and L. Fourds, "Transportation Routing Problem--A Survey," AIIE Transactions 6(4), 288-300, 1964.

44. Wollmer, R. D., "Multicommodity Networks with Resource Constraints: The Generalized Multicommodity Flow Problem," Networks 1(3), 245-263, 1972.