

FLIGHT TRANSPORTATION LABORATORY
REPORT R 72-3

ROUTE NETWORK IMPROVEMENT
IN AIR TRANSPORTATION
SCHEDULE PLANNING

YU-PO CHAN

MAY 1972

MIT

DEPARTMENT
OF
AERONAUTICS
&
ASTRONAUTICS

FLIGHT TRANSPORTATION
LABORATORY
Cambridge, Mass. 02139

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Flight Transportation Laboratory

Report FTL-R72-3

ROUTE NETWORK IMPROVEMENT IN
AIR TRANSPORTATION SCHEDULE PLANNING

FTL COPY, DON'T REMOVE
33-412, MIT 02139

Yu-po Chan

June 1972

ABSTRACT

One of the routing and scheduling problems faced by an airline is to configure a route network. It seeks to answer the following two questions: First, should scheduled service be provided for a city pair market? Second, if market entry is warranted, should the city pair be served by a non-stop, multi-stop, or connect routing? A profit-maximizing airline, in trying to answer these questions, has to abide by the route regulations imposed by the Civil Aeronautics Board. The airline has to take into account the intercarrier route competition. It has to recognize that its share of the passenger demand is a function of the level of service offered, and that passengers usually want to reach their destination in the most convenient routing for themselves.

An optimization model is formulated for the route network configuration problem. Because of the huge combinatorial dimensionality inherent in the problem, a special solution method has to be devised. Only a handful of the most promising, feasible route candidates are identified at a time. An optimal choice is immediately made out of the few candidates. These route candidates are generated "as needed" by graph theoretic schemes, while route selection is performed by solving an integer program characterized by an ill-behaved objective function. At each generation/selection step, route network improvement is made by the optimal selection of the route candidate (i) to add to an existing network, (ii) to replace an unprofitable route, or simply (iii) to be deleted from the route network.

The solution algorithm is based on the method of successive approximation in dynamic programming. Primal feasibility is maintained at all times. If the algorithm is stopped prematurely, due to limited computational resources, an improved (but not necessarily optimal) solution is always available.

A 40-routine computer software package for the algorithm has been developed. It was successfully used to analyze a case study from American Airlines. Our limited computational experience showed that execution time is at least seven times faster than a comparable algorithm.

ACKNOWLEDGMENTS

The author would like to thank Professor Robert Simpson for his guidance and supervision of this dissertation. It was through Professor Simpson that the author was introduced into the stimulating field of air transportation. Furthermore, the author is particularly appreciative of the support from the Flight Transportation Laboratory which provided him with the research freedom which is quite a luxury in view of the current sluggish economy.

Thanks are due to the other members of the thesis committee, Professors David Marks, Marvin Manheim, and Wayne Pecknold. Professors Marks and Pecknold have offered valuable suggestions when they patiently waded through draft versions of the thesis.

The following sponsors have made this piece of research possible: the Slater Funds for Flight Transportation and the NASA research grant -- Joint University Research Program for Air Transportation Needs (NGL 22-009-640). A fellowship from the M.I.T.-Harvard Joint Center for Urban Studies provided for part of the computer expenses.

Without the professional editing and typing help of Miss Terry Rothra and Mrs. Joan Bonney, this thesis might never have become a reality.

TABLE OF CONTENTS

TITLE PAGE	1
ABSTRACT	2
ACKNOWLEDGMENTS	4
TABLE OF CONTENTS	5
LIST OF TABLES	10
LIST OF FIGURES	11
CHAPTER 1 - ROUTE IMPROVEMENT IN AN AIRLINE NETWORK	14
1.1 Routing and Scheduling in an Airline Network	15
1.1.1 A Supply/Demand Categorization	16
1.1.2 A Hierarchical Categorization	18
1.2 Problem Definition	21
1.2.1 The Problem of Route Network Configuration	21
1.2.2 An Optimization Formulation of the Problem	28
1.3 Methodologies for Network Routing Analysis	35
1.3.1 Network Synthesis Techniques	37
1.3.2 Network Evaluation Techniques	39
1.3.3 Network Improvement Techniques	46
1.3.4 Route Improvement, Synthesis and Evaluation -- R.I.S.E.	52
1.4 The R.I.S.E. Model in the Schedule Planning Process	56
1.4.1 Sequential Coordination	57
1.4.2 Simultaneous Coordination	60
1.4.3 A Simul-Sequential Approach	61

1.5	Summary	72
	References	76
CHAPTER 2 - MODELLING THE AIRLINE FIRM AND INDUSTRY		79
2.1	The Airline Industry	80
2.1.1	Regulatory Environment	81
2.1.2	Competition Environment	85
2.1.3	Passenger Travel Behavior	88
2.2	The Airline Firm	90
2.2.1	The Firm and Its Market	91
2.2.2	Short-Run vs. Long-Run Analysis	96
2.2.3	Cost, Revenue and Profit	99
2.2.4	Marginal Profit Analysis	109
2.3	The R.I.S.E. Model	111
2.3.1	Regulatory and Competition Constraints	112
2.3.2	Profit Maximization and Travel Time Minimization	114
2.3.3	Solution Strategy	115
2.4	The Application of R.I.S.E. in Management and Public Sector Planning	116
	References	120
CHAPTER 3 - MATHEMATICAL MODEL FORMULATION OF ROUTE IMPROVEMENT		123
3.1	Route Generation: A Graph-Theoretic Approach	123
3.1.1	Configuring a Route Network	124
3.1.2	C.A.B. Route Authority	129
3.1.3	A Graph-Theoretic Representation of Route Authority	134
3.1.4	Generating Route Candidates via Graph-Theoretic	

Techniques	143
3.2 Route Selection: An Integer Programming Approach	151
3.2.1 Selecting Among Route Candidates	151
3.2.2 Constraints of the Integer Program	152
3.2.3 Objective Function of the Integer Program	174
3.2.4 Discussion on the Route Selection Integer Program	186
3.3 Generation and Selection: A Decomposition Approach	193
3.3.1 Simultaneous Route Generation and Selection	193
3.3.2 The Primal Decomposition Formulation	194
3.3.3 Route Network Configuration via Primal Decomposition	222
3.4 Summary	231
References	233
CHAPTER 4 - A SOLUTION ALGORITHM FOR THE MODEL	235
4.1 A Dynamic Programming Solution Scheme	235
4.2 Route/Routing Synthesis	238
4.2.1 A State-Stage Space Diagram	238
4.2.2 Synthesizing Routes/Routings in the Diagram	240
4.3 Route/Routing Evaluation	243
4.3.1 Passenger Flow Evaluation	246
4.3.2 Tabulation of Cost and Revenue	252
4.3.3 Connectivity Evaluation	254
4.4 Route/Routing Improvement	255
4.4.1 Marginal Profit Computation	255
4.4.2 Labelling the State-Stage Diagram	258

4.4.3	Phase I vs. Phase II of the Algorithm	263
4.5	Route Improvement, Synthesis and Evaluation	266
4.5.1	An Example	266
4.5.2	The Solution Algorithm	270
4.5.3	Some Computational Aspects	282
4.6	Theoretical Generalization	287
	References	296
CHAPTER 5 - A CASE STUDY AND SOME COMPUTATIONAL EXPERIENCES		298
5.1	The Status of Software Development	298
5.1.1	The Computer/Software System	299
5.1.2	The Implemented Part of the Model	300
5.2	A Case Study	302
5.2.1	Definition of the Case Study	302
5.2.2	Analysis Results	303
5.2.3	Conclusive Comments	311
5.3	Other Computational Experiences	312
	References	316
CHAPTER 6 - SUMMARY, CONCLUSIONS AND RECOMMENDATIONS		317
6.1	Modelling the Route Network Improvement Problem	317
6.2	Optimization Techniques for Problem Solution	319
6.3	Results and Findings	323
6.4	Research Status and Recommended Extensions	326
	References	328
BIOGRAPHICAL NOTE		329

APPENDIX A1 - WESTERN AIRLINES ROUTE CERTIFICATE 331

LIST OF TABLES

2*2*1	Annual System Expenses - Domestic Carriers (1962-1966)	103
2*2*2	A Percentage Breakdown of Airline Expenses	108
5*2*1	System Summary Statistics	313
5*3*1	Approximate Program Running Times	314

LIST OF FIGURES

1-2-1	Route Network Configuration	22
1-2-2	Route Network Configuration 2	26
1-2-3	Relationship between Route Frequency and Segment Traffic . . .	27
1-2-4	Integer Programming Tableau for Route Selection	31
1-4-1	Simul-Sequential Coordination	63
1-4-2	Simul-Sequential Approach to Schedule Planning	65
2-1-1	Number of Non-Stop Routes by Origin City	84
3-1-1	A Network and Its Adjacency Matrix	125
3-1-2	Examples of Segments and Subsegments	130
3-1-3	Cities in the Route Certificate	132
3-1-4	Venn Diagram Representation	133
3-1-5	Subsegment Graph and Its Contiguity Matrix	136
3-1-6	The Intersection of Two Subsegments and the Corresponding Contiguity Matrix	138
3-1-7	Splitting a Subsegment into Two	139
3-1-8	Split Subsegments and Segments	140
3-1-9	"Must Originate or Terminate" Restriction	142
3-1-10	The "Complete" Contiguity Matrix for the Route Certificate . .	144
3-1-11	Route Certificate Consisting of 2 Subsegments	145
3-2-1	Example Route Network	153
3-2-2	Integer Programming Tableau	154
3-2-3	Differences between a Route Network and a Highway Map	173
3-2-4	O-D Demand Function	176

3-2-5	Definition of p_{pq}	177
3-2-6	Original Integer Programming Formulation	187
3-3-1	The Original Form of the Integer Program	195
3-3-2	The Decomposable Form	196
3-3-3	The Decomposable Formulation	197
3-3-4	Node-Arc vs. Arc-Chain Formulation	202
3-3-5	Arc-Chain Formulation	204
3-3-6	Example Network	207
3-3-7	The Master Program	212
3-3-8	3-City Network	225
4-2-1	An Example State-Stage Diagram	239
4-2-2	Route/Routings in the State-Stage Diagram	241
4-2-3	A 2-Stop Route in the State-Stage Diagram	242
4-2-4	A 2-Stop Connect Routing	244
4-2-5	A Connect Routing Synthesized from a Route and a Routing . . .	245
4-3-1	Passenger Demand Function as Represented in the State-Stage Diagram	247
4-3-2	Traffic Flow Pattern as Represented in the State-Stage Diagram	248
4-3-3	The Composition of Segment Traffic in Terms of Local and Through Traffic	250
4-3-4	A Table to Bookkeep the Basic Unit Costs and Revenues	253
4-4-1	Labelling the State-Stage Diagram	259
4-4-2	Bookkeeping a Route/Connect-Routing	261
4-4-3	The Result of Re-Labelling and Re-Tabulating	262
4-4-4	Penalty Method to Obtain an Initial Feasible Solution	264

4-5-1	Passenger Demand Function	267
4-5-2	Substage 0.1	272
4-5-3	Substages 0.2 and 0.3	273
4-5-4	Substage 1.1	275
4-5-4a	Route Network at the End of Substage 1.1	276
4-5-5	Substage 1.2	278
4-5-6	Substage 1.3	279
4-6-1	Flow Diagram of R.I.S.E.	294
5-2-1	RISE-I Route Network	304
5-2-2	AA Route Network	305
5-2-3	Differences between RISE-I and the AA Route Network	307
5-2-4	A Comparison of the No. of City Pairs Served by Each Level of Service	309

CHAPTER 1

ROUTE IMPROVEMENT IN AN AIRLINE NETWORK

Domestic passenger air transportation in the United States is provided by a regulated industry made up mainly of eleven competing trunk carriers. Each carrier offers scheduled service to accommodate and promote the traveling public. Scheduled service is provided as a result of careful routing and scheduling analyses. Currently, the routing and scheduling analysis becomes even more critical. This is due to two factors. First, passenger traffic slumped from a 15% growth in the Sixties to 1.4% in 1970. Second, there has been a quantum increase in fleet capacity from the introduction of wide-body aircrafts such as the B-747's and DC-10's. An exacting scheduling process, among other measures, is required by each airline to face up to the sluggish traffic in an inter-carrier competitive environment.

In this dissertation, an analysis package is put forth for use in a profit-oriented carrier for the following routing and scheduling questions. From the competition and traffic potential point of view, (i) should the carrier enter into a C.A.B.* authorized market in providing service between the city pair; (ii) if market entry is warranted, should the city pair be served by a non-stop, multi-stop route, connecting service, or certain combinations of the three?

*The C.A.B. is the United States Civil Aeronautics Board, delegated with the power of economic regulation in air transportation.

This chapter, and, to a certain extent, Chapter 2, are self-contained overviews of the dissertation. Chapter 2, with its economic flavor, serves as a transition from the general discussions of Chapter 1 to the formal model building, solution and verification in the chapters that follow. Mathematical technicalities are kept to a minimum in the first two chapters. They are specifically written for a large number of leisure-deprived audience members. Those blessed with the inquiring mind will, at the same time, find the overview an informative guide to in-depth reading of subsequent chapters.

1.1 Routing and Scheduling in an Airline Network

The problem tackled in this piece of research has been identified as belonging to a class of transportation analysis techniques called network routing and scheduling. There is a large number of issues in routing and scheduling. These issues, or subproblems, will be categorized in this section. The categorization helps to identify our problem within the general framework.

Two categorization schemes will be used. First, we classify according to supply models vs. demand models. Second, we classify according to a priority hierarchy -- typical of a decision process in an organization. The classification scheme adopted here is tailored for introducing the analysis approach of our work. It is by no means the only way to break down the routing and scheduling problem. Simpson [1969] has another informative categorization which tends to be based on solution techniques. The author

acknowledges the benefits of his work.

1.1.1 A Supply/Demand Categorization

Transportation analysts today like to think in terms of the economic conceptual framework of supply and demand. Routing and scheduling models lend themselves readily to the supply vs. demand functional classification.

Passenger demand models are those that forecast travel potentials between origin and destination pairs.* They usually employ econometric techniques. The sophisticated ones relate demand as a function of the level of service. That is to say, the demand function traces out how travel potential varies in response to service improvement or degradation.

Supply models are those that help to provide scheduled service in the airline network. The main bulk of routing and scheduling models are supply models.** The following are typical examples:

- route network configuration
- aircraft/fleet assignment
- frequency determination
- timetable construction/aircraft dispatching
- fleet routing and fleet size determination
- crew scheduling
- passenger routing

*For a bibliography on demand models, see Vandersypen [1971].

**Again, for a bibliography on routing and scheduling models, see Vandersypen [1971].

Again, sophisticated versions of the above supply models address the level of service feedback loop between supply and demand. That is to say, a demand function is used in the model formulation whereby origin-destination travel potential changes as the scheduled service characteristics change.

In common practice, models are constructed such that they handle only the demand or the supply side, holding an exogenously fixed demand or supply pattern. Such models may be quite serviceable as an aid to decision-making. They yield, nevertheless, non-equilibrium solutions from a scientific point of view.**

Ideally, a level of service feedback loop between supply and demand should be built into the routing and scheduling process.* Models that recognize such intertwining relationships would yield equilibrium solutions. Viewed as a network flow problem, the procedure of equilibrium computation can be envisaged conceptually as coordinating aircraft routings, crew routings and passenger routings so that the appropriate number of aircrafts, and crew would service the potential traffic at the suitable segments and at the opportune time. All these are done so as to optimize a figure of merit, which may be profit to the carrier, or public convenience and necessity (some measure of welfare) to the Civil Aeronautics Board (C.A.B.).

The equilibrium solution(s) is (are) mainly manifested in two network

*See the "Schedule Planning Process" figure on page two of Simpson [1969], for example.

**The only case where they are equilibrium solutions would be when the "real world" demand/supply pattern is fixed.

characteristics: (i) a schedule(s) defined by a certain aircraft and crew routing pattern(s) in the network,* and (ii) a traffic flow(s) defined by a distribution of passenger routings in the network.*

Non-equilibrium models are more numerous than equilibrium models, simply because the state of the art is such that analytical techniques are by no means adequately developed for network equilibrium analysis.** The routing problem addressed in this dissertation pays particular attention to the level of service feedback loop. It is one of the exploratory researches in the direction toward a network equilibrium analysis.

1.1.2 A Hierarchical Categorization

The last section groups scheduling models in a supply vs. demand classification. This present section will group them into a hierarchical categorization.

Viewing an organizational structure from a hierarchical, or multi-level, standpoint is a very intuitively appealing concept. In the airline corporate structure, ranks go from "chiefs" to "indians" (probably with the "medicine man" in between). The schedule planning decisions are broken up accordingly. Inherent in the hierarchy is the sequential nature of the

*The term 'network' used in this context has a temporal dimension in addition to the usual geographical connotation. A more proper word for such a kind of network is the "schedule map" (coined by Leven [1969]).

**For a bibliography of models that address the whole schedule planning process, see Vandersypen [1971] once again.

"chain of commands" in problem-solving -- that the "chiefs" will handle the higher priority routing and scheduling decisions and the "indians" the lower ones. The output of the chief's office serves as input to the indian's desk. Furthermore, parallel to the multi-level concept is the economic idea of long-run vs. short-run decisions. The higher priority decisions are equated with a longer planning horizon, while the lower priority decisions are generally construed as more myopic.

In the following discussion, we will give typical examples of long-run vs. short-run vs. real time scheduling models. Included in the long run are policy/corporate issues such as:

- charting legislative, regulatory (such as routes and fares) policies
- forecasting economic activities and travel demand
- planning multi-period fleet acquisition and the associated financial investments.

Included in the short run category are the familiar routing and scheduling models that produce a cyclic (e.g., monthly) schedule:

- aircraft/fleet assignment
- frequency determination
- timetable construction/aircraft dispatching
- fleet routing and fleet size determination
- crew scheduling
- passenger routing.

Finally, there are the real time models that tackle "on line" schedul-

ing decisions. The typical examples in this category are "dynamic scheduling" and "schedule control."* Another name for dynamic scheduling is probably demand responsive scheduling, such as the Eastern Airlines shuttle service and the "dial-a-bus" or taxi-cab operation. Schedule control, on the other hand, refers to the contingency measures taken to replace a mechanically disabled aircraft scheduled for a flight or to remedy the situation when several flights are cancelled due to unexpected bad weather.

In spite of the organizational "chain of command," the different levels in the hierarchy do interact with each other. This is particularly true in finalizing a schedule plan. A typical example is the interaction between the crew scheduling and aircraft maintenance requirement teams with the aircraft/fleet/frequency assignment and timetable construction teams. The feedback process will be termed the "schedule refinement loop."** Such a loop is presently done manually, for good reasons: because the state of the art in modelling is by no means near the degree of sophistication achieved by experienced personnel.

The route network configuration problem addressed in this dissertation is neither a long-run nor a short-run model. It is most appropriately called a medium-range planning problem. It takes the current route regulatory constraints (e.g., route authorities and fare) as given and suggests a rational route network structure for use in the short-run scheduling models.

*Both terms are coined by Simpson [1969], the author believes.

**The idea is borrowed from Simpson [1969], who calls a similar iteration the "schedule evaluation loop."

Although the schedule refinement loop is not built into our problem formulation explicitly, provisions are made for the feedback interaction. We will come back to this looping discussion at the end of the chapter (section 1.4.3).

1.2 Problem Definition

The last section broke up the routing and scheduling problem into classifications, which allowed us to identify the role our route network configuration problem plays in the general framework. It has been pointed out that the route network configuration model yields equilibrium solutions, since the level of service feedback loop is built into the formulation. The routing model is used for medium-range planning purposes. It interfaces the long-run regulatory/economic policies with the short-run scheduling operations.

The purpose of this present section is two-fold. First, we will outline a formal definition of the route network configuration problem. Second, a schematic optimization formulation will be offered to model the problem.

1.2.1 The Problem of Route Network Configuration

Perhaps a good way to introduce our route network configuration problem is via a graphic example. In Fig. 1-2-1 is shown a five-city map: Chicago (CHI); Washington, D.C. (WAS); New York City (NYC); Dayton, Ohio (DAY) and Columbus, Ohio (CMH). Two of the origin-destination (O-D) travel demand potentials are shown: from NYC to CHI, and from CMH to WAS. Sup-

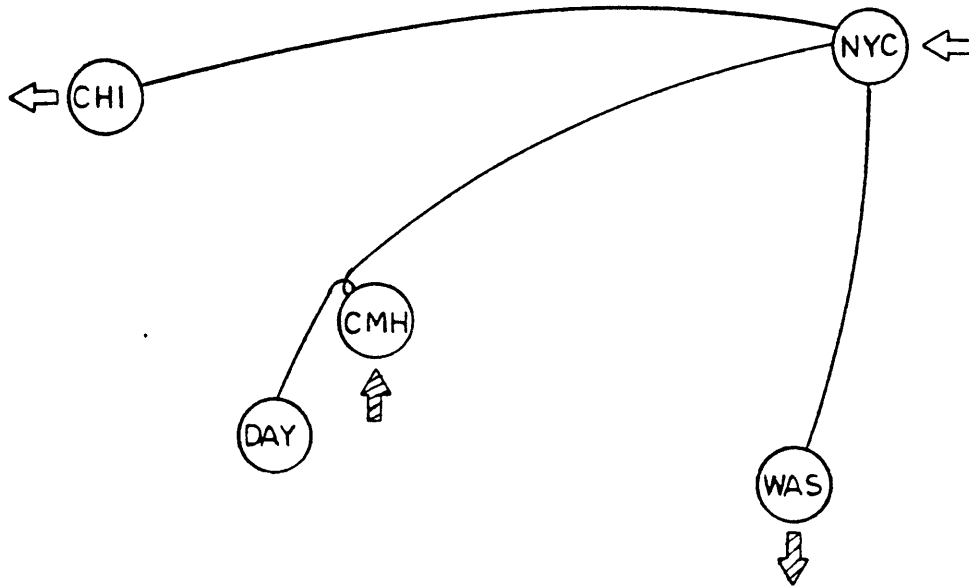


FIG. 1-2-1 ROUTE NETWORK CONFIGURATION

pose for the time being the airline under consideration has the necessary equipment and personnel to provide scheduled service between these five cities. Our problem is this: How should aircraft routes be configured in this map to facilitate the passenger routings from their origins to their destinations? This is question number one.

In the same diagram (Fig. 1-2-1) is drawn a particular route network configuration. This route network consists of two non-stops: (i) between CHI and NYC, (ii) between NYC and WAS; and one one-stop: between NYC, CMH and DAY.* In this route network, passengers from NYC to CHI can simply take a non-stop routing on the non-stop route between the city pair. Passengers who want to go from CMH to WAS, however, have to make a connect routing via NYC, patronizing one leg of the one-stop DAY-CMH-NYC and then the non-stop route NYC-WAS. It is part of our concern in this dissertation to answer the following question: Is this particular route network and passenger routing pattern the most desirable configuration, or can we improve on it? This is question number two.

In order to answer the above two questions, considerations have to be given to the context of the route network configuration problem. In our analysis, we have identified four issues that a route planner should keep in mind. First, route network configuration is subject to the route regulation of the Civil Aeronautics Board (C.A.B.). An airline cannot serve a city pair market unless it has been authorized a Certificate of Public Con-

*Notice that while this is a one-stop route for NYC-DAY, it is a non-stop route for NYC-CMH and CMH-DAY.

venience and Necessity (usually referred to as the route certificate). The route certificate spells out specifically whether the city pair can be served by a non-stop or multi-stop, and how. In spite of the apparent restrictions imposed by the certificate, there is still a huge number of route configurations that are possible within the confines of the authorization. Second, route network configuration is subject to the route competition pressure exerted by other carriers. Many city pair markets are served by more than one airline. The carriers in these markets compete with each other in a number of ways -- one of them being route competition. When one's competitor(s) is (are) offering a non-stop service, usually there is very little alternative course of action but to schedule a comparable route ("comparable" in this case probably means "non-stop"). Third, route network configuration has to cater to the preferred routings of the passengers. In general, passengers prefer to execute their trips on the most expedient routing -- which can be interpreted as the shortest time path for most business travellers. Route planning has to take this travellers' behavior into account. Fourth, the route network configuration has to take into consideration some facts about the profit potentials of a route. Due to the pricing scheme and cost structure of operating an airline, short and low density route segments are usually unprofitable, while long and dense segments are profitable. One of the tasks in route planning would be to minimize the profit disadvantages of short and low density segments.

Let us summarize the logic behind the route network configuration analysis procedure. For a different set of aircraft routes introduced into

the map, a different competition status and passenger routing pattern results. An example will make this point clear. Instead of the set of aircraft routes shown in Fig. 1-2-1, we reconfigure a different set of routes in the same five-city map (Fig. 1-2-2). Two one-stop routes now serve the system: CHI-WAS-NYC and CHI-CMH-DAY. The competition status of Fig. 1-2-2 is entirely different from that of Fig. 1-2-1. While CHI-NYC was served by a non-stop, now the same city pair is served by a one-stop. The passenger routing pattern is completely changed. The NYC to CHI traffic has to go via WAS, and the CMH to WAS traffic makes the connection at CHI (instead of NYC). We have witnessed a network relationship between an input set of routes and the corresponding output competition and passenger routing pattern. Profit to the carrier, at the same time, is a function of the aircraft routes and passenger routings. To see this, it is convenient to recall that profit is the difference between revenue and cost. A non-stop route draws more passengers than a multi-stop route, and hence more revenue. To fly a longer route segment is more costly in direct operating cost than a short segment. To service a route with a dense passenger flow segment requires a higher route frequency than a route with all sparse density segments (Fig. 1-2-3 illustrates this point). And to operate a higher route frequency means higher cost. In short, a particular route structure input yields a corresponding output of competition and passenger routing pattern, which in turn determines a specific profit figure for the airline.

The route network configuration problem can now be summarized. It is analyzing how to align aircraft routes to the preferred passenger routings

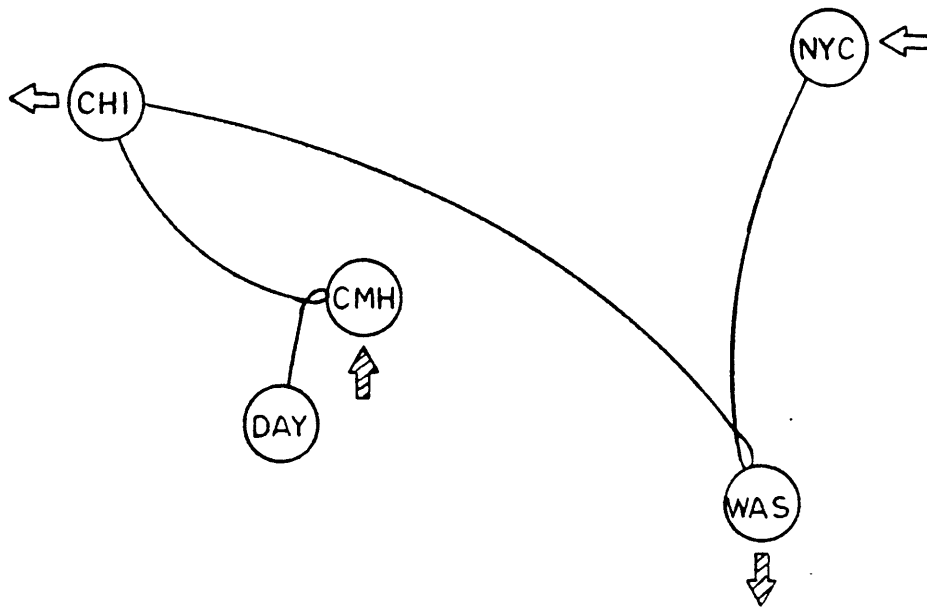
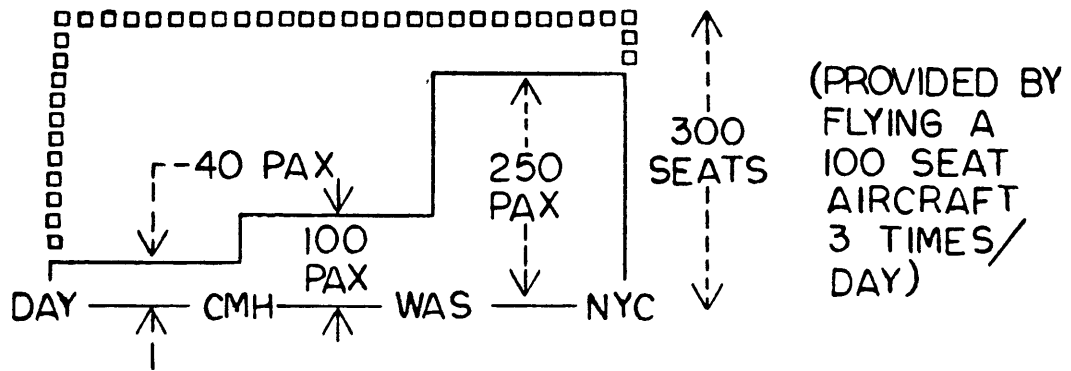
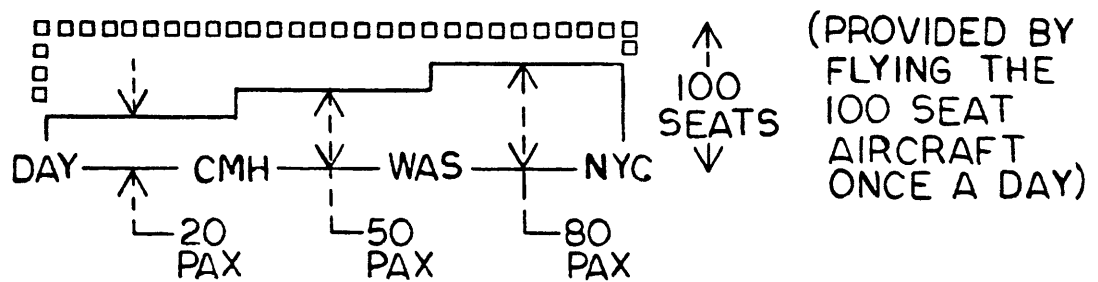


FIG. 1-2-2 ROUTE NETWORK CONFIGURATION 2



HIGH ROUTE FREQUENCY REQUIRED



LOW ROUTE FREQUENCY

FIG. 1-2-3 RELATIONSHIP BETWEEN ROUTE FREQUENCY AND SEGMENT TRAFFIC

in a network. The air carrier is cognizant of the regulatory and competitive environment in which it operates. Consistent with the profit-making motive of a corporate business, the airline plans a route structure that minimizes the profit disadvantages of short and sparse-density segments. This is achieved by judiciously feeding traffic to fill up the underutilized seat capacities in the sparse density and/or short segments. Chapter 2 will elaborate on the route planning problem as a whole from a quantitative economic framework.

At this point, the problem-oriented readers who have little interest in optimization techniques may want to skip the rest of this chapter and proceed to Chapter 2. They should also take note that Chapter 5 contains a case study on the route structure of American Airlines.

1.2.2 An Optimization Formulation of the Problem

The last section has defined our problem in terms of matching aircraft routes to passenger routings. The routes provide the network structure between city pairs. They constitute the structural, or topological, part of the problem. The routings, on the other hand, describe the traffic flow pattern and the connectivity between city pairs on the route network. They impart quantitative characteristics on the route network topology. The structural aspect is usually handled by mathematical techniques classified under graph theory or combinatorics, while the quantitative aspects are usually solved by network flow or algebraic techniques.* In this section,

*A similar distinction is made by Elmaghraby [1970] and Fulkerson [1966].→

a graph theoretic/network flow programming optimization formulation is put forth to model our route/routing problem.

The optimization formulation will be presented in two steps: route generation vs. route selection. The generation phase mainly addresses the structure or topology of the network. The selection phase mainly addresses the quantitative characteristics -- i.e., passenger flow and city pair connectivities in the network. Generally speaking, generation is formulated in terms of graph theoretic techniques, while selection is formulated in terms of network flow programming. The generation step synthesizes all topologically feasible routes to be included in the network flow programming tableau. The selection step evaluates the set of feasible routes by flowing passengers over them. A subset of routes is then chosen which is deemed to be the optimal solution. The combination of generation and selection is referred to as the optimization package.

generation

Let us describe route generation. A route is topologically feasible only if it complies with the C.A.B. route authorities. A "contiguity matrix" is formulated to model this regulatory constraint. The route authority information is encoded graph theoretically in the matrix. By raising the matrix to its first, second, third, ..., powers, all the authorized non-stop, one-stop and two-stop, ..., routes are synthesized. If we denote this

The distinction is made mainly to "fix ideas." Network flow techniques have been used to synthesize network topologies to satisfy a specified criterion.

set of routes by R , we can write $R \subseteq C$, where we state that the set of feasible routes is generated within the combinatorial space represented by C . Routes synthesized this way are then inserted as columns in the mathematical programming tableau of the selection phase. They serve as candidates for the final choice.

selection

A mathematical program is formulated to select among the route candidates generated by the previous step. It is convenient to describe the formulation in terms of constraints and an objective function.

There are two parts to the constraints: passenger flow vs. city pair connectivity. Passenger flow is modelled as a multi-commodity (or multi-copy) flow problem, while city pair connectivity is formulated as a set covering problem. The passenger flow between each origin-destination pair is a separate "commodity" and takes up a copy of a node-arc incidence matrix (hence the term "multi-copy"). The node-arc incidence matrix is an algebraic description of the network. These matrices for all the origin-destination pairs align themselves in a block diagonal fashion in the tableau. By the process of "problem manipulation,"* the various node-arc incidence matrices can be transformed into equivalent arc-chain incidence matrices (Fig. 1-2-3), which is more amenable to a decomposition solution. The linking constraint for these various copies of arc-chain incidence matrices is the set covering matrix. The set covering constraint specifies

*This term is coined by Geoffrion [1970].

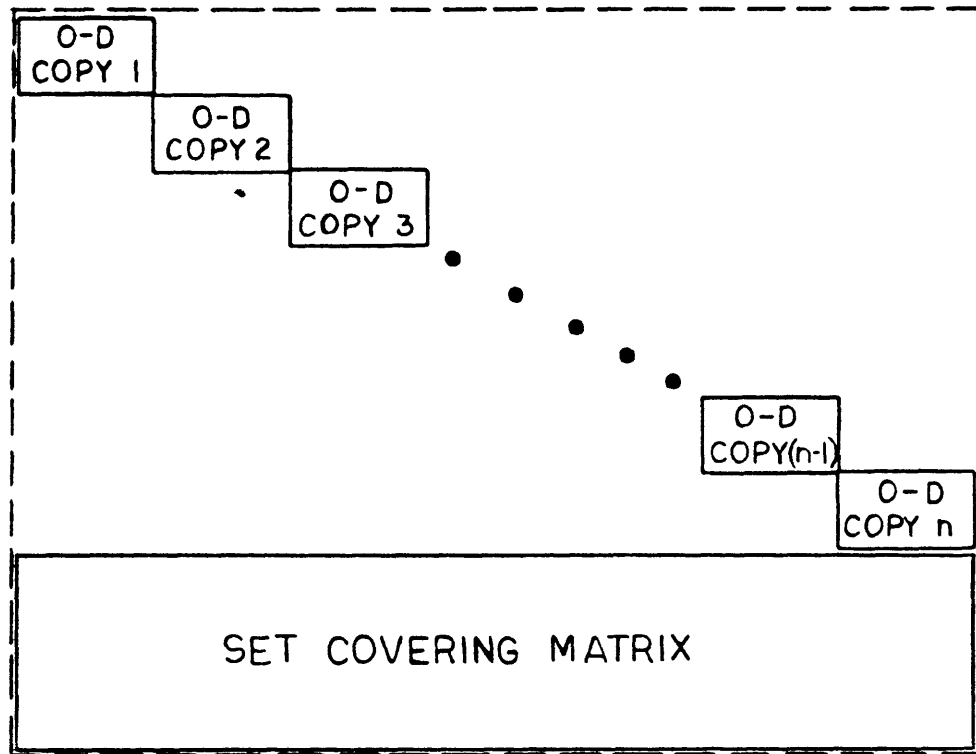


FIG. 1-2-4 INTEGER PROGRAMMING TABLEAU
FOR ROUTE SECTION

that each city pair must be covered by a two-stop routing or better, one-stop routing or better, or a non-stop In this way, intercarrier route competition is quantified.

Let us write the passenger flow constraint in the functional form $F = f(R)$, which states that the passenger flow pattern F is a function of the set of aircraft routes R . And let us write the route competition constraint as $K = c(R)$, which specifies that the routes R must contribute toward an acceptable covering pattern K for all the city pairs. The two constraints can be combined into an evaluation functional form $E = e(R)$, which says that the flow and covering evaluations on the set of routes R should yield a satisfactory system performance E . Alternatively, we can think of it this way: The input set of routes R is required to result in an output network response of E .

To be consistent with the profit-making concern of a corporate business, the mathematical program takes profit maximization as the objective function. Profit is the difference between revenue and cost, both of which are functions of the set of routes, the connectivities between the city pairs and the traffic flow pattern. In other words, profit depends on the input set of routes R and the output network system performance E . We write the objective function as $\max_R I(R,E)$.

the optimization formulation

Having explained the generation and selection phases, let us view the combined process as an optimization formulation. First, we recall that the

routes R are generated within the combinatorial space C via graph-theoretic techniques -- (1) $R \subseteq C$. Second, we recall that this set of input routes R is evaluated and required to yield an output network system performance of E -- (2) $E = e(R)$. The objective is to maximize profit to the carrier, I , which is a function of both the routes R and the system performance E -- $\max_R I(R,E)$. In summary, the total optimization package bears the following functional form:

$$\begin{aligned} \max_R Z &= I(R,E) \\ 1. \quad R &\subseteq C && (1:2:1) \\ 2. \quad E &= e(R) \end{aligned}$$

Viewing the above formulation as a problem-solving tool, the reader may want to ask: (i) What are the givens -- the information which is input to the package? (ii) What are the decision variables manipulated by the package? And (iii) What are the solution outputs from the package? The following three lists will answer these application-oriented questions:

GIVEN

- passenger travel demand as a function of routings
- C.A.B. route authority
- specification of the minimal level of service for a city pair in order to face up to inter-carrier route competition
- fleet specification (i.e., seat, speed and range)
- the airline city map (i.e., the specification of terminals and inter-city distances)
- revenue and cost functions.

DECISION VARIABLES

- route acceptance or rejection
- realized passenger demand (i.e., that part of the potential demand actually served)
- route (or segment) passenger flow identified by origin and destination
- assignment of aircraft type(s) to routes.

SOLUTION

- route network structure
- passenger traffic flow and routing
- realized passenger demand
- route frequency
- fleet type assignment
- fleet size requirement.

For an in-depth elaboration on the optimization formulation (1:2:1), the reader is referred to Chapter 3.

discussion on the optimization formulation

The purpose of this discussion is two-fold. First, the dimension of the optimization problem we are dealing with is given. Second, the special mathematical programming characteristics of the formulation are identified.

The dimensions given here are extremely conservative estimates. The contiguity matrix used in route generation at least measures N by N , where N is the number of cities in the system. The mathematical program used in route selection measures in the order of $N^3 \times N^4$ in its original node-arc

incidence matrix form. In its arc-chain incidence form (i.e., the decomposable form), it measures in the order of $N^2 \times N^2$.

Due to the huge combinatorics dimensionality of route generation, it is practically impossible to generate all the routes in the generation phase and include them all in the mathematical program for selection. The only practical approach would be to identify only those few promising routes to be included in the mathematical program for optimal selection.* We will come back to this point in sections 1.3.3 and 1.3.4.

The optimization formulation given in equation (1:2:1) involves a non-linear, discontinuous objective function. At the same time, we are dealing with an all integer programming problem. The characteristics and dimensionality of the problem exclude the use of existing software packages or "off-the-shelf" algorithms for its solution.

1.3 Methodologies for Network Routing Analysis

It has been pointed out that there are two parts to the route network configuration problem. There is the structural aspect of a route network and then there are the quantitative characteristics such as flow and connectivity. These two qualities are found in many of the network routing problems in transportation.

Network routing problems are observed to share the following common

*A similar approach is used by Hollaway [1970] in solving a special class of mathematical programs. The procedure has been termed by Hollaway as the combination of "identification" and "optimization."

attributes. A transportation network has multi-terminals, or multi-origin-destination pairs. Between a pair of terminals are routes, or routings, along which traffic flows. Under normal conditions, the traffic entering into a terminal equals the traffic coming out of the terminal. For example, the number of aircrafts landing at airport XYZ must be the same as the number of aircrafts taking off, over a planning cycle. Because the traffic between all the origin-destination terminal pairs is accommodated within the same network, a phenomenon called link sharing, or link bundling, would occur. By link sharing or bundling we mean that flows from diverse origin-destination pairs may compete using the facilities of a common link. An airline example would be that in a particular route segment we will find through and connect traffic as well as local traffic, all sharing a limited number of seats available in the segment. The combination of "multi-terminality," "conservation of incoming and outgoing flows," and "link sharing" gives rise to complex "network effects." Network effect refers to the interlocking relationship between the structural and quantitative characteristics of a transportation network. For example, the removal/addition of a link in a network would have profound effects on the traffic flow pattern and terminal pair connectivities.

Although there exists a wealth of solution strategies on the network routing problem, a great deal of frontier still awaits exploration, especially the more analytical approach to the problem. The emphasis on structural aspects of routing usually arises from network synthesis problems, which are concerned with the construction of the network topology ab initio.

Generally speaking, network synthesis problems are solved by graphy theory and combinatorics. The emphasis on quantitative characteristics of routing usually arises from network evaluation problems, which are concerned with the assessment of a particular network topology. Generally speaking, network evaluation problems are tackled by network flow and mathematical programming algorithms. In this section, we will briefly review these methodologies. The frontier will be pointed out. And our modest advancement of the frontier in the solution of the optimization formulation (1:2:1) will be identified.

1.3.1 Network Synthesis Techniques

Network synthesis is defined as the construction ab initio of a network topology to satisfy certain specified criteria. For example, in our route network configuration problem, routes have to be synthesized according to $R \subseteq C$ to provide the desired connectivities between all the city pairs (so that the route competition requirements are satisfied: $K = c(R)$).

There is the manual way and also a more analytical way to synthesize a transportation network. For years, experienced schedule planners in airline offices have been laying out route structures by sheer professional judgment. The shortcoming of this approach is that the human mind is often too feeble to comprehend all the combinatorial alternatives possible with a transportation network of any realistic size. As a result, a number of potentially more superior route configurations are missed in the schedule planning process.

It is only recently that the more analytical approaches to transporta-

tion network synthesis have become possible from a practical standpoint. Mathematicians and computer scientists share equal credit in their contributions. Here are a few examples of the synthesis problem solved by graph theory/combinatorics/network flow, among other techniques. The problem of the minimum spanning tree [Fulkerson - 1966, section II-10] is to come up with a network structure which will connect all the nodes in the network with the least costly set of arcs. For reliability purposes (e.g., in communication networks), we may require that at least k arcs must be suppressed in order to disconnect the network. This gives rise to the minimal k -connected networks [Fulkerson and Shapley - 1961]. Then there is the problem of "shortest path visiting specified nodes," [Dreyfus - 1969, section 5] which has a hybrid flavor of both the shortest path and travelling salesman problem. A final example is the "minimal chain covering of an acyclic network" problem [Levin - 1969, Fulkerson - 1966, section II-6].

The reader may notice that the above examples are chosen because they are in some way related to our route network configuration problem. We are concerned with the least costly way, among other figures of merit, to connect the city pairs in the airline map. In a number of cases, a city pair may warrant to be redundantly served by more than one type of route -- e.g., both a non-stop and a one-stop. The C.A.B. route authority may require a route from X to Y to visit the specified cities A and B or C . Inter-carrier route competition requires an airline to structure a minimal cost (and maximum revenue) set of routes to ensure that certain city pairs are covered by two-stop routings or better, one stop or better, etc.

It is safe to say that the state of the art does not allow for a totally satisfactory solution of all the above four example problems.* For some of the problems, not only are the theoretical bases incomplete, computational approaches likewise have dimensionality limitations. It is felt that, while our problem is complex enough to encompass flavors of all of the four examples, a reasonable solution approach would be to devise special algorithms tailored to our problem. Our algorithm does not claim to solve a general class of synthesis problems. But it is efficient for the route network configuration problem at hand. Here is the logic underlying our synthesis algorithm. A "promising" route network topology is synthesized. The question of "how good is this particular synthesized network" cannot be answered until after the evaluation step is carried out on the given network.

1.3.2 Network Evaluation Techniques

Network evaluation is defined as the assessment, via quantitative models, of the performance (or figures of merit) of a given network topology. For example, in our route configuration problem, passenger traffic is flown over the given network (i.e., $F = f(R)$) and the connectivities between city pairs in the given network are checked (according to $K = c(R)$), just to see "how good" the network is. The traffic flow and the connectivity checks are carried out in the evaluation function $E = e(R)$ formulated as a mathematical program. In general, network evaluation techniques trace out the

*See, for example, Fulkerson [1966] and Dreyfus [1969], Held & Karp [1970].

complex network effects corresponding to different network topologies.

Quite a number of quantitative models have been developed to try to answer the question "how good is a given transportation network." No attempt is made here to review this literature in a comprehensive manner. Rather, only research results related to our problem are summarized. They address the two quantitative characteristics: network connectivity and traffic flow.

network connectivity

Special algorithms have been devised to calculate vertex pair connectivity. They make use of special data structures in computer programming and clever algorithmic shortcuts. One of the most recent algorithms that comes to our attention is that by Steiglitz and Bruno [1971]. These two authors revised Frisch's algorithm, giving it a new and conceptually simple form. Their algorithm is related to the Ford and Fulkerson labelling algorithm.

The connectivity problem of ours is a good deal more involved than the one cited above. We make a careful distinction between the types of city pair connectivities. A city pair connected by a 1-arc chain (i.e., a non-stop) is different from that connected by a 2-arc chain (i.e., a one-stop). Not only are we concerned with the fact that a city pair is connected, but also that it be connected with the appropriate kind of chains.

The special algorithm used in this thesis is particularly adept for use in conjunction with the "list structure" way of storing the set covering matrix data. It reduces the connectivity computation procedure to a scan-

ning instruction on a row of $(m + 1)$ entries, where m is the number of intermediate stops in a route.

traffic flow

The traffic flow part of network evaluation is actually much more involved and challenging than the connectivity part. There have been, up to now, two approaches to the traffic flow problem. They are: (i) the heuristic techniques which are often called "traffic assignment," and (ii) the mathematical programming techniques, generally using multi-commodity flow algorithms. These two approaches are based on entirely different behavioral assumptions on how a unit of traffic, which in our case is a passenger, chooses it/his/her path from origin to destination.* The former (heuristic) approach assumes that each transportation user tries to follow a path from origin to destination which is most convenient for himself. We call this the descriptive traffic flow assumption, or "user optimizing," as some researchers like to call it. It describes the observed behavior of most travellers. The latter (mathematical programming) approach assumes each transportation user can be persuaded to follow a path which is not necessarily the most convenient for himself, but contributes toward promoting the global objective for the system as a whole. We call this the prescriptive (normative) traffic flow assumption, or "system optimizing." The decision-maker "from above" (or the objective function of the mathematical program) prescribes the travel routings of his subordinates (the transportation

*Credit is usually attributed to Wardrop [1952] for making the behavioral distinction.

users). There are advantages and disadvantages of both approaches. Descriptive flows are obviously based on a more behaviorally realistic assumption. Generally speaking, computer packages based on this approach can handle a practical size network. But they are heuristic algorithms, lacking the convenience and theoretical "optimality" of the more analytic approaches. Prescriptive flows, on the other hand, are based on a normative assumption which does not apply in the context of our problem. The reason is that it would mean an air carrier can persuade his customers to travel in such a way, not for their comfort and convenience, but to contribute toward the system profit to the airline. Generally speaking, multi-commodity flows can handle only limited size networks.* The merit of this analytic approach is the convenience in carrying out sensitivity and parametric analysis. Mathematically minded researchers find the multi-commodity flow approach a more structured way to modelling. "Optimal" solutions are obtained.

DESCRIPTIVE FLOW

Descriptive (or user optimizing) flow models have been constructed by a variety of researchers. Gagnon [1961] reported a "Passenger Allocation Model" used by Air Canada for flowing passengers over an airline schedule network. Gunn [1964] of Lockheed-California Company reported a similar, although much more aggregate, model for the problem.** A familiar uncapaci-

*A typical multi-commodity (or multi-copy) traffic flow mathematical program has a row dimension in the order of N^3 , where N is the number of cities in the system.

**Gunn's model actually considers a number of other factors in schedule planning besides traffic flow.

tated traffic flow model to the author is the "Schedule Planning and Evaluation Model" [Kingsley - 1968], used by both the McDonnell Douglas Aircraft Company and American Airlines.* The model evaluates "how good" a given schedule is by flowing traffic over it. The traffic flow output reveals over-loaded flight segments, city pair traffic demand not served, among other statistics. The evaluation therefore prompts the schedule planner to take the necessary remedial measures to upgrade the given schedule, such as adding more frequency to an over-loaded flight, etc. Our survey would not be well balanced if we left out the huge effort urban transportation planners have put into traffic flow research, which they prefer to call "traffic assignment." The programs developed by the United States Bureau of Public Roads [undated] have been widely used in transportation planning offices. The M.I.T. Incremental Traffic Assignment method and its subsequent evolutions have refined the capacitated traffic assignment techniques to a sophisticated level [Ruiter - 1968]. The M.I.T. researchers pay particular attention to the fact that travel demand is a function of the level of service. An "incremental assignment" method is used to handle the level of service feedback loop between supply and demand.

It should be noted here that although all these descriptive methods are heuristic in nature, they are generally based on some sort of minimum path computations.

*The author worked on an extension of the model in both companies.

PRESCRIPTIVE FLOW

Prescriptive (or system optimizing) flow models are usually formulated as a multi-commodity (or multi-copy) flow mathematical program. The application of the multi-commodity flow formulation to transportation traffic flow problems was initiated by Charnes and Cooper [1961]. These two authors modelled the traffic from each origin node as a "commodity" flowing over a copy of the network. The topology of the network copy is represented by a node-arc incidence matrix. There are as many copies of these matrices along the diagonal of the mathematical programming tableau as the number of origin nodes (hence the term "multi-copy"). The linking constraint for the various copies imposes capacity constraints on each link, which carries a "bundle" of traffic from diverse origins. The advantage of a multi-commodity/multi-copy formulation is that we can distinguish the components of the bundle of traffic flow in a link. For example, in a flight segment of an airline schedule, distinction is made between local vs. connecting traffic, since passengers of each origin-destination pair could be identified as a separate "commodity."

Tomlin [1966] was the first to show the mathematical relationship between the node-arc incidence matrix formulation and the arc-chain incidence formulation. The transformation was related to the Dantzig-Wolfe decomposition principle [1961]. Jarvis [1969] further extended Tomlin's findings. He unveiled the relationship between the Ford and Fulkerson column generation procedure [1958] and the procedure of annexing a chain to the arc-chain tableau. In a much more applied context, Jessiman, et al. [1970]

overcame the "curse of dimensionality" problem in the multi-copy formulation by using a special solution procedure for each subproblem copy, at the cost of losing optimality. Their computerized package models traffic flow over an airline network.

It should be commented here that all of the above mathematical programming formulations have an essentially fixed (or perfectly inelastic) travel demand.

DESCRIPTIVE VS. PRESCRIPTIVE FLOW

A couple of researchers have examined the relationship between the descriptive vs. prescriptive flows. Jorgensen [1963] reported that if link capacity is not reached, traffic assignment and single commodity (i.e., single copy) mathematical programming give the same traffic flow solution. Here is our extension and interpretation of this finding to the multi-copy/multi-commodity formulation. It is found that multi-copy prescriptive flow computation without the linking capacity constraint would yield a descriptive flow pattern. This is actually quite intuitively appealing. Without the linking constraint, the various network copies along the diagonal of the tableau would be "untied" into separate origin-destination minimum path problems. The computation of separate minimum paths for each origin-destination pair precisely yields the uncapacitated descriptive flow solution. At the final phase of this dissertation, Dafermos [1971] published research results that established some theoretical connections between descriptive vs. prescriptive flows (or, as she put it, "user optimizing" vs. "system optimizing").

In this piece of research, we have succeeded in using the more behaviorally accepted descriptive flow assumption but solving the flow problem in an optimization scheme (instead of the usual heuristic approach). Travel demand is formulated as a function of the level of service. Because a descriptive flow assumption is formulated in a multi-copy integer program, we end up with a discontinuous objective function (its exact form appears in Chapter 3). It is added here that the row dimension of the original multi-copy node-arc formulation is huge -- in the order of N^3 , where N is the number of cities in the system. We consider it part of our contribution to have partially bridged the gap between the heuristic descriptive flow and the analytic prescriptive flow approaches. It is rather gratifying that not only the solution method can handle the huge dimension of a multi-copy formulation, but also that it yields a supply/demand equilibrium solution (instead of the usual non-equilibrium solution).

1.3.3 Network Improvement Techniques

Network improvement is defined as upgrading the performance (or figures of merit) of a network by a combination of synthesis and evaluation techniques. For example, in our route network configuration problem, we may be given the existing route network of the airline. The question is then asked "Can we do better than the existing route structure, and if so, how?" The answer we offer is that improvement could possibly be made via a practical optimization procedure made up of alternating synthesis and evaluation steps. The synthesis step suggests alternative network configurations. The evaluation step helps to select the configuration with the best figure

of merit. Through the iterative steps of synthesis and evaluation, the route network could be improved, provided the initial route network is not the "optimal" configuration to begin with. This iterative improvement technique is a way to successively approximate the supply/demand equilibrium solution. Aircraft routes are successively aligned to the preferred passenger flow routings (or routings to routes). This route/routing alignment procedure is exactly the level of service feedback loop between the origin-destination travel demand and the supply of routes serving the city pair. Network improvement can therefore be viewed as a process converging toward traffic flow equilibrium in a transportation network.

There are two approaches to the network improvement problem. Again, we categorize in terms of the more qualitative vs. the more quantitative techniques.

qualitative techniques

Transportation network analyses are usually of such a complexity that professional experience and judgment often play an important part in the planning process. It is the present state-of-the-art to couple a large amount of qualitative decisions with quantitative results to arrive at the final output. The obvious example that comes to the author's mind is the Schedule Planning and Evaluation Model used at McDonnell Douglas Aircraft Company and American Airlines. The model by itself is purely a traffic flow evaluation tool. It helps to show the inadequacies of an existing schedule by evaluating it through passenger flow. The ultimate objective of the schedule planning process, of course, is to improve the schedule,

which involves both the evaluation step and the synthesis step. The synthesis step in this case is handled by the schedule planners. Based on the nature and context of the shortcoming of a certain route in the existing schedule, the schedule planner would synthesize in his mind a new alternative to replace the given mode of operation. A former multi-stop route may be upgraded to a non-stop route or vice-versa. This new route would be incorporated into the schedule, and the new system schedule is evaluated again via the traffic flow model. In this way, a large amount of qualitative judgment enters into the network improvement.

quantitative techniques

We will review only those quantitative techniques on network improvement which are related to our routing problem. We recall the route network configuration problem requires three analysis techniques. They are: (i) the route synthesis techniques used in the generation phase, (ii) the set covering and (iii) multi-copy flow evaluation methods used in the selection phase. It was pointed out that the route synthesis techniques, being topological in nature, are comprised of enumerative type combinatorial schemes. Multi-copy/multi-commodity flow, on the other hand, belongs to a class of network flow problems which has more algebraic flavor. The set covering problem, however, has been solved by both enumerative schemes and algebraic programming methods. With this in mind, we will review the literature on network improvement in terms of algebraic vs. enumerative methods.

ALGEBRAIC METHODS

The matrix algebra underlying classical methods in mathematical pro-

gramming, such as the simplex method, is actually an improvement scheme. A basic feasible solution is first obtained. Then the basic feasible solution is improved upon via column (variable) entries and exits. More refined solution strategies used in large scale programming achieves savings in storage requirements by synthesizing the columns only as needed. An evaluation rule is set up to select the best column to enter into the basis, resulting in an improvement in the objective function. The multi-copy flow programming algorithm follows the idea of this column generation scheme.

To solve the multi-copy flow problem, the original node-arc incidence matrix formulation is first transformed into the decomposable arc-chain master program [Tomlin - 1966, Jarvis - 1969, Wollmer - 1970], with the arcs on the rows and chains as columns. Chain columns are synthesized as needed by solving each subprogram (a "copy") as a minimum route/chain problem, with arc costs modified by the shadow price from the master program. An evaluation criterion is set up to select the most promising chain columns to enter into the basis of the master program, resulting in an overall improvement in the objective function. The resulting shadow prices from the master program are again fed back to the subprograms, modifying the arc costs. New minimum route/chain computations are performed for each of the subprograms and new chain columns are synthesized to be appended to the master program. In this iterative manner, the traffic flow solution is incrementally improved.

Our solution strategy on the route network configuration problem proceeds in a similar manner as above. The aforementioned method used in pass-

ing shadow prices (dual variable) back and forth is not applicable because of the particular discontinuous objective function and the fact that we are dealing with an integer program. Instead, we use a primal decomposition approach where primal feasibility is always maintained. While the chain column synthesis (or "column generation") part is the same, the evaluation of chain columns for the best selection is based on a "marginal profit" concept. The chain column with the largest marginal profit potential will be accepted, resulting in an improvement of the route network configuration. Further discussion on the marginal profit concept in our primal decomposition method appears in both Chapters 2 and 4. Chapter 2 motivates the marginal profit idea via its economic context. Chapter 4 exposes the algorithm of determining marginal profit with a fair amount of technical detail.

ENUMERATIVE METHODS

Enumerative methods usually have less rigorous mathematical structure than the algebraic methods. As a solution improvement tool, it has its merits. Clever algorithms of the enumerative nature have solved a number of practical problems which appear to be insoluble from a more rigorous algebraic framework [Balinski - 1965]. In the generic name of enumerative methods, we have included dynamic programming, implicit enumeration, branch-and-bound, etc. Names like "tree search" or "combinatorial programming" have been used to refer to the same method.

Enumerative algorithms can be viewed as an improvement scheme. A tree structure is synthesized by a branching strategy. Evaluation of the figure of merit is performed at certain nodes of this tree. The evaluation

step may take the form of simple max/min operators as in numerous common uses of dynamic programming, or it may involve solving a linear program as in the Land and Doig method [1960]. But invariably, the evaluation helps the subsequent synthesis procedure by excluding obviously unpromising branching strategies. The figure of merit is incrementally improved as the result of the combination of the synthesis and evaluation steps.

Implicit enumeration and branch-and-bound can be viewed as special cases of dynamic programming where the huge enumeration tree is "collapsed" by clever feasibility/exclusion/dominance type pruning rules. The more general context of enumeration techniques is dynamic programming in an "unbounded horizon," where the convergence scheme is achieved via the "method of successive approximation" [Bellman, et al. - 1970, Wagner - 1969]. This method involves looping through labelling the graph numerous times until two consecutive solutions remain unchanged. The method of successive approximation can be viewed as a generalized network improvement algorithm.

Our route network improvement algorithm involves a tree synthesis and connectivity/traffic flow evaluation sequence similar to the above-mentioned framework. The tree is synthesized by raising the power of a "contiguity matrix." The trees synthesized represent the minimum distance non-stop, one-stop, two-stop, etc., routes between city pairs. At certain nodes of these trees, connectivity between city pairs is checked and traffic flow is updated according to the current route configuration. The evaluation step computes marginal profit of a route and helps to prune the trees. Branch-and-bound rules further accelerate the enumeration algorithm.

Our route network configuration problem belongs to a class of "unbounded horizon" problems in dynamic programming. The method of successive approximation has to be used to loop through the state-stage space a number of times. Each looping improves the solution. If two successive loops give no improvement, the algorithm is stopped since the optimal solution has been found. Notice that since the method improves on the solution while keeping primal feasibility, the algorithm can be stopped anytime and an acceptable solution better than the one we started with is guaranteed. This is one of the biggest advantages of the algorithm from a practical application viewpoint.

1.3.4 Route Improvement, Synthesis and Evaluation -- R.I.S.E.

The route network configuration problem was defined and formulated into an optimization format in section 1.2. Section 1.3 up to this point has been devoted to a review of the techniques related to the analysis of our routing problem. Here the total optimization procedure for our problem is summarized. The readers will see how the original formulation is transformed into a framework amenable to decomposition through "problem manipulation,"* and how an acceptable, improved route configuration is obtained within reasonable computational requirements through a decomposition "solution strategy."*

Recall that the route network configuration problem has two components to it. They are the topological part concerned with network geometry and

*Terms borrowed from Geoffrian [1970].

the quantitative part dealing with connectivity/passenger flow attributes in the network. Specifically, the route structure provides the city pair connectivities whereby passenger flows are facilitated. Route candidates are synthesized in the route generation phase. These candidates are then evaluated in the route selection phase. Generation and selection cannot be carried out as totally separate, independent processes, for this would mean, for any reasonably sized network, that hundreds and hundreds of route candidates would be synthesized in the first step and they would be subject to evaluation in a huge dimension in the second step. The dimensionality problem would be overwhelming. The only computationally feasible approach is to break the original problem up, or to decompose it, into a number of partial generation/selection steps via "problem manipulation." The coupling of generation/selection in a repetitive sequence means that a route would be synthesized and then readily evaluated, so that obviously undesirable candidates could be excluded from the dimension space "early in the game." Through such a "solution strategy," the route network configuration is successively improved, with prudent computational requirements.

problem manipulation

How do we manipulate the original constrained optimization problem formulation into a decomposable form? The answer, in a nutshell, is by re-writing the node-arc formulation into an arc-chain formulation. Recall there are two parts to the optimization statement in Equation (1:2:1). The first part, $R \subseteq C$, says that the set of route candidates R is generated by enumerative scheme in the combinational space C . The second part, $E = e(R)$,

says that the route candidates R are selected according to the evaluation function $e(\cdot)$, where the function is mainly a mathematical program expressed algebraically in terms of node-arc incidence matrices. The two parts represent the generation vs. selection phases, separately. The aim of problem manipulation is to transform (1:2:1) into a format whereby the enumerative generation step could be combined with the algebraic selection step, so that only a handful of feasible routes would be considered at a time. It turns out that by re-writing the node-arc incidence matrices as arc-chain incidence matrices, the "chains" correspond physically to the routes and routings. The routes/routings synthesized in an enumerative fashion in $R \subseteq C$ can therefore be readily inserted as columns in the mathematical programming tableau, to be evaluated in an algebraic manner $E = e(R)$. Through problem manipulation, the gap between generation and selection is narrowed to an extent that the following improvement solution strategy can be applied.

solution strategy

A primal decomposition solution strategy is used to improve the route network configuration. The improvement procedure is comprised of elements of both enumerative schemes like dynamic programming, branch-and-bound, etc., and algebraic methods such as column/row generation in a mathematical programming tableau.

The route synthesis step of improvement can be thought of as the tree branching step in the solution algorithm. Route evaluation, being the other step in improvement, is performed at certain nodes of the tree. The

decomposition algorithm then proceeds as follows. Each origin-destination city pair constitutes a subprogram. A city pair corresponds exactly to an "arc," or route segment, of the arc-chain formulation. By considering one city pair subprogram at a time, "arcs" are incrementally appended row-wise to the mathematical programming tableau. Solving the shortest route problem for each of these city pair subprograms yields the chains to be inserted column-wise to the tableau. The shortest route algorithm can be visualized in the state-stage space of dynamic programming as tree construction. To summarize the decomposition scheme conceptually in one sentence: The tree construction process via contiguity matrix manipulation provides the shortest routings for the descriptive passenger flow by the multi-commodity mathematical program at the nodal points of the tree. In this manner, the enumerative techniques work hand-in-hand with the algebraic methodologies.

The algorithm is a primal one in the sense that no shadow prices are used in the solution strategy. Instead, the concept of climbing in the feasible direction with the best marginal profit is used. Because primal feasibility is always maintained, the algorithm can be stopped prematurely and an improved solution is still obtained.

The method of successive approximation used in our dynamic programming solution strategy is a numerical way to align routes to preferred routings, converging toward a network supply/demand equilibrium. The method of successive approximation would converge more expediently if a good existing route configuration is available to be used as the "initial policy."

comments

The name R.I.S.E. has been given to our route configuration model, the solution algorithm of which has been encoded in a FORTRAN IV-G software package consisting of forty routines. R.I.S.E. is an abbreviation for Route Improvement, Synthesis and Evaluation, which adequately describes the methodologies utilized in the solution procedure. Quite incidentally, the connotation of the word "rise" conveys the idea of the successive approximation way of "climbing" toward a more and more profitable route configuration.

Standing at his biased position, the author wishes those schedule planners using our solution method to be executives "on the RISE." For the particularly aspiring types, the further details of the algorithm (to success!) are contained in Chapter 4.

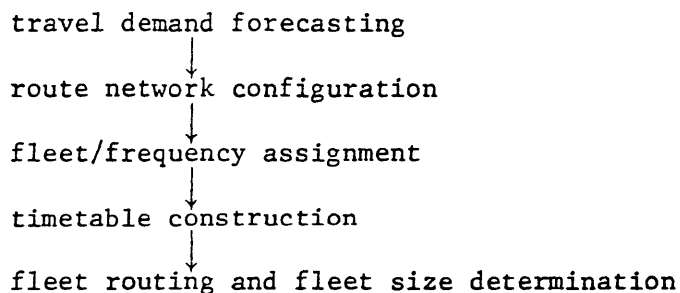
1.4 The R.I.S.E. Model in the Schedule Planning Process

It has been pointed out in section 1 that the route network configuration problem modelled as R.I.S.E. is only one out of many routing and scheduling issues. The question is raised, "How does the contribution of this dissertation fit into the schedule planning process?" In this section, we will discuss how R.I.S.E. coordinates with the other scheduling models of the hierarchy in a supply/demand equilibrium context. Although the discussion will only be of a conceptual nature, the careful reader will find the proposed ideas quite promising to be used in an extension of the present piece of research.

There are various ways to coordinate the different routing and scheduling tasks. Here, we have categorized them into three groups: sequential coordination, simultaneous coordination and finally the simul-sequential approach.

1.4.1 Sequential Coordination

The sequential way of schedule planning is the most widely practiced scheme. It involves splitting up the logical tasks of routing and scheduling in some sort of hierarchical sequence. A quantitative model or qualitative analysis is used to handle each task. The sequential planning procedure then involves screening through such a series of models or analyses; the output of one model/analysis serves as input to the next. For example, the following is a common schedule planning sequence:



In the above hierarchical series, the airline's share of passenger demand for the season is forecasted in the first step. The existing route network configuration is modified, usually manually, according to the forecasted potential traffic. Then fleet type and frequency of service are assigned to each route of the route network. The arrival and departure times at the stations along a route come next, resulting in a timetable. Fleets are now circulated in this timetable regarding overnight stays and routine

maintenance. A by-product of fleet routing is the fleet size required to satisfy a given timetable.

The sequential way of schedule planning has its advantages. It is a logical way to break up the huge problem of schedule planning into its priority ranked components. The "curse of dimensionality" of the schedule planning process is avoided. The serial ranking of long run over short run tasks in this way roughly corresponds to the "chain of command" found in an organization.

Actually, the sequential way of schedule planning is more than a "one-way traffic" type chain of command. Analyses performed at the bottom of the hierarchy may uncover some facts which the upper echelons might have overlooked. This would require a revision of the initial plan from the top, and then screening through the lower levels again. In section 1.1.2, this feedback loop has been referred to as the "schedule refinement loop." For example, in the fleet/frequency assignment model, an average fleet utilization figure of x hours per day usually has to be assumed. This assumed figure cannot be verified until after the fleet routing model lower down in the hierarchy has been analyzed. If the precise fleet utilization rate determined in the fleet routing model is drastically different from the assumed figure, the original fleet/frequency assignment has to be performed once again, or revisions have to be made. The output from the revised assignment would be screened through subsequent analyses once again. And the question of whether the assumed utilization is realizable arises one more time. This looping may have to be cycled several times before an acceptable

schedule is finalized.

The sequential coordination approach to scheduling has the following drawbacks. If the schedule evaluation loop is not cycled enough times, as is usually the case due to time constraints, concessions have to be made. Take the same fleet utilization example: Suppose the assumed fleet utilization rate is too high and actually not enough fleet exists to satisfy the frequency assignment. Instead of lowering the assumed utilization figure and re-assigning fleet and frequency, the departure times in the timetable may be shifted earlier or later. This allows the fleet to make more flights by cutting down on the ground time. Such manipulation may just yield an adequate number of flight hours to satisfy the assigned frequencies or, in optimization jargon, the schedule is now fleet feasible. But the "rules of the game" are violated in this case. By shifting the departure time away from the passengers' most preferred time would result in losing patronage to a competitor. Also, the ground crews may complain about the tight schedule they have to follow in order to get an aircraft rushed to the next flight shortly after it arrives. This example illustrates two points. First, the sequential planning scheme may in practice give rather unsatisfactory schedules (not to say "optimal" schedules). Second, the "level of service feedback loop" is usually ignored. Seldom is analysis performed to relate the passengers lost due to a less preferred departure time. In this sense, a non-equilibrium solution, instead of a supply/demand equilibrium solution, results.

1.4.2 Simultaneous Coordination

The truly simultaneous way of schedule planning could be described as the utopia towards which many would aspire. It refers to a large-scale model in which all the different issues in scheduling are resolved together. A quantitative approach is used. The advances in mathematical programming and computer technology have helped to partially fulfill the dreams of many a simultaneous planning advocate.

The following is a graphic display of simultaneous schedule planning:

subproblem 1 \longleftrightarrow subproblem 2 \longleftrightarrow ... \longleftrightarrow subproblem n

One simultaneous approach would be to formulate the above n issues in a large mathematical programming tableau. Solution techniques are then applied toward the tableau for the "optimal" schedule. It has to be commented that quite usable codes such as MPSX/370 on IBM machines and OPHELIE/LP on CDC machines do exist to facilitate the simultaneous method. Several researchers have tried to attack the schedule planning process this way. Included in the list is Vandersypen [1971], whose partial simultaneous model includes a demand function, fleet/frequency assignment and the temporal aspects such as departure time.

The simultaneous approach to scheduling has obvious advantages. The schedule evaluation loop and level of service feedback loop are built into the large mathematical programming tableau. The schedule obtained would be feasible, optimal and also in supply/demand equilibrium (if a demand function is used). On the other hand, there are numerous disadvantages. First of all, there is the dimensionality problem. Row numbers in a mathematical

program for networks grow in the order of N^k , where N is the number of nodes in the network and k is usually greater than two. With the existing computer technology, it is clearly impractical, if not impossible, to resolve all scheduling issues in one gigantic mathematical program. Another difficulty is in finding a solution algorithm once the program is formulated. Including all scheduling issues in a tableau would introduce numerous non-linearity, non-convexity, mixed integrality problems in the formulation. The state of the art in mathematical programming theory is a far cry from solving a non-linear, convex and mixed integer program of the large size we are talking about, not to mention production oriented computer codes.

Clearly, the simultaneous approach is at best utopian. Breaking up the schedule planning process into parts is necessary, not only for dimensionality reasons, but also in accordance with the division of labor philosophy in an organization. And there do exist logical break points in routing and scheduling, such as the crew scheduling problem's being logically different from the problem of gate utilization at a station, etc. The crux of the problem is where to segment the schedule planning process and how to coordinate the interactions between the subproblems. The following section is our proposed answer to this question.

1.4.3 A Simul-Sequential Approach

It has been pointed out that a number of scheduling decisions can be quite naturally ranked in a hierarchy. The long-run or far-reaching issues should logically be resolved before the short-run, or the more immediate, details. But it should be recognized that there are interactions between

the different levels of the "chain of command" -- the result of short-term analysis may reveal it is advantageous to revise the initial long-term goals. This interaction has been named the schedule refinement loop. Looking at the total scheduling system, we have identified another kind of interaction. The demand for travel is a function of the level of service offered by the schedule, and likewise a better schedule induces more patronage on the system. We call this interface between supply and demand the level of service feedback loop. In short, there is a hierarchical, serial, or sequential structure to the schedule planning problem and then there are system interactive, or simultaneous coordinations between the individual subproblems in this serial structure. The simul-sequential approach is a proposed, quantitative analysis scheme designed to exploit the sequential characteristics of scheduling to our computational advantage, while at the same time an iterative algorithm takes care of the simultaneous interaction of the subproblems in the sequence. The iteration converges on to an equilibrium solution which is "optimal" with respect to the adopted figure of merit. The original problem is not only decomposed longitudinally in a multi-level fashion, but also laterally in a stagewise manner, as shown in Fig. 1-4-1.

The following simul-sequential coordination scheme is specifically tailored for the routing and scheduling problem. The nature of other large-scale problems may motivate other ways to break up and structure the process. The suggestions put forth here are therefore not to be taken as a

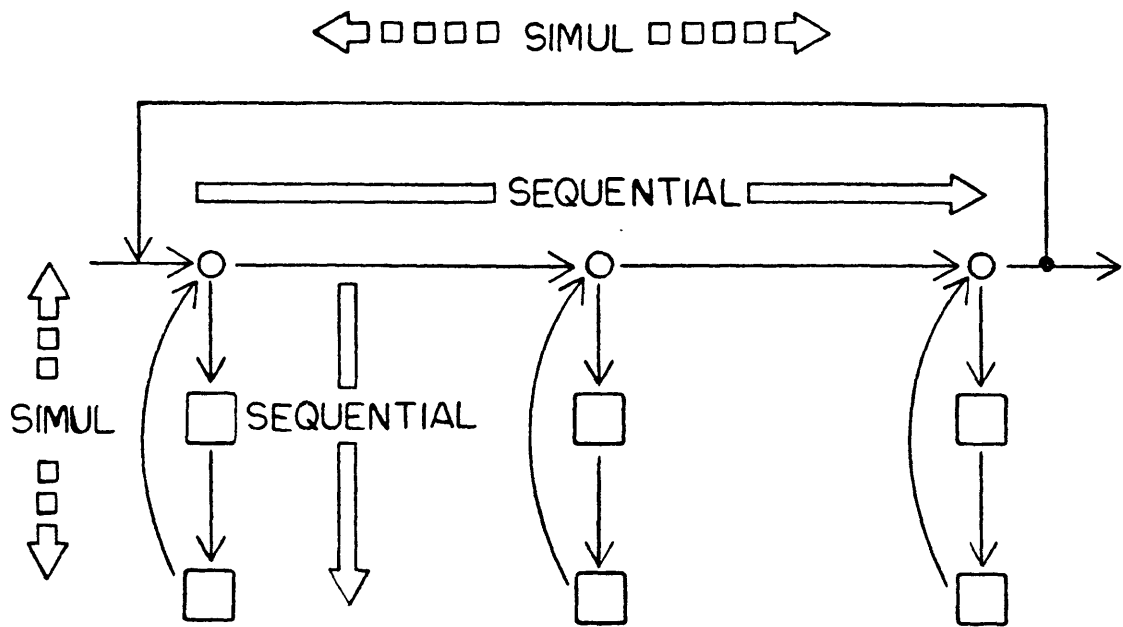


FIG. 1-4-1 SIMUL-SEQUENTIAL COORDINATION

generalized decomposition method.*

In the example given below, a coordination structure is put forth to concatenate R.I.S.E. with the fleet/frequency assignment and fleet routing models. The above three models are structured hierarchically in a longitudinal decomposition. There is another decomposition in the lateral dimension (see Fig. 1-4-2). The lateral decomposition is exactly the stage-wise decoupling of routes into non-stops vs. one-stops vs. two-stops, etc. Due to space limitations, the discussion here has to be cursory. A number of statements will be made with an intuitive explanation at best. For a more in-depth treatment of a similar approach applied in ground transportation networks, see the author's master's thesis [Chan - 1969].

longitudinal decomposition

The "chain of command," or hierarchy, of scheduling decisions is hung onto the simul-sequential structure as a "leg" (see Fig. 1-4-2). A schedule refinement loop is built into the "leg" to handle the interactions between the models in the different levels of the hierarchy. The longitudinal decomposition serves two functions in this example of simul-sequential structure. First, it puts into a route network configuration obtained in R.I.S.E. the more detailed attributes such as fleet type, frequency, departure time, etc. This is done by screening the output from R.I.S.E. through a series of lower level models. The refinement allows for a more precise evaluation, or assessment, of exactly how good the given route structure is

*For a more "complete theory" of multi-level systems, see the work of Mesarovic, et al. [1970].

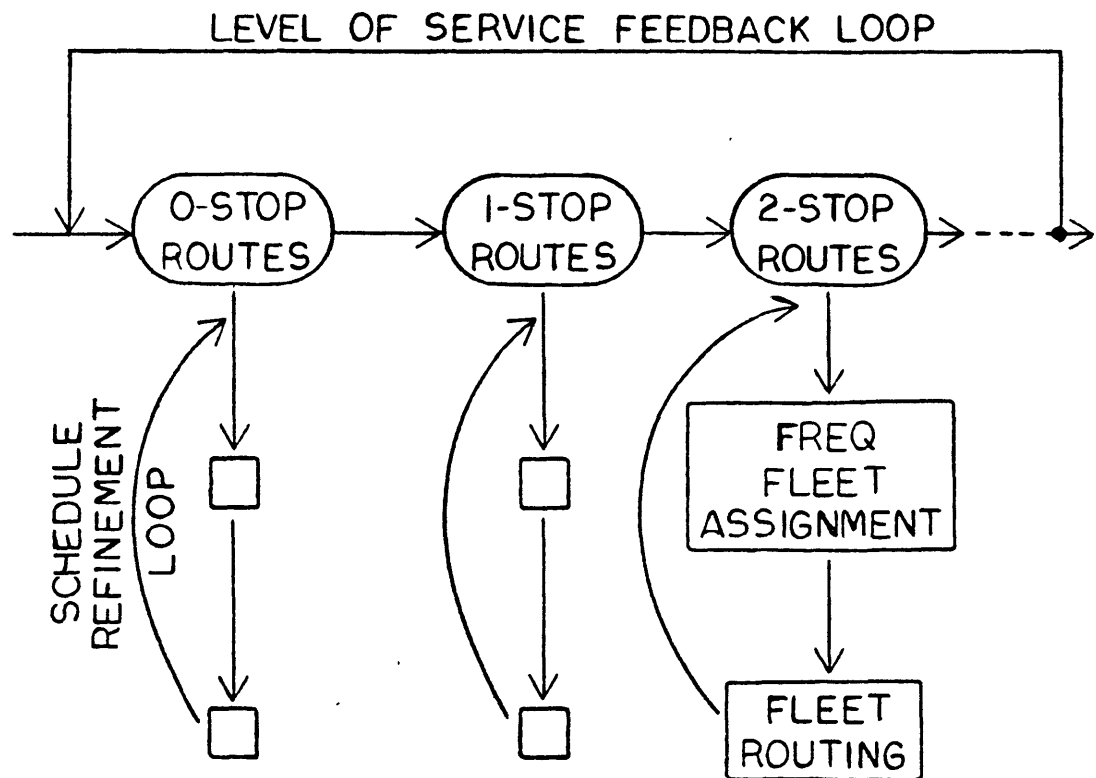


FIG. 1-4-2 SIMUL-SEQUENTIAL APPROACH
TO SCHEDULE PLANNING

after we have considered frequency and departure time, etc. Second, the longitudinal decomposition iterates on the sequence of three models in the leg until feasibility and optimality (in the mathematical programming sense) are obtained.

To facilitate the longitudinal decomposition, "conservative upper bound" figures have to be used in the higher level models so that feasibility and optimality would be guaranteed in lower level models. We will elaborate this point immediately below.

The longitudinal decomposition starts with R.I.S.E. In R.I.S.E., travel demand is modelled only a function of whether the routing is non-stop, multi-stop, or connection. Other attributes that affect demand, such as frequency and departure time, are fixed in this level, but they will be considered in lower level models. The demand function used in R.I.S.E. would have to, in the spirit of retaining optimality and feasibility in a multi-level context, assume a practical upper bound of daily frequency and the "best" departure times. This is done so that a route would not be prematurely rejected because of scanty traffic. If traffic is indeed scanty, the route will be "knocked down" later on in the fleet/frequency assignment model. For the same optimality and feasibility reasons, imaginary aircraft types which can fly the shortest as well as the longest flight range allowed in the fleet will have to be used. Similarly, the imaginary aircraft has to be the least expensive to operate in the fleet for the particular route under consideration. Notice that no fleet size constraints exist in R.I.S.E.; the fleet requirement is actually an output. The primary

solution from R.I.S.E. is a route network configuration.

The next level in this longitudinal series is the fleet/frequency assignment model. In this level, a further dimension has been added to the demand function. Travel demand is now not only responsive to whether the routing is non-stop, one-stop or connect; but it is also a function of the daily frequency.* The demand figures should be practical upper bounds based on the fact that all flights are dispatched at prime times, thus capturing the best traffic possible. This is again done for feasibility and optimality reasons in this multi-level framework. If, in fact, some departure times have to be scheduled differently from the prime times, the corresponding patronage will be "knocked down" in the fleet routing model a level down. As mentioned earlier, an average fleet utilization figure is assumed. A fleet size availability constraint is placed on the system. The output from this second level model is an assignment of fleet type and frequency to the routes obtained from the first level model R.I.S.E.

In the third (and bottom) level is located the fleet routing model. Yet another dimension is added to the demand function -- the time of day. Demand in this case is assumed to vary over the day, assuming "peaks and valleys." The fleet routing model actually selects the exact departure time in the process of routing the fleet around. Passenger patronage is further "knocked down" from that output from the second level model because

*This is actually referring to the market share curves used in some existing models in the Flight Transportation Laboratory at M.I.T., where the author is performing this piece of research.

it may turn out that it would be a better trade-off to dispatch at slightly off-peak times at some stations for a more expedient fleet routing over the whole network. The result of the fleet routing model will be a finalized timetable, with refined cost and revenue figures for each flight.* As we mentioned in section 1.4.1, if the average fleet utilization figure given by this fleet routing model disagrees with the assumed utilization in the fleet/frequency assignment model, iteration has to be carried out between the second and third level. That is why the schedule refinement loop is present.

Notice that the different attributes of a schedule assume definite figures as we screen through the longitudinal sequence of models. First, a tentative route structure with aggregate estimates on cost, revenue and fleet requirement is obtained from R.I.S.E. In the second level, some of the routes suggested by R.I.S.E. may be rejected as more detailed assignment of traffic, frequency and fleet types is made. The cost and revenue figures are refined accordingly. Finally, in the third level, the time of day is determined. And some of the frequencies assigned in the second level may be rejected -- finalizing to a definite timetable. The exact cost/revenue implications are obtained, correspondingly. The longitudinal decomposition process, viewed in this light, serves as a refined evaluation on how good the initial route structure really is. This refined evaluation is carried out through a series of models, which successively introduce more de-

*For an illustrative example of such a model, see Chan, et al. [1970].

tailed parameters (first frequency, then time of day).

lateral decomposition

Lateral decomposition refers to partitioning the schedule planning process into stagewise decisions as shown graphically in Fig. 1-4-2. The interaction between the different stages in the lateral structure is the supply/demand interface. A level of service feedback loop is built into the lateral decomposition procedure to improve upon, until finally yielding the equilibrium solution. It is to be noted that this improvement looping is specifically designed to take place in the most aggregate level of the hierarchy, where the route network is being configured, so that dimensionality problems would be minimized.

We are motivated by the inherent nature of the route configuration problem to examine it in a number of stages. Schedule planners think of non-stop routes first, then one-stops, two-stops, etc. R.I.S.E. partitions the problem correspondingly into the non-stop stage, one-stop stage, two-stop stage, etc. The existence of network effects, or system interactions, requires a feedback loop to coordinate the various lateral stages. Let us give an example. In the non-stop stage, a non-stop route may be configured to serve a city pair. In the one-stop stage later, the same city pair may warrant an additional one-stop service. There are two effects from the one-stop stage decision. First, the total market share of traffic carried by the airline for the city pair is likely to increase because of an upgraded service consisting of both non-stop and one-stop flights. Second, there is likely to be a diversion of some traffic from the former non-stop route to

the one-stop route, which may necessitate cutting down on the frequency for the non-stop. The traffic flow and cost/revenue picture is quite different after the implementation of the one-stop route. These system effects are due to (i) the interlocking relationship between the components of a transportation network and (ii) a demand function which is responsive to the routing level of service. The method of successive approximation is used to loop through the various lateral stages in order to redistribute traffic and re-adjust cost/revenue computations. Each iteration loop improves on the route network configuration by matching supply and demand more closely to the equilibrium solution. In a nutshell, the tree searching algorithm generates and selects aircraft routes which are most aligned with the preferred passenger routings.

Let us describe how lateral decomposition articulates with longitudinal decomposition. Graphically, the vertical schedule refinement loops are suspended as "legs" from the various "stages" of the horizontal level of service feedback loop (Fig. 1-4-2). The reader should notice that there are no lateral interactions between the legs except at the top. All the supply/demand matching is done in the aggregate route network level, where only the spacial dimension, instead of both spacial and temporal dimensions, of routing and scheduling is present. The author feels that this aggregate level, where R.I.S.E. is located, is the most appropriate level to handle the level of service feedback loop, for obvious dimensionality reasons. It is also the most suitable dimensional space to synthesize imaginative route network alternatives which could represent innovative points of departure

from the existing practice. For example, the fundamental questions of whether to enter into a city pair market, or to discontinue serving a city pair, are raised, instead of the more day-to-day questions of whether to advance the flight by five minutes. Since the dimensionality space is still comparatively small, more combinatorial search can be carried out in R.I.S.E. than when the schedule is as detailed as in the timetable construction level.

The lateral decomposition algorithm is an improvement procedure consisting of the synthesis and evaluation steps. Only after careful evaluation is performed on a synthesized route can the route be verified as being able to improve the schedule. R.I.S.E. generates different route candidates for consideration at each node of the combinatorial tree. The selection of the final set of routes can only be determined after subjecting the proposed route network to the whole hierarchy of models in each leg of the simul-sequential structure -- starting from connectivity and traffic flow evaluation in R.I.S.E. to fleet routing evaluation. We have pointed out that each longitudinal leg actually refines a route network configuration to the last details such as departure time. The schedule refinement loop can therefore be thought of as an extension of the evaluation step in R.I.S.E. It carries the evaluation of a route network alternative to completeness. The vertical suspension from the horizontal structure therefore (i) contributes toward a judicious route improvement algorithm, and (ii) finalizes the route network to a schedule timetable.

1.5 Summary

This chapter has been written as a self-contained overview of the dissertation. In the current section, we will recapitulate the salient points made in the chapter, and, for that matter, the whole thesis. A more formal summary of contributions is deferred until the final chapter, where conclusions and extensions are discussed.

In this piece of research, a particular routing problem in the general class of routing and scheduling problems is addressed. We call it route network configuration. A profit-oriented airline is concerned with the question of how to align non-stop/multi-stop routes to the preferred passenger routings in its network. The service offered by the airline has to be acceptable in the given regulatory and competitive environment.

Such a problem is formulated in an optimization framework, where the airline is maximizing system profit -- i.e., system/prescriptive optimizing -- while the passenger is minimizing his individual travel time -- i.e., user/descriptive optimizing. The optimization formulation can be conceptually distinguished into two parts. First, the feasible set of non-stop/multi-stop routes is defined within the space spanned, or generated, by a "contiguity matrix." Second, the "optimal" subset of routes is to be selected from the feasible set in an integer program. The integer program has a discontinuous objective function. The constraints are made up of node-arc incidence matrices and a set covering matrix. Because of the huge number of feasible routes that could be generated in the combinatorial space, the optimization solution cannot be treated as two disjointed problems of

generation vs. selection. Rather, a combination of generation and selection steps have to be carried out numerous times -- each time exploring a certain part of the feasible region. In other words, a handful of promising route candidates is identified each time and an optimal choice is made out of these few candidates. In order to do this, we have to first manipulate the original integer program into a decomposable form. Node-arc incidence matrices are transformed into arc-chain incidence matrices. The decomposable formulation is made up of a master program and a number of subprograms.

Motivated by the solution strategy, the name R.I.S.E., standing for Route Improvement, Synthesis and Evaluation, has been given to our optimization model. Repetitive application of synthesis and evaluation steps is used to incrementally improve on the solution. The most promising route candidates are synthesized via graph-theoretic techniques. The route candidates are then evaluated in terms of their marginal profits. The combination of synthesis and evaluation steps suggests the most promising feasible direction for route network improvement.

The solution strategy can be conceptualized in a graph of trees. Raising the contiguity matrix serves as the scheme to synthesize, or generate, the trees in the graph. Multi-commodity flow computation at certain nodes of the graph serves as the algebraic method to evaluate the alternative branches of the tree -- resulting in the selection of certain chains in the graph as the solution subset of routes. The enumerative tree-building process also yields minimum paths (routings) to be used in the descrip-

tive traffic flow computation. The graph of trees allows for rapid scanning to see that the set covering constraints are satisfied, thus ensuring the specified level of connectivity between city pairs.

The R.I.S.E. algorithm is a primal decomposition procedure. It maintains a primally feasible solution as the iteration is carried out between the subprograms and the master program. Each origin-destination subprogram boils down to the solution of a minimum-path problem. These minimum chains (routes) synthesized by the subprograms are appended in the master program for evaluation. The evaluation yields a marginal profit for each route. Selection of the route to improve the route network is made on the basis of these marginal profits.

The R.I.S.E. algorithm is a stagewise dynamic programming procedure, where non-stops are examined prior to one-stops, and one-stops prior to two-stops, etc. The solution trees of the dynamic program are trimmed by branch-and-bound techniques. The method of successive approximation is employed to label the graph of trees repetitively until two successive sets of labels remain unchanged. If a good existing route network is available to be used as the "initial policy" in the algorithm, the rate of convergence will be enhanced.

The R.I.S.E. model fits readily into a simul-sequential decomposition framework for the schedule planning process. In this framework, the dimensionality of schedule planning is broken down serially into sequential stages/levels. The system interlocking effects, or network effects, are coordinated by simultaneous looping. The simul-sequential approach is mo-

tivated by the nature of the routing and scheduling problem, where there is a sequential hierarchy "chain of command" as well as a feedback between the various levels.

R.I.S.E. serves as the backbone of the simul-sequential analysis. Routes are aligned with preferred passenger routings in a supply/demand equilibrium seeking fashion. The level of service feedback loop is exactly the improvement step consisting of synthesis and evaluation phases. Other scheduling models such as the fleet/frequency assignment and fleet routing model can be "hung" onto the backbone as extensions of the evaluation phase. The series of tagged-on models refines the route network to a finalized schedule with the help of the schedule refinement loop among themselves. Notice that this refined evaluation step can be performed as often or as seldom as the computation time and budget would permit. An infrequent application of the schedule refinement procedure yields a suboptimal, but nevertheless improved, solution. Furthermore, since R.I.S.E. uses a primal method, the algorithm can be stopped before completion -- where a feasible, improved, though not optimal, schedule is always guaranteed.

REFERENCES

1. Balinski, M. L. "Integer Programming: Methods, Uses, Computation," Management Science, vol. 12, no. 3, November, 1965.
2. Bellman, R. E.; Cooke, K. L., and J. A. Lockett. Algorithms, Graphs and Computers, Academic Press, 1970.
3. Chan, Y. P. "Optimal Travel Time Reduction in a Transport Network: An Application of Network Aggregation and Branch-and-Bound Techniques," Department of Civil Engineering, M.I.T., Research Report R69-39, July, 1969.
4. Chan, Y. P.; Kershner, D., and B. Kullman. "Fleet Routing and Scheduling Model," term project for Course 16.77 at M.I.T., May, 1970.
5. Dantzig, G. B., and P. Wolfe. "The Decomposition Algorithm for Linear Programs," Econometrica, vol. 29, pp. 767-778, 1961.
6. Dreyfus, S. E. "An Appraisal of Some Shortest Path Algorithms," Operations Research, vol. 17, pp. 395-412, May, 1969.
7. Elmaghraby, S. E. "The Theory of Networks and Management Science," (Parts I and II) Management Science, vol. 17, nos. 1 and 2, September and October, 1970.
8. Fulkerson, D. R., and L. S. Shapley. "Minimal k-arc-Connected Graphs," The RAND Corporation, P-2371, 1961.
9. Gagnon, G. G. "A Model for Flowing Passengers over Airline Networks," Transportation Science, vol. 1, no. 3, August, 1967.
10. Geoffrion, A. M. "Elements of Large-Scale Mathematical Programming," Management Science, vol. 16, no. 11, July, 1970.
11. Gunn, W. A. "Airline System Simulation," Operations Research, vol. 12, March, 1964.
12. Held, M., and R. M. Karp. "The Travelling-Salesman Problem and Minimum Spanning Trees," Operations Research, vol. 18, no. 6, 1970.
13. Holloway, C. A. "A Mathematical Programming Approach to Identification and Optimization of Complex Systems," Western Management Science Institute, working paper no. 161, University of California, Los Angeles, August, 1970.

14. Jarvis, J. J. "On the Equivalence Between the Node-Arc and Arc-Chain Formulations for the Multi-Commodity Maximal Flow Problem" Naval Research Logistics Quarterly, vol. 16, no. 4, pp. 525-529, December, 1969.
15. Jessiman, W. A.; Basile, J.; Ward, D. E., and D. P. Maxfield. "Inter-city Transportation Effectiveness Model," paper presented at the 37th National Meeting of O.R.S.A., April, 1970.
16. Jorgensen, N. O. "Some Aspects of the Urban Traffic Assignment Problem," I.T.T.E. Graduate Report, University of California, Richmond, California, July, 1963.
17. Kingsley, Joe D. "Airline Operations and Planning Model," Aircraft Division, McDonnell Douglas, Long Beach, California, Douglas Paper 5058, May, 1968.
18. Levin, A. "Fleet Routing and Scheduling Problems for Air Transportation Systems," Report No. FTL R68-5, M.I.T. Department of Aeronautics and Astronautics, January, 1969.
19. Mesarovic, M.; Macko, D.; and Y. Takahara. Theory of Hierarchical, Multi-Level Systems, Academic Press, 1970.
20. Ruiter, E. R. "A Prototype Analysis: Search and Choice in Transportation Systems Planning," vol. II, M.I.T. Department of Civil Engineering, 1968.
21. Simpson, R. W. "Scheduling and Routing Models for Airline Systems," FTL Report R68-3, Department of Aeronautics and Astronautics, M.I.T., December, 1969.
22. Steiglitz, K., and J. Bruno. "A New Derivation of Frisch's Algorithm for Calculating Vertex-Pair Connectivity," BIT, vol. 11, no. 1, pp. 94-106, 1971.
23. Tomlin, J. A. "Minimum-Cost Multi-Commodity Network Flows," Operations Research, vol. 14, no. 1, 1966.
24. U.S. Bureau of Public Roads. "Electronic Computer Program for Assignment of Traffic to Street and Freeway Systems," undated.
25. Vandersypen, H. private communications, The Transportation Center, Northwestern University, 1971.
26. Wagner, H. M. Principles of Operations Research, Prentice-Hall, 1969.

27. Wardrop, J. G. "Some Theoretical Aspects of Road Traffic Research," Proc. Inst. Civil Engineers, part 2, pp. 325-378, 1952.
28. Wollmer, R. D. "Multi-Commodity Supply and Transportation Networks with Resource Constraints: The Generalized Multi-Commodity Flow Problem," RAND Memorandum RM61-13-PR, March, 1970.

CHAPTER 2

MODELLING THE AIRLINE FIRM AND INDUSTRY

In order to construct a meaningful mathematical model of a managerial or socio-economic system, the context from which the problem arises has to be clearly understood. R.I.S.E.* models an airline operator. An airline is an economic firm competing with other airlines to serve the public demand for travel or shipping. Air transportation is a regulated industry in the U.S. The Civil Aeronautics Board (C.A.B.) and the Federal Aviation Administration (F.A.A.) have been delegated the regulatory powers. In this chapter, we will provide a thumbnail sketch of the regulatory/economic environment in which the airline firm operates, the behavioral observations of the travelling public, the institutional/managerial constraints that face the airline, and finally the airline as a profit-oriented firm. To each of these managerial or socio-economic factors, we offer an approach to quantify and represent them in the state-of-the-art and/or research-extended mathematical tools. This chapter serves as a link between the problem statement and subsequent model formulation/solution. It is written to place the systems analysis techniques in their proper perspective, so that the trees will not be taken for the forest.

*R.I.S.E. stands for Route Improvement, Synthesis and Evaluation. It is an optimization model for route network planning.

2.1 The Airline Industry

In this section, we will outline the regulatory, competition and behavioral aspects of the U.S. domestic trunk industry in which an airline firm operates. The section that follows will then concentrate on the individual airline firm.

The impact of the regulatory and competitive industrial environment on the performance of the airline operators is most profound. The impact is dramatized by a comparative study of the U.S. vs. the Western European airline industries [E.A.R.B. - 1970]. As a general statement, Western Europe is less than half of the size of the U.S., yet European airlines operate close to twice the route miles of the U.S., at twice the average operating cost. Such difference in performance can be attributed to a number of socio-economic and cultural factors. Regulation and market structure, however, play an important part in bringing about this dramatic difference [E.A.R.B. - 1970, Wheatcroft - 1956].

A systems analysis study which brushes off lightly a realistic modeling of the managerial, socio-economic and institutional constraints, or makes idealized assumptions about them, has doubtful pragmatic use. There has been some expressed skepticism,* for example, on the conclusion of R.E. Miller's [1963] work on the U.S. domestic airlines. In his work, Miller indicated that the U.S. domestic traffic in 1957 could have been served at under half of the actual reported cost, given (1) a perfectly competitive

*See the work of J.C. Miller III [1968], which has expressed some reservation about R.E. Miller's conclusion.

market, (2) a perfectly inelastic demand, (3) no indivisibility of production, and (4) an exogenously fixed load factor of 60%. The contribution of Miller is much more of a political economics nature than those made by this dissertation. It has to be noted that Miller's thesis is on the exemplification of the efficiency in pure competition. We, on the other hand, try to point out what managerial improvement can be made recognizing the regulatory, competitive and institutional constraints that partially tie the hands of the airlines.

2.1.1 Regulatory Environment

The U.S. domestic airline industry is regulated in the economic sphere by the C.A.B. and safety-wise by the F.A.A. We address only economic regulation in this dissertation. The C.A.B. has the responsibility of determining the fare and routes granted to airlines. A specific fare level places the air transportation industry at a certain competitive position with other modes of transportation, such as buses, private automobiles and trains. Route authorization, on the other hand, controls the entry of new carriers and regulates the competition between carriers in a number of city pair markets.

This dissertation will concentrate on discussing the route authorities, while taking the fare as given. In order to operate in a certain city pair market, an airline has to be granted a Certificate of Public Convenience and Necessity, which spells out in specific terms how and where the airline is allowed to provide service. Because of the way the C.A.B. regulates the routes, the U.S. airline route structure is "linear"

in configuration. By linear we mean that an airline is usually authorized to fly a linear string of cities. When an airline is granted such a route certificate, it is allowed to fly the string of cities in a large number of ways. Generally, the airline can fly non-stop or multi-stop between any city pairs in the string of cities, and can originate or terminate at any city in the string.* Combinatorially, there is a great deal of freedom on how the airline can serve the string of cities once the certificate is authorized. In comparison, the route structure in Europe is "star shaped," in which non-stop hops emanate from the hub capital city to a number of foreign capital cities. This is a result of the bilateral agreement between nations. Generally, there is less freedom compared to the linear route structure of the U.S., to plan different route configurations. The two different route regulatory system is attributed as one of the factors causing the dramatically higher cost of operation in Europe [Wheatcroft - 1956, E.A.R.B. - 1970]. A linear route structure allows multi-stop flights to be planned, whereby the passenger flow can be channelled in such a way that the seats in each segment of the flight itinerary could be filled. Traffic flow can be bundled up into high density operations by careful multi-stop route planning to the extent that airline stations are economically utilized. A star shape route structure rules out, to a large extent, the freedom of planning multi-stop routes and hence puts European airlines

*There are exceptions and complications about this general statement. A fuller discussion on the Certificate of Public Convenience and Necessity is given in Chapter 3.

at a cost disadvantage.

The practices of the C.A.B. have given the U.S. airlines even more freedom in planning their schedules than what the formal specification of the route certificate would suggest. Historically, there have been very few cases where the C.A.B. strictly enforced the minimum service requirement [Richmond - 1961]. As a general statement, the C.A.B. has been regulating the airlines on an approval or disapproval basis when a route application/fare increase case is brought to its attention. Rarely does the Board, out of its own initiative, assume the active role of prosecuting an operator, for example, for not providing the minimum required frequency between a city pair. The airlines, therefore, could conceivably be able to offer no service between cities where they are authorized or required to do so. In Fig. 2-1-1, we show both the authorized non-stop routes out of each city in the American Airline system in summer, 1970, and those that are actually served. It is observed that only 41% of the authorized non-stops originating from a city are scheduled. Thus it can be seen that within the formal route authorization, there is a good deal of flexibility left to the airlines to configure their routes.

In performing systems analysis for an airline, it is important to be aware of all the route planning alternatives, from which we will select those routing alternatives that best serve our purpose. In order to expedite generating these alternatives, we have quantified the qualitative statements of the Route Certificate in graph-theoretic manner. We have succeeded in representing the route authority of an airline in a research-

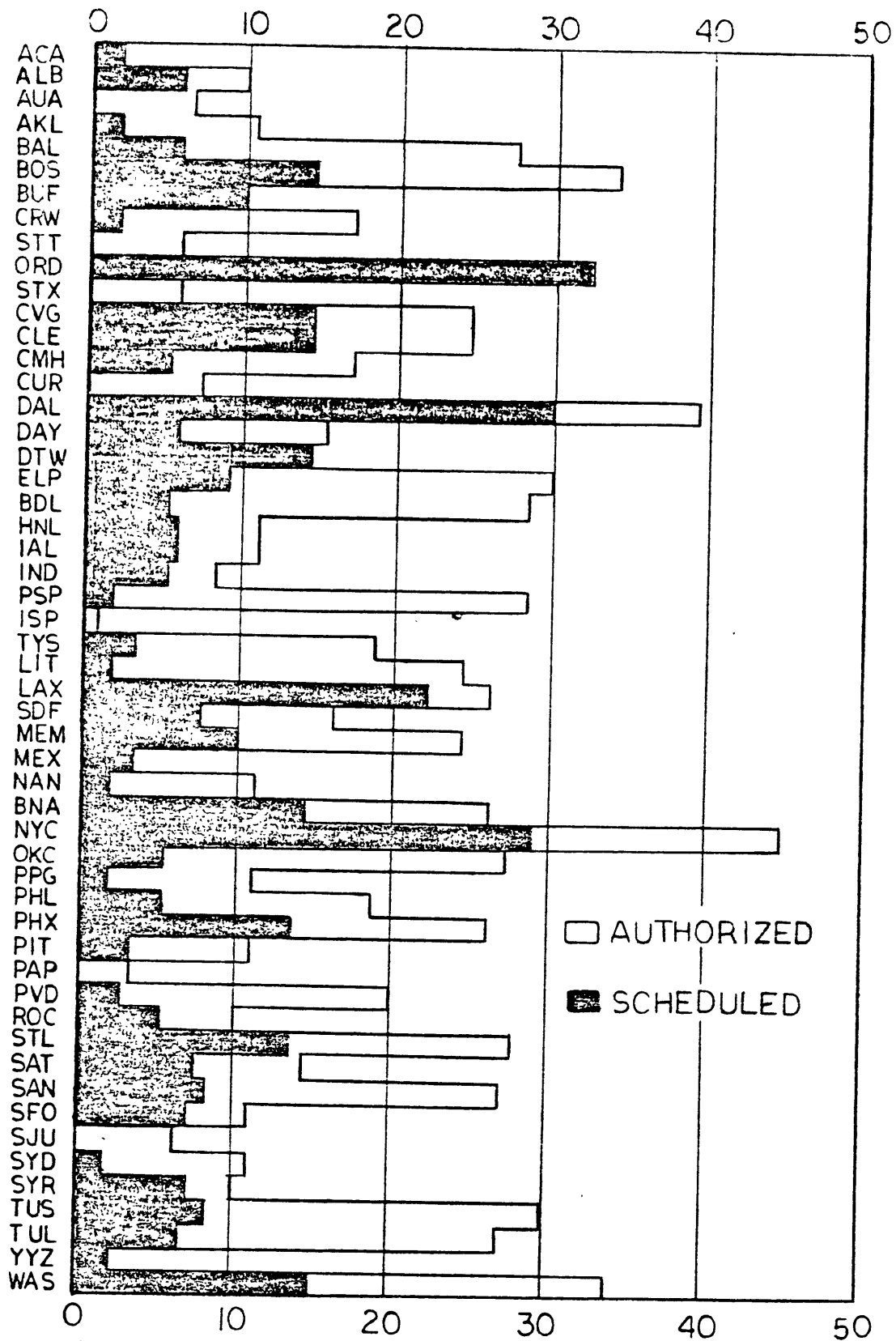


FIG. 2-1-1 NUMBER OF NON-STOP ROUTES BY ORIGIN CITY

extended "contiguity matrix,"* A. Through pure algebraic manipulation, viz. raising the matrix to its own power (A^2 , A^3 , etc.), we would generate all the different route configurations within the confines of route authorization. Through such a graph-algebraic technique, we have modelled the regulatory constraint facing the airline firm. A detailed discussion of the contiguity matrix is deferred until Chapter 3.

2.1.2 Competition Environment

Airlines compete with other modes of transportation in providing and inducing the movement of passengers and goods. Within the U.S. domestic air transportation market, airlines compete with each other for traffic. In this piece of research, we focus our attention on inter-airline competition, taking two competitors at a time.

The U.S. domestic trunks operate in an oligopoly market. A cartel arrangement is set up among the eleven trunklines. The fare is fixed by the C.A.B. at a certain percentage above the cost of providing the service -- a percentage the C.A.B. deems to be the fair return on investment. The entry into a city pair market by an "outside" airline has to be approved by the Board. The airline firms already serving the market compete for profit. Each airline in this case tries to offer a different type of service so as to attract certain sectors of the travelling public. In economic jargon,

*The contiguity matrix is a variant of the adjacency matrix. It records the paths, rather than counting the number of paths, as it is raised to its own power. The term "contiguity matrix" is coined by the author, since he is not aware of a formal name given to this type of matrix.

competing firms try to differentiate their products in non-price competition. The oligopolistic market structure is quite unique to the U.S. domestic operations. In Europe, for example, many of the carriers are flag bearers of their nations. A European carrier may be heavily subsidized by the government just to "show the flag around." In this case, non-economic factors complicate the inter-airline competition picture.

Broadly speaking, the U.S. domestic trunks compete with each other in three ways [Richmond - 1961]. First, they make themselves appealing to different classes of travellers via a labyrinth of different promotional fare structures. Second, they compete with each other in operational/managerial efficiency. Thirdly, they differentiate their "product" (or, more appropriately, their service) via schedule competition, equipment competition and in-flight service competition. In this dissertation, we shall confine our discussion to efficiency and schedule competition only. As such, the applicability of our model would conceivably be limited to the business traveller who usually patronizes the airline that offers the most convenient schedule.

Let us define what we mean by efficiency competition and schedule competition. An efficiently managed firm commands a more superior position in the inter-airline competition picture. For example, an efficient, low-cost operator can afford to offer a better level of service, which tends to result in higher patronage. An efficient, low-cost operator is likely to yield a higher profit. And a healthy record of corporate profit certainly helps in raising new capital for expansion or innovation -- which,

in the long run, again places the efficient firm at an upper hand vis-a-vis competing airlines.

Schedule competition is regarded as an important form of competition actively manipulated in the industry today [Miller - 1968, Richmond - 1961]. There are three types of schedule competition -- departure time, frequency and route. A scheduled departure at prime hours is likely to draw more passengers than one at off-peak hours. A higher daily frequency of departures results in a larger share of the traffic. Finally, a non-stop flight between a city pair is a more favorable service than a multi-stop or connect service. In our model, we address only route competition. Route competition is a good deal more involved than it may at first appear. It is comparatively straight-forward to talk about route competition for a city pair alone, setting aside the rest of the network. Route competition should be viewed as a network problem. Two airlines do not compete only between city pair X-Y; they compete over all the other city pairs. Route competition is actually network competition. Notice a city pair X-Y can be served non-stop by a segment of a one-stop route X-Y-Z, or it could be served by a simple non-stop route X-Y. Say airline A is offering route X-Y-Z to cover city pair X-Y non-stop, while airline B is offering the route X-Y to cover the same city pair. If we narrow our attention to city pair X-Y alone, the two airlines are offering comparable route service (i.e., both offering non-stop). Taking network effects into account, airline A's X-Y-Z route serves not only X-Y non-stop, but also Y-Z, and furthermore it serves X-Z via one-stop. Viewed in this light, airline A's X-Y-Z route appears to command a better competitive advantage over airline B's X-Y route. It can be seen that route

competition is not only city pair route competition; it is actually route network competition. Network effects (or system effects) are one of our main concerns in inter-airline route competition.

The oligopoly market is an area where micro-economic theory has so far been unsuccessful in describing the whole picture analytically [Bishop - 1968]. We recognize the complexity of the oligopoly market and have adopted an approach which takes only two competitors at a time. While airline A is examined, the route structure and schedule of the rest of the carriers are held fixed. Then we look at carrier B, with airline A pooled into the camp of competitors whose schedules are now held fixed. In this manner, R.I.S.E. is applied to an airline firm at a time, ceteris paribus, so as to simulate the trajectory of a series of partial equilibria in oligopolistic competition.

The route competition pressure faced by an individual airline is expressed in a set covering matrix, B . On the row of this set covering matrix are the city pairs. The columns of this matrix are routes. By expressing each row of the matrix as an inequality or equality, a city pair is specified by the user to be covered by a non-stop, multi-stop, or connect route (i.e., $B \geq \underline{1}$). The set covering formulation quantifies what used to be a qualitative route competition statement. We would defer a detailed discussion of the set covering formulation until Chapter 3.

2.1.3 Passenger Travel Behavior

In the last two sections, we have discussed the regulatory and competitive environment facing an airline firm. Now we will turn our attention

to the travelling public (or the "consumers" or air travel) that we would like to point out.

First, we recognize that demand for travel is a function of the level of service. A higher level of service would result in a larger patronage, while a lower level of service would result in a smaller patronage. Therefore, there is a distinction between the potential demand and the actually realized demand. The potential demand is expressed as a demand function. In R.I.S.E., demand is expressed as a function of the level of service of the route -- i.e., whether the routing is non-stop, multi-stop or connect. A non-stop service draws a higher demand than a multi-stop, and a multi-stop attracts a larger ridership than a comparable connection. The demand function assumes the form of a "bar chart," which will be discussed in detail in section 3.2.3 of Chapter 3, where we present the formulation of the mathematical program. Given the potential demand function, the realized demand is the number of passengers that actually travel for a particular routing level of service.

Second, we recognize that for each passenger that travels (and we focus on the average business traveller), he wants to go from origin to destination in the most expedient manner. We interpret 'expedient' to mean the shortest time and least stop/transfer path. Such a behavioral assumption about the way passengers flow in a network is called the "descriptive" traffic assignment [Hershendorfer - 1966, Dafermos - 1970]. Descriptive flow describes, or simulates, the travel behavior of passengers. It is to be distinguished from the "prescriptive" flow assumption where the decision-

maker prescribes the traffic distribution in such a manner so as to achieve a certain normative goal (such as maximizing profit to the airline or maximizing aggregate social welfare). R.I.S.E. has adopted the descriptive approach in order to realistically model, or simulate, the observed traffic pattern. In doing so, we have to put minimum operators within a maximum overall operator in the objective function, which looks like $\max\{\min(\cdot), \min(\cdot), \dots\}$. For the realism we purport to include in R.I.S.E., we pay the price of solving a more involved mathematical program.

To quantify the "descriptive" behavioral observations of passenger flow, a matrix min-path method is used to assign traffic to the shortest paths from all origins to all destinations. The matrix used is exactly the "contiguity matrix," A , used in quantifying the C.A.B. route authorizations. By raising the matrix to its powers, A^2 , A^3 , ..., not only are authorized routes generated, passenger routings and the shortest paths are also available. Section 3.1.4 in the next chapter will elaborate on the subject. Passenger traffic assignment is formulated as a multi-copy (or multi-commodity) flow problem with each of its block diagonal subproblems representing an origin-destination copy (or commodity). For a schematic illustration of this, please review Fig. 1-2-4.

2.2 The Airline Firm

In the last few sections under 2.1, we have outlined the industrial environment in which an airline firm operates. In the current section, we will concentrate on the airline firm. The practices and institutional

characteristics specific to an airline will be introduced. Then we will present a model, R.I.S.E., constructed as a representation of an airline firm. The background discussion on the managerial and economic aspects of an airline is intended to help the reader to place our model in proper perspective.

2.2.1 The Firm and Its Market

The organization of this section is as follows. First, we will discuss the "product market" in which the airline sells its service. Then we point out the "production process" by which an airline produces its service. This is followed by a statement of the corporate objective of an air carrier. We conclude with a scheme to compute the demand/supply equilibrium for the given product market and production process, under the stated corporate objective.

product market

We have pointed out that the airline firm operates in an oligopoly market. External factors such as regulation, inter-airline competition and the behavior of the travelling public all contribute towards defining a particular product market which the airline firm faces.

The "product" an airline "sells" is not really a product in its usual sense of the word. Economists, who like to view an airline as an example of a general "firm," have been using seat-miles, or ton-miles, as a measure of the "product" of an airline. Practitioners of the airline business, however, like to think in terms of an airline offering a service. A proposed unit to quantify this type of travelling service is seat-departure.

We tend to endorse the latter point of view. The level of service that attracts passengers cannot be characterized adequately by a capacity figure such as seat-miles. A jumbo jet of, say, 300 seats scheduled only once a day simply does not have the same appeal as a 100-seat B-727 with three departures a day, although both offer the same number of seat-miles. A unit like seat-departure takes into account the appeal of departure frequency besides capacity. It is a more appropriate unit for measuring output in the product market. We have included in the unit the notion that demand is responsive to the frequency level of service.

In this dissertation it is recognized that an airline sells in a "product" market where the passenger patronage is a function of level of service offered by the airline. R.I.S.E. uses a rather simplified representation of the "product market." Origin-destination demand for an airline is a function of whether the service is non-stop, multi-stop or connect. Other more detailed levels of service attributes like frequency will be an explanatory factor for demand market share in a lower level model in our scheduling hierarchy. R.I.S.E., as an aggregate model in the hierarchy, makes decisions only on how a non-stop vs. multi-stop vs. connect service would affect demand.

To complete the description on the product market an airline faces, we have to point out how revenue is derived by an airline via selling its product (actually, service). Each passenger that 'buys' a seat brings in a 'yield'* to the airline. Yield per passenger is therefore our simplified

*Yield per passenger, in airline practices, is the actual revenue from each passenger after various promotional fare discounts have been taken into

representation of the 'product value' (a service value). In section 2.2.3, a set of revenue functions and yield figures will be introduced, which will serve as our straight-forward product value model.

production process

There are a number of characteristics found in the production process of an airline firm. First is the indivisibility of production. A unit of production (or service) is a flight. When a flight is scheduled, a quantum of seats (say a 100-seat B-727) is offered. An airline does not conduct business by selling an individual, single seat. The basic unit of production is a discrete quantity of seat-departures. The cost of operating the scheduled flight is essentially the same whether the flight is filled to capacity or nearly empty. The revenue from a flight, however, is highly dependent on the number of passengers carried on board [A.T.A. - 1961]. Such an indivisibility feature requires R.I.S.E. to take on 0-1 type binary variables. We have to resort to integer programming techniques in our optimization model.

A second complication about the production process of an airline firm is the "network effects" inherent in scheduling. A route that goes from A to B to C carries not only the "local traffic" from A to B in segment A-B, but also "through traffic" from A to C, and possibly connect traffic from cities other than A, B or C. In such a context, when a schedule planner considers whether or not to schedule a route, he must include in the marginal revenue not only the yield from the passengers that travel between

account.

the three cities A, B and C, but also the connect traffic. In estimating marginal cost for the route, network effects also come in. The cost of serving the route depends on the number of times the route is flown (i.e., the route frequency). Route frequency is determined as a function of the traffic density on each segment of the route. And traffic density, being closely related to the passenger flow, is highly sensitive to the route network configuration. When a route is added or deleted from a network, the passenger flow pattern may be disturbed to such an extent that it requires us to re-examine the assigned frequencies to all the other routes. When route frequency is changed, so is the marginal cost of the route. There is this particular network effect inherent in the production process of an airline firm. R.I.S.E. is a network model specifically designed to take care of the network effect complications in marginal analysis.

corporate objective

Among a number of objectives an airline may have, profit is one of the prime considerations. There are a number of reasons why profit is of concern to the airline. Similar to any other corporate business, the source of funds is vital to the airlines. Retained earnings (profit) is a ready internal source of funds. Other internal sources of capital such as depreciation and amortization, deferred taxes and investment tax credit all depend on the reported earnings figure for their ultimate realization [Varga - 1970].

The eleven trunks had combined assets of \$10 billion in 1969, of which 83% represented gross value of aircrafts. The financing of the fleet

mainly comes from external sources such as banks, stocks and the aircraft manufacturers. The funds are not only used for new aircrafts. Much of it is used to sustain debt that the carrier has previously incurred. Most of the financiers of external funds to an airline would tend to base their decisions largely on the profitability of the business in which they invest their money. Profit is therefore also critically important because of its impact on the availability of outside funds.

Finally, profit is important for competitive reasons. We have pointed out in section 2.1.2 that one type of inter-airline competition is operational/managerial efficiency. An efficiently run airline can afford to offer better service at a lower cost and higher yield. It has an edge over its competitors. For all the above reasons, R.I.S.E. has taken profit maximization as the objective function.

demand/supply equilibrium

Many transportation analysts of today like to think in the conceptual framework of supply and demand. In our context, on the supply side is the production process of an airline offering the transportation connectivities between city pairs. On the demand side is the product market in which passengers purchase the offered service -- passengers who need to travel between origin and destination pairs. There has been a wealth of scheduling research accumulated over the years. Much of it, as pointed out in Chapter 1, really consists of submodels in the sense that they deal with subsets of the demand issues or supply issues. The application of these submodels yields non-equilibrium solutions. Models like R.I.S.E. that address both

supply and demand to bring about supply/demand equilibrium solutions, are relatively scarce. R.I.S.E. takes into account both the product market that faces the airline and its production process. We can visualize the demand between origin-destination pairs as the 'desire lines' of travel, and the airline trying to supply route connectivities to match the desire lines between city pairs, so that high route load factors would result. What complicates the picture is that the desire lines are ever-changing, depending on the connectivity level of service supplied by the airline. R.I.S.E. reflects the dynamics of equilibrating demand and supply in that it utilizes an iterative computational scheme. The equilibrating or optimizing algorithm, which employs dynamic programming and gradient search techniques, will be described in Chapter 4.

2.2.2 Short-Run vs. Long-Run Analysis

In Chapter 1, we have categorized scheduling models in a hierarchical structure. Those in the macro-level include models of fleet planning, route structure configuration, etc. Those in the detailed level include models for dispatching and schedule control. In the short run, we may find fleet size, routes and even daily frequency fixed. In the long run, new fleet can be acquired, new routes could be awarded by the C.A.B. and daily frequency can be increased or decreased.

The economic concept of short-run vs. long-run schedule planning goes hand in hand with our idea of multi-level optimization. In the macro-level of schedule optimization, the more aggregate decisions are made with minimal constraints, since the usual constraints like fleet size can be expand-

ed or shrunk in the long run. After these high level decisions have been made, they will be input as fixed constraints for the lower level decision models, which assumes a short-run time frame in which fleet size, among other variables, have been fixed by a higher level fleet planning model.

Short-run models are comparatively easier to construct and solve because they are non-equilibrium models in the supply/demand conceptual framework. By non-equilibrium, we mean that these models usually assume an objective function of cost minimization to satisfy a fixed demand, under a given fleet size constraint and frequency pattern. Long-run models, on the other hand, are more beyond the state of the art. Since they are equilibrium models, in which the demand is a function of the level of service, iterative coordination of supply and demand is required to arrive at the solution.

Scheduling models require as input unit cost and revenue figures. These figures are easier to come by for the short-run models. Cost is typically estimated by statistical methods from historical operating data. These historical data are the result of a fixed past operating condition -- i.e., a certain fleet composition, route structure, frequency pattern, etc. If such cost figures are used in short-run models where these past operating conditions are upheld, we expect them to be accurate. In the long run, however, past operating conditions are changed by definition. Historical cost figures may be meaningless in the present-day framework. Reliable cost figures are therefore more difficult to come by for the purpose of long run models.

Short run models have been useful tools in the airline business in the 1960's, for example. During the Sixties, traffic growth, economy and aircraft technology had a definite trend or pattern. There was observable stability in the projection of the economic and technological environment. Practitioners in the airline business had been able to, on a short term basis, predict cost figures from past data. At the turn of the decade and beginning of the Seventies, the U.S. traffic growth has dwindled sharply, and a new fleet of wide-body aircrafts have abruptly increased the unwanted production capacity of all carriers. It is under this time of change that a more basic, scientific, long-run type scheduling model could have significant contributions toward the managerial/planning process. Airlines find it necessary to examine on a more rational basis macro decisions such as fleet planning, route structure and frequency pattern.

R.I.S.E. makes a close examination of the route structure. Such an examination may reveal that a market formerly served by a non-stop may be more profitably served by a one-stop due to the sluggish traffic growth or a changed competitive condition. Alternatively, in a different city pair, a former one-stop route could be upgraded to a non-stop with substantial improvement in service yet negligible strain on fleet availability, since there may be excess fleet capacity when wide-body jets are added to a stagnant traffic. Frequency, likewise, can be adjusted by R.I.S.E., although to a more limited extent. By judicious upgrading or degrading the service via re-configuring route structure and frequency, a large portion of the economic setback could be reversed.

2.2.3 Cost, Revenue and Profit

The purpose of this section is to take a closer look at the production process of an airline firm and try to answer the question, "What parts of the production process determine the profitability of operating an airline?" In more specific terms, we examine the prime determinants of cost/revenues in the scheduling process. An understanding of these factors would allow us to suggest profit maximization schemes.

There are a few items worthy of mention in regard to the production efficiency (or scheduling efficiency) of an airline. Due to the existing pricing scheme and cost structure, long and high density routes are recognized as money makers, while short-hop and low density routes are usually unprofitable. Part of the reasons that the Europeans are operating at a cost almost twice that of the U.S. is related to their comparatively shorter average stage length and the fact that the U.S. traffic density, measured in passenger miles per station, is 4.5 times that of Western Europe [E.A.R.B. - 1970]. The cost disadvantage of short-hop, low density routes can be explained as follows. A portion of an aircraft hop is the 'unproductive' time spent in maneuvering at airport and acceleration/deceleration. The cost associated with this portion of the hop is minimized when spread over the large seat-miles of a long trip, but becomes significant when the trip is short. Moreover, a portion of the total operating cost (over a third) is relatively fixed and can be regarded as an overhead charge incurred. Longer and denser routes would spread this overhead cost.

It is our contention that by judicious arrangement of the route con-

figuration, an unprofitable short segment could be tagged on to a profitable long segment to serve as a feeder. In such a manner, the cost disadvantage of the short leg is compensated by the additional profit from the traffic fed into the long segment. Traffic density, on the other hand, is a result of the passenger flow pattern in the network, which can be controlled by manipulating the route structure and connectivity between origin-destination pairs. Viewed in this light, we believe* that re-examining existing route structure and frequency holds promise to overcome the profit disadvantage of short and low traffic density segments. This contention of ours will be verified by a case study of the American Airlines network system in Chapter 5.

cost

In order to discuss the profit to an airline firm, we have to address ourselves first to cost and revenue. Here we outline how cost is estimated.

It has been the practice of the U.S. airlines to classify costs into direct operating cost (DOC) and indirect operating cost (IOC). DOC is a cost item incurred as a necessary result of and directly related to flying the aircraft. IOC, on the other hand, is incurred in providing operating services on the ground, and the usual overhead associated with administration or management of business. Any classification is to some extent arbitrary. For the purpose of our model, we find it more systematic to categorize cost into fixed vs. variable costs. Whether a cost item is fixed or

*This belief is shared by [Wheatcroft - 1956].

variable depends on the adopted time frame. Conceptually, all costs are 'variable' in the long run. R.I.S.E. is a medium range (i.e., neither short nor long range) model. Based on this time frame, we offer the following definitions for fixed vs. variable cost. Fixed costs refer to the basic system overhead and joint costs associated with the institutional, administrative and promotional functions. They are the costs that R.I.S.E. does not assume control over. Variable costs, on the other hand, refer to the traceable, incremental expenditures in providing an additional service unit (e.g., a route or a flight). They include both aircraft operating cost, aircraft financial costs, and the supporting ground, non-flight operating costs. R.I.S.E. assumes control over optimizing these costs. We will elaborate our definitions on fixed vs. variable costs by reference, wherever possible, to the C.A.B. Form 41 accounts.

Let us start with the fixed system costs. We include in this category the following accounts from IOC and DOC.*

- IOC - reservation and sales
 - general and administrative
 - advertising and publicity
 - servicing administration
 - depreciation of ground property and equipment
 - maintenance, ground property and equipment

*Strictly speaking, the cost of capital for financing ground property and equipment should be included in this list, since R.I.S.E. has a longer time frame than "short run." Practically, however, such an expense is negligible (according to [Miller - 1968], p. 130).

DOC - aircraft maintenance burden

Since the above are basic system costs which R.I.S.E. takes as given, they will be computed for the whole network system in an after-the-fact, or ex post, manner at the completion of the optimization procedure. To facilitate the computation, we include in Table 2*2*1 three regression equations from Taneja and Simpson [1967] for estimating three major items of fixed costs.

We now come to variable cost. Variable cost refers to the incremental cost traceable to an additional service unit (which in our case is a route). Since R.I.S.E. is a medium-range planning model, variable costs refers to items with a longer time frame than the 'out-of-pocket' type expenses. The following are the C.A.B. Form 41 accounts included in our definition of variable cost:

- aircraft financial cost
- DOC - flight operations (fuel and crew)
 - maintenance, flight equipment (minus maintenance burden)
 - ownership, flight equipment (depreciation and insurance)
- IOC - aircraft servicing
 - passenger service (in-flight)
 - traffic servicing

The three DOC items can be aggregated into a fundamental unit cost figure for each aircraft type -- cost per block hour. Cost per block hour appears to be a constant independent of trip length (due to Simpson [1970]). Furthermore, the cost per block hour figure for the common air-

TABLE 2*2*1
ANNUAL SYSTEM EXPENSES
Domestic Carriers (1962 - 1966)

1. Reservation and Sales Cost (\$)

$$= -0.619 \times 10^6 + .00385 \text{ RPM} + .0585 \text{ RM} + 0.360 \text{ RPO}$$

$$R = .989 \quad F = 769 \quad \text{Std. Error} = 2.56 \times 10^6 \text{ (\$/yr.)}$$

2. General and Administrative Cost (\$)

$$= .073 \times 10^6 + .00156 \text{ RPM} + .0483 \text{ RM}$$

$$R = .979 \quad F = 1359 \quad \text{Std. Error} = 1.43 \times 10^6 \text{ (\$/yr.)}$$

3. Advertising and Publicity Costs (\$)

$$= .744 \times 10^6 + .00172 \text{ RPM}$$

$$R = .950 \quad F = 489 \quad \text{Std. Error} = 1.77 \times 10^6 \text{ (\$/yr.)}$$

4. Aircraft Servicing Cost (\$)

$$= -3.55 \times 10^6 + .00421 \text{ RPM} + 37.6 \text{ D}$$

$$R = .933 \quad F = 1456 \quad \text{Std. Error} = 2.09 \times 10^6 \text{ (\$/yr.)}$$

5. Passenger Service Cost (\$)

$$= -0.784 \times 10^6 + .00549 \text{ RPM}$$

$$R = .994 \quad F = 4654 \quad \text{Std. Error} = 1.83 \times 10^6 \text{ (\$/yr.)}$$

6. Traffic Servicing Cost (\$)

$$= -12.96 \times 10^6 + 2.46 \text{ E} + 466000 \text{ E/D}$$

$$R = .888 \quad F = 96.7 \quad \text{Std. Error} = 6.77 \times 10^6 \text{ (\$/yr.)}$$

7. Sum of the Above six accounts

TOTAL IOC (\$)

$$= -7.20 \times 10^6 + .0146 \text{ RPM} + .645 \text{ RM}$$

$$R = .998 \quad F = 4111 \quad \text{Std. Error} = 5.58 \times 10^6 \text{ (\$/yr)}$$

NOTE:

RPM = revenue passenger miles/year

RM = revenue aircraft miles/year

RPO = revenue passenger originations/year

D = aircraft departures/year

E = passengers enplanements/year

SOURCE: Taneja and Simpson [1967]

craft types (fanjets) used in the domestic trunk lines are statistically related to the gross weight of the aircraft [Chan - 1970a]:

$$(\text{Cost per block hour (in \$)}) = 291 + 1.01 \times (\text{Weight of aircraft in thousands of lbs})$$

$$R^2 = .8930 \quad F(1,9) = 75.12$$

For route planning purposes, such an aggregate unit figure is quite adequate for variable cost estimations.*

Part of the output of R.I.S.E. is fleet requirement. R.I.S.E. tells the user the mix and the number of flying hours by aircraft type required to offer the most profitable level of service (in terms of whether a city pair should be served non-stop, multi-stop or connect). In this context, fleet size is regarded as expandable or contractable in the planning horizon of the model. The acquisition of an aircraft involves financial costs to raise the necessary capital, besides the depreciation and hull insurance (jointly called ownership cost included in the DOC). In our definition, financial cost refers to the daily share of interest payments for the use of the equipment purchase capital, while the ownership cost is the daily share of aircraft purchase price. According to Miller [1968, p. 128] the following is an estimate of the daily financial costs:**

*If required, trip cost, trip cost per seat, and cost per seat-mile curves can be derived from the cost per block hour figure provided a block time vs. distance curve is also available. See [Simpson - 1970] or [Chan - 1970b] for details.

**In arriving at these figures, Miller [1968] assumed a 10% rate of return, a 12-year equipment useful life, and a zero scrap value.

4 engine	\$ 1,841
3 engine	\$ 1,080
2 engine	\$ 785

These costs of capital can be added on top of the cost per block hour in arriving at the total "long run" unit cost of operating an aircraft for an hour.

Much of the components of aircraft servicing, passenger service and traffic servicing are salary and joint costs which are not separable among flight A or flight B* [P.R.C. - 1966]. The practical way of estimating these expenses is by statistical regression on a system-wide annual basis (see Table 2*2*1 for typical regression equations). It is rather out of context to ask the traffic servicing personnel how much of his salary applies toward handling passengers for flight #123 vs. flight #456, or toward servicing passenger Mr. A vs. passenger Mr. B. Although we have been trying via a number of ways to attribute these joint costs to individual routes, we have had no success to date. R.I.S.E. is an aggregate model for route planning. And considering aircraft, passenger and traffic servicing amount to only about a third of the cost per block hour figure (in which aircraft financial costs are included), we propose the following way to handle these joint, inseparable costs. In the same manner as we estimate the fixed costs, we would compute the aircraft, passenger and traffic

*Over 85% of traffic servicing, 55% of aircraft servicing and 40% of passenger service are salaries alone, not to mention other types of joint costs (see Chan and Simpson [1971]).

servicing cost on an ex post and annual system expense basis.

Let us summarize the practical way to estimate cost for R.I.S.E. The traceable cost attributable to an individual route is computed from a cost per block hour unit figure which includes in it the cost of financing an aircraft. The rest of the cost is lumped under system annual expense and is to be estimated ex post by equation 7 of Table 2*2*1.

revenue

We will confine our discussion mainly to the fare structure set by the C.A.B. One of the rulings of the nine-phase Domestic Passenger-Fare Investigation is to set the domestic fare at a "cost-plus" scheme. Fare is set at a certain percentage (such as a 'fair return on investment') above the cost of providing the trip.* This cost-plus pricing scheme is commonly found in an oligopoly market. The key rationale behind the rate structure is quoted below [C.A.B. - 1968]:

- Fare should be primarily related to distance
- Fare per mile should decline with distance at a rate generally consistent with the behavior of unit costs
- The structure should be based on jet day coach service
- Fares should be based on non-stop city center to city center great circle distances.

The following structure is recommended for U.S. domestic fares in the find-

*After completing six phases of the nine-phase Domestic Passenger-Fare Investigation, the C.A.B. reached a 12% figure as a reasonable rate of return for investment for the trunklines [Aeronautics and Astronautics - 1970].

TABLE 2*2*2

A PERCENTAGE BREAKDOWN OF AIRLINE EXPENSES

Domestic Trunks - 4th quarter, 1970

	% of total operating cost
Variable, Traceable Costs: - flight operations (fuel and crew)	26.5
- aircraft ownership (depreciation and insurance)	13.6
- passenger service (in-flight)	10.2
- maintenance, flight equipment (minus burden)	9.0
- traffic servicing	8.2
- aircraft servicing	7.3
- aircraft financial cost (interest and tax), equivalent to 6.5% of total operating cost	
Fixed, System Costs: - reservation and sales	8.3
- aircraft maintenance burden	6.5
- general and administrative	3.5
- advertising and publicity	2.4
- depreciation of ground property and equipment	1.9
- maintenance of ground property and equipment	1.5
- servicing administration	.8

ings of the final phase of the Investigation [C.A.B. - 1972]:

$$\begin{aligned}
 (\text{Coach fare})^* &= \$12.00 + 6.28\text{¢}(0 < \text{distance in miles} < 500) \\
 &+ 5.86\text{¢}(500 < \text{distance in miles} \leq 1000) \\
 &+ 5.44\text{¢}(1000 < \text{distance in miles} \leq 1500) \\
 &+ 5.23\text{¢}(1500 < \text{distance in miles} \leq 2000) \\
 &+ 5.02\text{¢}(2000 < \text{distance in miles})
 \end{aligned}$$

First class fare is set at 30% above coach fare. The new rate structure (i) raises the intercept of the fare equation from what existed in 1966, \$ 6.44, to \$ 12.00, and (ii) shows a greater taper in cost per miles in longer stage lengths. Short range carriers are likely to be helped out by the new fare formula.

It is seen that, unlike cost estimation, revenue rates are standardized for the industry. However, the whole maze of promotional, discount, and supplemental fares tends to complicate the picture. For computational purposes, many find it convenient to talk about the actual revenue for passenger, yield, which amounts to about 85% of coach fare on the average.

2.2.4 Marginal Profit Analysis

Profit is revenue minus cost. The total system profit from a route network is the sum of individual profits from each of the routes that make up the route network. Suppose when a route is included in the network only

*The fare formula actually notes that fares are computed from a \$12.00 fixed charge and 6.28¢/mile extra for the first 500 miles, 5.86¢/mile for the second 500 miles, etc. Thus the plot of fare against distance should be a continuous curve over the different 500-mile portions, instead of a discontinuous curve, as some mathematically inclined readers may be led to believe.

when it contributes toward increasing the system profit, then the resulting route network would have a promise to be an acceptable one. This is the intuitive idea behind the R.I.S.E. solution algorithm.

How do we compute the marginal profit due to a route? The marginal profit is the passenger fare revenue from the route minus the variable cost traceable to the inclusion of the route in the system. The passenger fare revenue includes the total yield from passengers carried by the route. For example, a one-stop route A-B-C would reap a total revenue from passengers between A and C, and A and B, and B and C, plus connecting passengers. The traceable cost of the route includes the variable costs discussed above. Essentially, it is the block time of the route multiplied by the cost per block hour figure (which contains the aircraft financial cost). As we pointed out before (section 2.2.1), the route cost is practically the same regardless of whether the plane is 10% filled or 90% filled. The revenue, on the other hand, is strongly dependent on the load factor [A.T.A. - 1961]. We can see that marginal profit, being revenue minus cost, is very much a function of how well the seats are occupied. The routing strategy is therefore critically important to ensure picking out the most profitable load of passengers in as many of the segments of the route as possible.

There is a conceptual advantage of including only a route at a time into the route structure. We mentioned earlier (section 2.2.1) that one of the features of the airline firm is the indivisibility of production -- that the airline does not sell a seat mile at a time; it sells a quantum of

seats in a departure. By choosing a route as a basic unit of the optimization procedure, we have used the indivisibility of production feature to the advantage of the algorithm. Also, by analyzing a route at a time, the large network problem is decomposed into a series of smaller subproblems. The decomposition approach is essential for computational feasibility.

The marginal profit concept just outlined is the basis of the profit maximization algorithm used in R.I.S.E. The idea of including in the route network only the route with the greatest marginal profit potential is analogous to the idea of the "method of steepest ascent" used in "hill-climbing" or "gradient search" optimization procedures [Wagner - 1969, Sargent - 1971, Wilde and Beightler - 1967]. R.I.S.E. is a constrained optimization problem. A primal decomposition solution algorithm with dynamic programming flavor will be put forth in Chapter 4 for its solution.

2.3 The R.I.S.E. Model

In sections 2.1 and 2.2 we have discussed the economic and institutional context from which our routing problem arises. We have indicated ways by which a qualitative description of the problem can be quantified. In this section, we will summarize how R.I.S.E. models the routing problem under consideration as a constrained optimization problem. The model assumptions will be recapitulated. This is done to lay the way for a more formal mathematical programming formulation in Chapter 3. We conclude the present chapter by pointing out how the model can be useful in managerial and possibly public sector decision-making.

2.3.1 Regulatory and Competition Constraints

R.I.S.E. is a constrained optimization model of an airline firm. In reviewing how R.I.S.E. models the airline firm and industry, it is convenient to present the summary in terms of the R.I.S.E. constraints and the R.I.S.E. objective function.

Let us start with the model constraints in this section. The route network of an airline is subject to the economic regulation of the C.A.B. In order to enter into serving a city pair market, the airline has to be granted a route certificate, which spells out the specific geographic routing and conditions under which the airline can operate. These route authority constraints have been quantified in R.I.S.E. in a graph-theoretic manner via a "contiguity matrix," A . By raising the matrix to its powers, A^2, A^3, \dots , all the authorized routes as well as passenger routings can be generated.

The U.S. domestic trunks compete with each other in an oligopoly market. Each airline is to some extent constrained by pressure from its competitors. When a city pair is served by a competitive carrier by a non-stop route, for example, the airline under consideration often has to face up to the competition by offering a comparable service in order to partake in the market share. This aspect of route competition is quantified in a set covering tableau, $B \geq 1$. Thus, given a particular route pattern of its competitors, the airline concerned makes its decision based on this existing competitive condition.

The set covering tableau has city pairs on the rows and routes as col-

umns. It is to be noted that these route columns are generated, where needed, by raising the aforementioned contiguity matrix to its powers.

From the combinatorial space viewpoint, there is an astronomical number of route network configurations that will satisfy both the C.A.B. route authority and competition pressure. The job of the R.I.S.E. algorithm is to generate route column only as needed and delete dominated route columns, where appropriate, to converge toward a final optimal route network.

Whenever route columns are added or deleted, a different route network results. The passenger flow from origin to destination is constrained by the particular route network given. The passenger routings are a function of the available aircraft routes. Therefore, as the solution algorithm proceeds, we redistribute traffic. The passenger flow is modelled in the set of constraints as a multi-commodity, or more accurately, multi-copy, problem. There is a block-diagonal subproblem, R_{pq} (i.e., a network copy or a commodity) defined for each origin-destination pair, pq . The set of R.I.S.E. constraints is therefore a combination of a multi-commodity (or multi-copy) flow problem and a set covering problem:

$$\begin{array}{r}
 R_{12} \\
 R_{13} \\
 \dots \\
 R_{pq} \\
 \dots \\
 R_{N-1,N}
 \end{array}
 \begin{array}{l}
 \vdots \\
 = \underline{d} \\
 \vdots \\
 \geq \underline{1}
 \end{array}
 \quad (2:3:1)$$

←----- B ----->

2.3.2 Profit Maximization and Travel Time Minimization

What is the objective function of R.I.S.E.? R.I.S.E. is a mathematical model of an airline firm, which for financial and efficiency reasons would be expected to value profit as one of the prime concerns. An airline is marketing its service to a travelling public, which R.I.S.E. assumes to be time-consciencious travellers, each of whom wants to get to his destination in the shortest path. Passenger routings are simulated by such a 'descriptive' traffic assignment presumption. R.I.S.E. therefore maximizes profit for the airline firm, keeping in mind that each passenger wants to minimize his travel time from origin to destination.

To quantify the profit maximization motive for the airline and the travel time minimization preference of the passengers, we have an involved objective function for the optimization problem. A minimization operator is applied toward the travel time between each origin-destination pair. An overall maximization operator is used for the whole network with all its origin-destinations. This results in a large number of minimization operators rested within a maximization operator:

$$\max\{\min(.) + \min(.) + \dots + \min(.)\}$$

$$\begin{array}{ccc}
 R_{12} & & | \\
 & R_{13} & = \underline{d} \quad (2:3:2) \\
 & \dots & | \\
 & & R_{N-1,N} \\
 \leftarrow B & \dots & \geq \underline{1}
 \end{array}$$

2.3.3 Solution Strategy

Given these constraints and objectives, R.I.S.E. quantifies the complex production process of an airline. It is a medium-range planning model where fleet size is expandable or contractable. The production process of the firm is characterized by the "network effects" -- that is, the passenger flow pattern is constantly perturbed as routes are added or deleted from the route network. An airline is also characterized by the indivisibility of production. In R.I.S.E. we either offer a certain type of route between cities A and B, or not at all. There is no fractional 'product' (or service) like half of a route. As an optimization problem, R.I.S.E. is an integer program.

The airline modelled in R.I.S.E. faces a travel demand function which is responsive to the route level of service -- i.e., whether the passenger routing is a non-stop vs. a multi-stop vs. a connection service. The solution method then matches the authorized routes to the routings preferred by the passengers in order to arrive at a supply/demand equilibrium for the profit-oriented airline firm. Considerations are given to inter-airline competition and C.A.B. route authorization.

An iterative hill-climbing type algorithm is put forth in this dissertation to arrive at the network equilibrium. Each indivisible production unit, a route, is taken as a subproblem. Each time a route subproblem is solved, a reconfiguration of the route network results. Route networks are redefined by the addition or deletion of a route column from the tableau. With each reconfiguration, passenger flow pattern is improved. The

solution of a route subproblem yields a marginal profit to the airline firm. It is based on this marginal return economic concept that we hill-climb to a synthesized route network. And if the synthesized route network is only a local optimum, we proceed with the hill-climb to improve on the solution. All these are carried out in an adaptive, stagewise manner on a large scale network. Alternatively, the existing airline route network can be used as a starting feasible solution. The initial phase I of R.I.S.E. is in this case substituted by using the existing route network. Improvements can then be made on this existing route network via phase II. These procedures are carried out using the method of successive approximation of dynamic programming. The huge network problem is decomposed into a series of route subproblems defined for each origin-destination pair. It is a primal decomposition algorithm, in which primal feasibility is always maintained as the solution is getting closer and closer to optimality. The primal approach has an appealing practical advantage over dual algorithms such as the classic work of Dantzig and Wolfe [1961]. If program execution is stopped before completion due to budget limitations, a suboptimal solution is obtained which is better than the feasible solution we started out with. In most practical applications, only improved solutions, not necessarily optimal solutions, would be adequate, particularly when there are only limited computer resources.

2.4 The Applications of R.I.S.E. in Management and Public Sector Planning

At the time this dissertation research is being carried out, the U.S.

is facing a rather stubborn, stagnant economy. Passenger traffic growth has slowed down from 15% per year in the Sixties to 1.4% in 1970. The introduction of wide-body, large-capacity aircrafts has added a quantum jump in the unwanted seat capacity. Keen non-price competition between carriers in a "lean" market incurs additional strain on the airlines. In the author's opinion, it is a time when more basic, systematic planning is needed for both the carriers and the regulatory agency.

In the previous section, it has been shown how the regulatory/competitive environment and institutional factors have been abstracted and quantified in our model. One of the advantages of model construction is that the model serves as a laboratory in which the consequences of managerial and regulatory decisions can be readily tested.

R.I.S.E. would be a convenient tool in the planning level (such as the corporate planning department) of an airline. It helps to re-examine existing route structure, fleet requirements and route frequency. In this regard, it holds promise to make the best out of the route authority an airline has been granted. By a careful channelization of the traffic flow in a system of feeders and long-haul operations, formerly short-hop, uneconomical segments could become profitable to serve. A traffic flow pattern that results in high route density operations would bring about cost savings to the carrier. In short, systematic route planning minimizes the profit disadvantages of low density and short-hop operations.

The next higher level of the application of R.I.S.E. is obtained through sensitivity analysis. Suppose an airline is considering the merits

of filing an application for a route authorization. The contiguity matrix in which the C.A.B. route authority is encoded can be parametrically changed to represent the inclusion of the route under consideration into the network system. Analysis can then be carried out to assess the cost/effectiveness of the hypothetical new route award.

We can carry the application further to the timely subject of mergers. Obviously, R.I.S.E. cannot help merger decision in its entirety. It can, however, serve as a tool to predict the routing/scheduling implications when two sets of route authorities merged into a conglomerate network system. It may assist in answering questions like: "Could economies of scale in routing/scheduling be expected from the merger?" and "Would there be diminishing marginal returns?"

The above-mentioned route awards and merger decisions obviously concern a regulatory agency like the C.A.B. besides the carriers in consideration. The regulatory agency can conceivably find R.I.S.E. useful. There is an additional application of R.I.S.E. which is of particular interest from the regulation viewpoint. Many economists have used econometric models to assess whether air transportation should be de-regulated. There are stipulations that the existing non-price competition between carriers is introducing inefficiency. The micro-economic theory to explain oligopoly market is far from complete [Bishop - 1968]. R.I.S.E. can be used as a simulation, experimental tool in this regard to predict the effect of different degrees of route competition. This is readily done by parametrically varying the set covering matrix constraints, which quantifies the route

competition pressure exerted on the airline under consideration by the rest of the industry. When the total picture of all the domestic route carriers is wanted, R.I.S.E. can be applied in the following pairwise manner. We run R.I.S.E. on airline A given the existing competition pressure from the rest of the carriers. Then we run R.I.S.E. on airline B with airline A now grouped into the rest of the industry. This repetitive application of R.I.S.E. is a way to empirically trace out the trajectory of the equilibrium points for an oligopolistic competition industry.

If R.I.S.E. is to be useful to a regulatory agency such as the C.A.B., there should be an additional measure of effectiveness besides profit to the carriers. One of the chief guidelines of the C.A.B. has been "public convenience and necessity." In economic jargon, the Board is concerned with the well-being, or welfare, of the users of air transportation. Welfare is often quantified as "consumer surplus" or "willingness to pay." Travel demand is modelled in R.I.S.E. as a function of whether the level of service is a non-stop, multi-stop or connection. Such a formulation of the demand function lends itself conveniently to a measurement of consumer surplus. Together with the profit (or producer surplus) which is explicitly computed as the objective function of R.I.S.E., the C.A.B. has a model that addresses itself to both the well-being of the travelling public and that of the domestic air transportation industry.

REFERENCES

1. Aeronautics and Astronautics, American Institute of Aeronautics and Astronautics, May, 1971.
2. Air Transport Association of America, "Management for Service: The Story of Airline Scheduling," 1961.
3. Bishop, R. L., Printed Lecture Notes for Course 14.121, M.I.T., 1968.
4. Chan, Y. P., "Aircraft Direct Operating Cost - Related to Gross Weight," FTL memo M70-6, Flight Transportation Laboratory, M.I.T., 1970a.
5. Chan, Y. P., "An Outline of Airline Economics," term paper for course 16.77, M.I.T., 1970b.
6. Chan, Y. P., and R. W. Simpson, "A Structure for Airline Operating Costs," FTL memo 71-6, Flight Transportation Laboratory, M.I.T., 1971.
7. Civil Aeronautics Board (Rates Division), "A Study of the Domestic Passenger Air Fare Structure," January, 1968.
8. Civil Aeronautics Board, "Domestic Passenger Fare Investigation, Phase 9 - Fare Structure," Docket 21866-9, served April 7, 1972.
9. Dafermos, S. C., "An Extended Traffic Assignment Model with Applications to Two-Way Traffic," Transportation Science, vol. 5, no. 4, November, 1971.

10. Dantzig, G. B., and P. Wolfe, "The Decomposition Algorithm for Linear Programs," Econometrica, vol. 29, pp. 767-778, 1961.
11. European Airlines Research Bureau, "Comparative Examination of EARB and United States Airlines," EARB/403, April, 1970.
12. Hershendorfer, A. M., "Equilibrium of Supply and Demand," Transportation Research Forum Proceedings, 1966.
13. Miller, J. C., "Scheduling and Airline Efficiency," Ph.D. Thesis, Department of Economics, University of Virginia, June, 1969.
14. Miller, R. E., Domestic Airline Efficiency, M.I.T. Press, 1963.
15. Planning Research Corporation (by A. G. Harvey and H. D. Orenstein), "Operating Cost Estimates for Long Haul Subsonic Jet Aircraft," PRC R-894A, February, 1966.
16. Richmond, S. B., Regulation and Competition in Air Transportation, Columbia University Press, 1961.
17. Sargent, R. W. H., "Minimization Without Constraints," Imperial College of Science and Technology, London. (Presented at the NATO International Summer School, Louvain, Belgium, July 26 - August 6, 1971.)
18. Simpson, R. W., Lecture Notes for Courses 16.751 and 16.752, M.I.T., 1970.
19. Taneja, N. K., and R. W. Simpson, "A Multi-Regression Analysis of Airline Indirect Operating Costs," Flight Transportation Laboratory,

R-67-2, M.I.T., 1968.

20. Varga, G. J., "Financial Methods and Instruments in the Air Transportation Industry," Master's Thesis, Department of Aeronautics and Astronautics, M.I.T., 1970.
21. Wagner, H. M., Principles of Operations Research: With Applications to Managerial Decisions, Prentice-Hall, 1969.
22. Wheatcroft, S., The Economics of European Air Transport, Harvard University Press, 1956.
23. Wilde, D. J., and C. S. Beightler, Foundations of Optimization, Prentice-Hall, 1967.

CHAPTER 3

MATHEMATICAL MODEL FORMULATION OF ROUTE IMPROVEMENT

It has been pointed out in the previous chapters that the route network configuration problem, viewed in a constrained optimization framework, has two parts to it: route generation vs. route selection. Feasible route candidates are generated by graph-theoretic techniques. The candidates are then included in an integer program which selects the best subset of routes. It was recognized that for dimensionality reasons, generation and selection cannot be carried out as two disjointed steps, whereby the comprehensive set of route candidates are synthesized in the generation step, and then subject to a final choice in the selection step. Rather, only a most promising handful of candidates should be identified at a time, to be evaluated simultaneously--resulting in a choice of the route that appears to be optimal at the time. Consecutive generation and selection have to be performed cyclically over a number of times to come up with the optimal route network. These three issues: (i) generation, (ii) selection, and (iii) simultaneous generation and selection will be addressed in this chapter. They will be formulated in a mathematical model with the necessary amount of technical detail.

3.1 Route Generation: A Graph-Theoretic Approach

The route network configuration problem is basically a geometrical exercise of drawing routes between city pairs on the map so as to

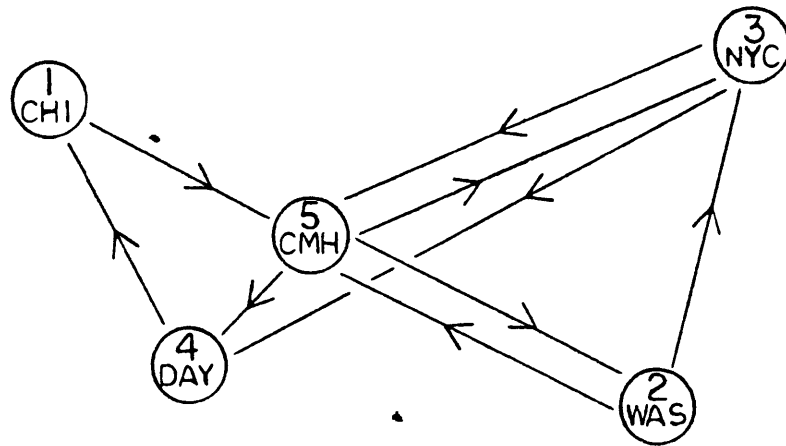
provide the desirable connectivities. There are a number of rules that must be observed in carrying out this exercise. An important rule is that a route must abide by the Civil Aeronautics Board (C.A.B.) route authorities. In this piece of research, we have devised a way of encoding the C.A.B. route authorities in a contiguity matrix via graph-theoretic techniques. The matrix, when raised to its own power, would generate all the routes that are possible within the confines of route authorization.

3.1.1 Configuring a Route Network

It is a well known fact that network topologies can be represented algebraically as matrices [Avondo-Bondino, 1962]. One of such matrices is the adjacency matrix, in which two adjacent nodes connected by a link or arc is recorded as a corresponding entry of "1" in a square matrix with a dimension equal to the number of nodes in the network. Figure 3-1-1 shows an example of such a matrix corresponding to the given network.

If labels are put in the place of the "1"'s to denote the pair of nodes that are connected to each other, as shown below for the same network of Figure 3-1-1, we have written a "contiguity matrix"

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \left[\begin{array}{ccccc} (1,1) & 0 & 0 & 0 & (1,5) \\ 0 & (2,2) & (2,3) & 0 & (2,5) \\ 0 & 0 & (3,3) & (3,4) & (3,5) \\ (4,1) & 0 & 0 & (4,4) & 0 \\ 0 & (5,2) & (5,3) & (5,4) & (5,5) \end{array} \right] \end{matrix}$$



	1	2	3	4	5
1	0	0	1	0	1
2	0	1	0	0	1
3	0	0	1	1	1
4	1	0	0	0	0
5	0	1	1	1	0

FIG. 3-1-1 A NETWORK AND ITS ADJACENCY MATRIX

The contiguity matrix has the nice property that when raised to its second power, all the one-stop routes can be generated, as can be seen below;

$$\begin{array}{c}
 \Lambda^2 = \\
 \begin{array}{cc}
 & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \\
 \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \left[\begin{array}{c}
 (1,1) \cdot (1,1) \\
 0 \\
 (3,4) \cdot (4,1) \\
 (4,1) \cdot (1,1) + (4,4) \cdot (4,1) \\
 (5,4) \cdot (4,1) \\
 (1,5) \cdot (5,2) \\
 (2,2) \cdot (2,2) + (2,3) \cdot (3,3) + (2,5) \cdot (5,2) \\
 (3,5) \cdot (5,2) \\
 0 \\
 (5,5) \cdot (5,2)
 \end{array} \right]
 \end{array}
 \end{array}$$

• (2,3)+(5,5) • (5,3) (5,3) • (3,4)+(5,4) • (4,4)+(5,5) • (5,4) (5,2) • (2,5)+(5,3) • (3,5)+(5,5) • (5

0

(4,4) • (4,4)

(4,1) • (1,5)

• (3,3)+(3,5) • (5,3) (3,3) • (3,4)+(3,4) • (4,4)+(3,5) • (5,4) (3,3) • (3,5)+(3,5) • (5,5)

• (2,3)+(2,5) • (5,3)

(2,3) • (3,4)+(2,5) • (5,3)

(2,2) • (2,5)+(2,3) • (3,5)+(2,5) • (5

(1,5) • (5,3)

(1,5) • (5,4)

(1,5) • (5,5)

3

4

5

If we read "." as "and" and "+" as "or" in the preceding A^2 matrix, it can be verified that city pair 1-4 can be served by a one-stop route composed of the two segments (1,5) and (5,4). Similarly, city pair 2-5 can be served, in the mathematical sense, by a one-stop route composed of a "self-circuit" segment (2,2) and the segment (2,5), or by a route made up of (2,3) and (3,5), or (2,5) and circuit (5,5). The reader can check these algebraic results geometrically in Figure 3-1-1.

Raising A to the third power generates all the two-stop routes, as can be shown for the city pair 2-4:

$$A^3 = A^2 \cdot A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & & 4 & & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \left[\begin{array}{cccccc} - & - & - & & - & - \\ - & - & - & (2,5) \cdot (5,3) \cdot (3,4) + (2,3) \cdot (3,5) \cdot (5,4) & - & - \\ - & - & - & & - & - \\ - & - & - & & - & - \\ - & - & - & & - & - \end{array} \right] \end{matrix}$$

In the above matrix, we have omitted all the entries except (2,4), where only the non-circuitous two-stop routes from 2 to 4 are displayed. There are two alternative routes: 2→5→3→4 or 2→3→5→4.

The above method of representing the topology of route networks in contiguity matrices can be used to quantify the C.A.B. route authorities in an algebraic fashion. This will be the subject of discussion in the following three sections 3.1.2, 3.1.3 and 3.1.4.

3.1.2 C.A.B. Route Authority

The C.A.B. "Certificate of Public Convenience and Necessity" spells out in specific terms the way a carrier is authorized to provide service among a set of cities in a "route". An example of such a Route Certificate for Route 63 of Western Air Lines is given in Appendix A.1. The example is used in the discussions of 3.1.2 and 3.1.3. The author acknowledges having the benefit of the work of Sobel [1969] and the comments of Professor R. W. Simpson for the development of the concepts documented here and the section that follows (3.1.3).

C.A.B. route authority convention

GENERAL GUIDELINES

A convenient "basic building block" to describe route authority is a "subsegment"*. All the city pairs within a subsegment can be served by non-stop flights.

Two or more subsegments make up a "segment". In order to serve a pair of cities on different subsegments, at least one intermediate stop must be made at a city common to both subsegments. The set of cities common to both subsegments are called "subsegment junction points".

Referring to subsegment {A-a} (Figure 3-1-2), any city pair, like between JSB and SLC, or between LGB and PSP..., non-stop flights can be scheduled; similarly for the city pairs in subsegment {A-b}. To fly from SLC to OAK, however, at least one intermediate stop must be made

*a term coined by Sobel [1969] for the purpose of our present discussion.

SEGMENT {A}

SUBSEGMENTS

{A-a}

{A-b}

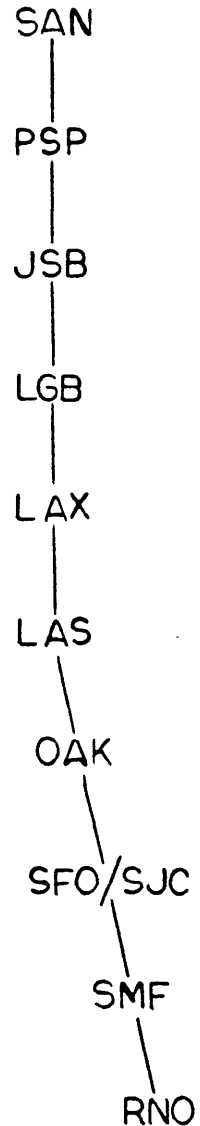
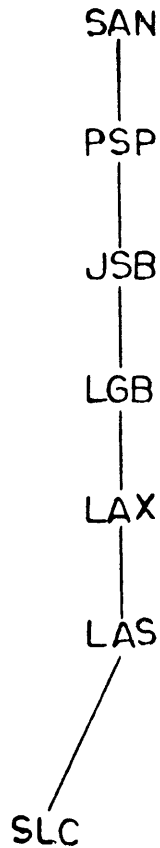
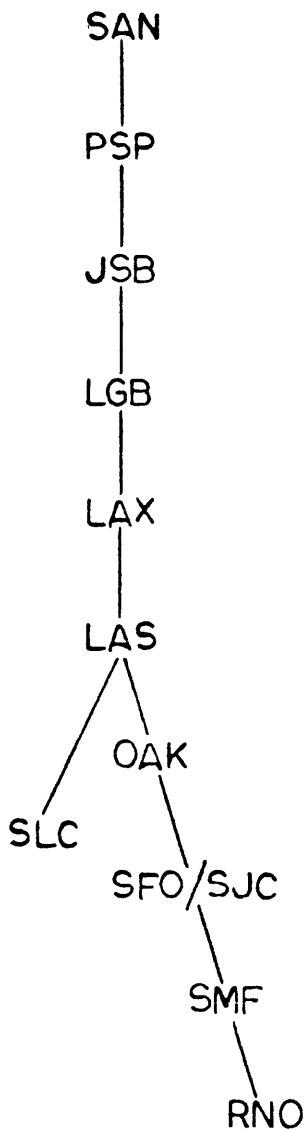


FIG. 3-1-2 EXAMPLES OF SEGMENTS AND SUBSEGMENTS

at one of the set of cities common to both {A-a} and {A-b}, e.g., LAS.

Two segments intersect in "segment junction points." Similar to the case of subsegments, a service between two cities in different segments must serve at least one segment junction point as an intermediate stop.

One or more segments make up a route. Services between routes must visit at least one "route junction point" as intermediate stop(s).

At the "junction points," a city may belong to more than one subsegment, segment or route. To illustrate this, the following examples are given (refer to Fig. 3-1-3):

- LAX, being a member of all 3 sets -- segment {B}, subsegments {A-a} and {A-b}, can fly to and from any city in the system non-stop.
- OAK and SFO/SJC, being members of both segment {B} and subsegment {A-b}, can fly to and from any city in {B} and {A-b}.
- LAS, being a member of both segment {B} and {A-b}, can fly to and from any city in {B} and {A-b}.

Notice that since OAK and SFO/SJC are both in {B} and {A-b}, any junction point(s) between {B} and {A-a} or {A-b} and {A-a} can serve as intermediate stop(s) between {OAK, SFO/SJC} and any city in {A-a}. Similarly, since LAS is both in {A-a} and {A-b}, any junction point(s) between {A-a} and {B} or {A-b} and {B} can serve as intermediate stop(s) between LAS and any city in {B}.

The relationships in the above examples are conveniently summarized in the "Venn Diagram" of Fig. 3-1-4.

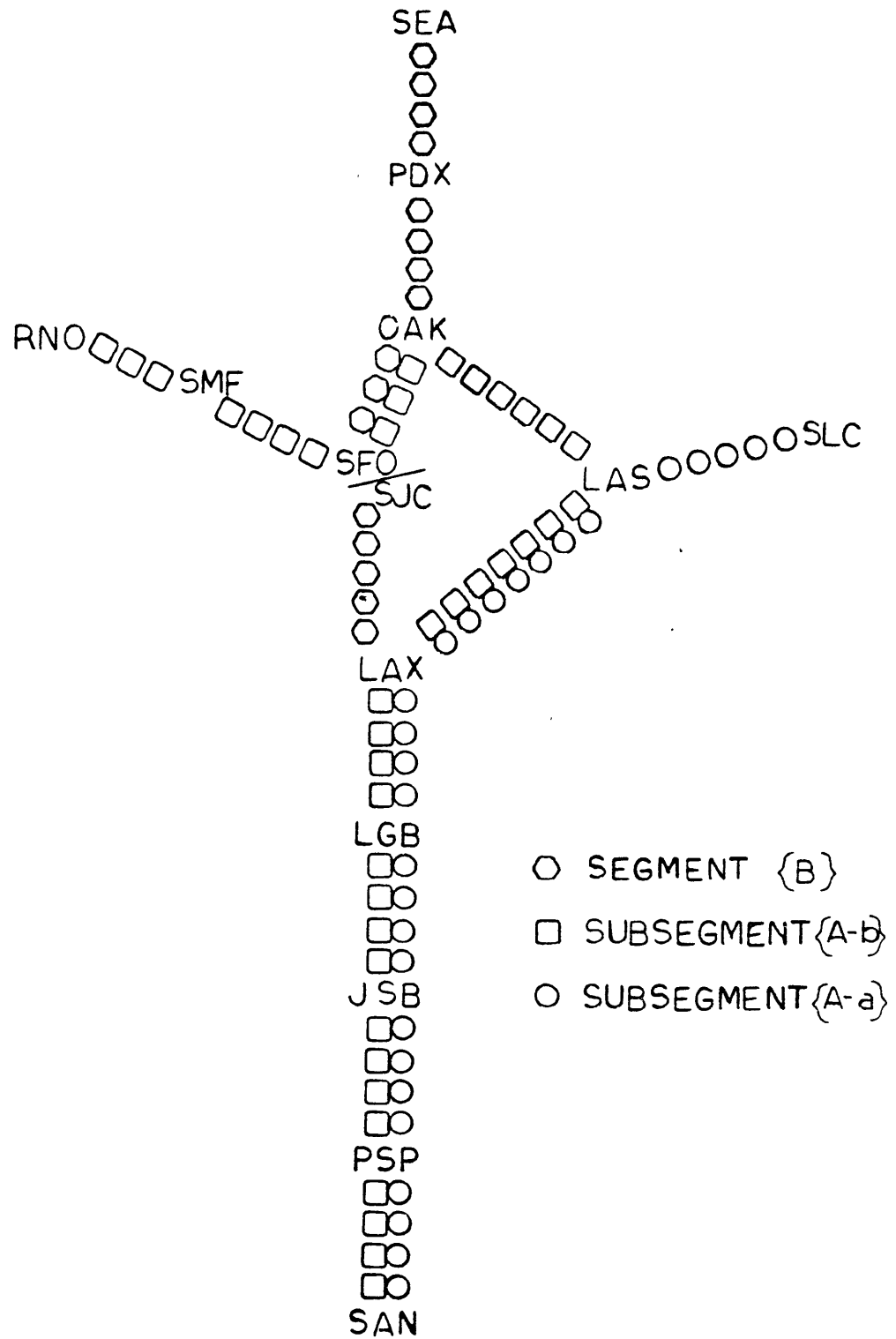


FIG. 3-1-3 CITIES IN THE ROUTE CERTIFICATE

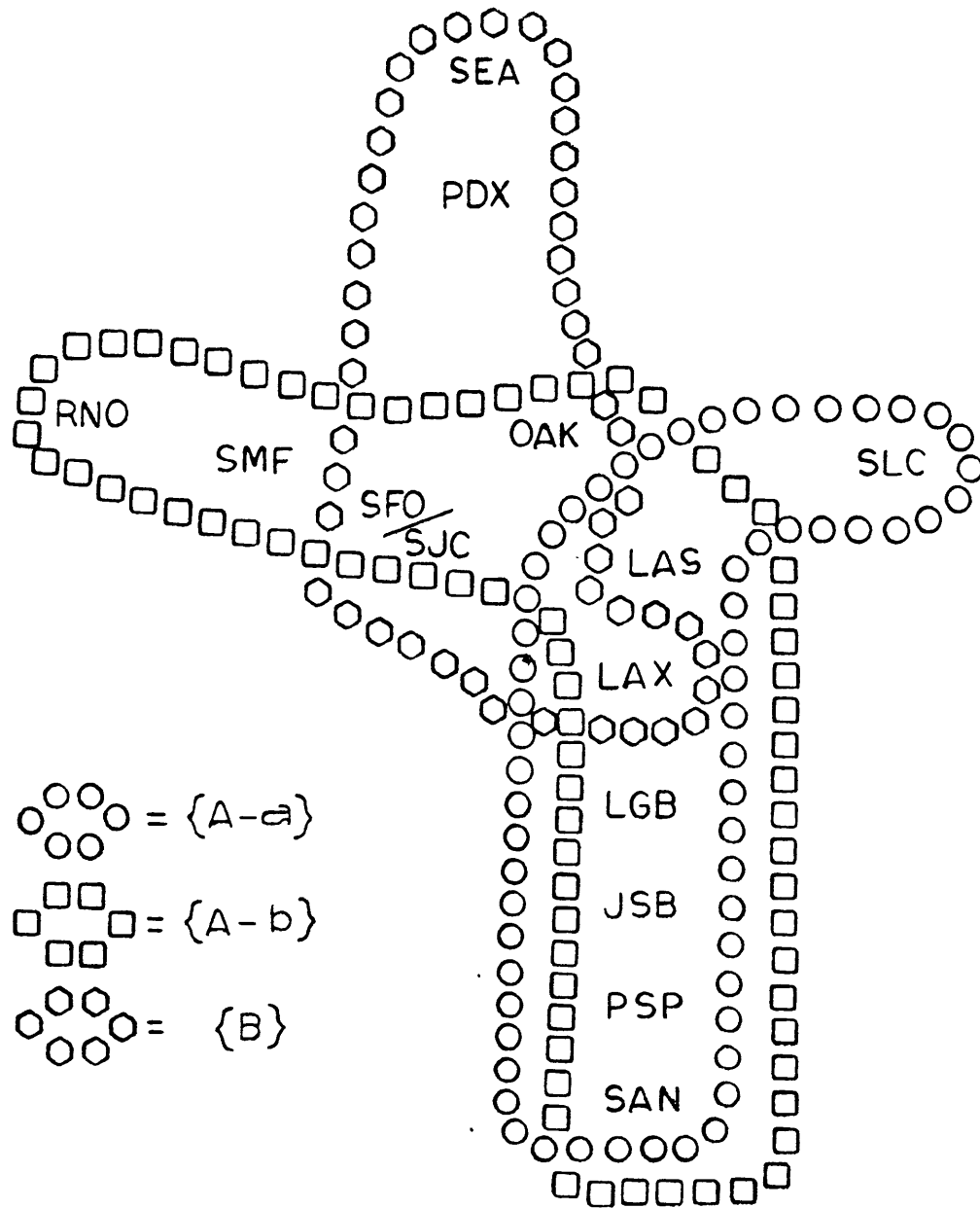


FIG. 3-1-4 VENN DIAGRAM REPRESENTATION

special qualifications to general guidelines

The above systematic description of route authority is unfortunately complicated by quite a few "exceptions" to the general rule:

"NO SINGLE PLANE" SERVICE

Example:

Clause (4) of the Western Airlines Route Certificate specifies that no single plane service is allowed between LAS and RNO. According to the general specification, LAS and RNO both belong to a subsegment set (Figure 3-1-4) and one would expect that non-stop service is allowed according to the general terms. Clause (4), however, qualifies that LAS and RNO may not be served by single plane non-stop or multi-stop flights.

"MUST ORIGINATE OR TERMINATE" SERVICE

Example:

Whenever American Airlines serves SAN on its route 4, the flight must originate or terminate at a set of cities east of PHX (PHX inclusive). From the general rule alone, one would have guessed that a non-stop flight can be scheduled from any city to SAN, turn around, and then return to the origin city (or the other way around).

"No turn around" service restriction is a special case of the above where no origin or termination cities are specified.

3.1.3 A Graph-Theoretic Representation of Route Authority

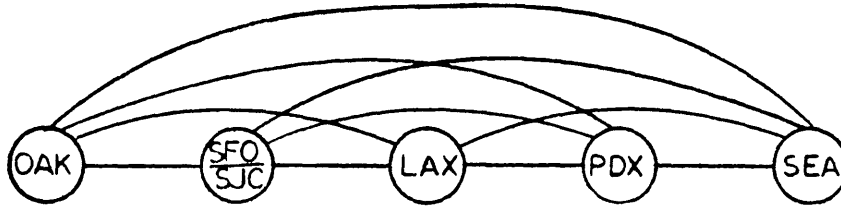
With the background interpretation of the route authority from the last section, we present here a graph-theoretic representation of the route authority. The graph can be described by the contiguity matrix

described in Section 3.1.1. Legal non-stop, one-stop ... routes can then be generated by raising the contiguity matrix to its first, second, ... powers.

representation of the general guidelines

In a subsegment, every city i (j) can reach another city j (i) in the same subsegment by a non-stop flight. Representing an aircraft hop by an edge (i.e. non-directional arcs), the number of edges incident on a city vertex is $(n-1)$, where n is the number of vertices in the subsegment. A graph-theorist says that every city pair in a subsegment is "strongly connected", and that every vertex in the subsegment is of "degree" $(n-1)$. The subsegment graph for subsegment {B} is shown in Figure 3-1-5, where we also show the contiguity matrix representing the graph. For a subsegment without special qualifications (such as those given in Section 3.1.2) the matrix is solidly filled. The matrix shown says that non-stop routes can be scheduled between all city pairs. Multi-stop routes can be built by stringing together these basic non-stop segments, in the manner explained in Section 3.1.1.

Two subsegment sets of cities intersect at junction points. Connections between cities in two different subsegment sets are restricted to make intermediate stop(s) at the junction points. If the intersection set contains only one city, that single city is an "articulation point" in graph-theoretic terms. Without the articulation point, the two subsegment sets of cities will be disconnected. If there are two or more junction points, the two groups of cities are said to "biconnect". In the example of Figure 3-1-6, the two subsegment



	1	2	3	4	5
	OAK	SFO SJC	LAX*	PDX	SEA
1 OAK	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
2 SFO SJC	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
3 LAX	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
4 PDX	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
5 SEA	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

FIG. 3-1-5
SUBSEGMENT GRAPH AND ITS CONTIGUITY MATRIX

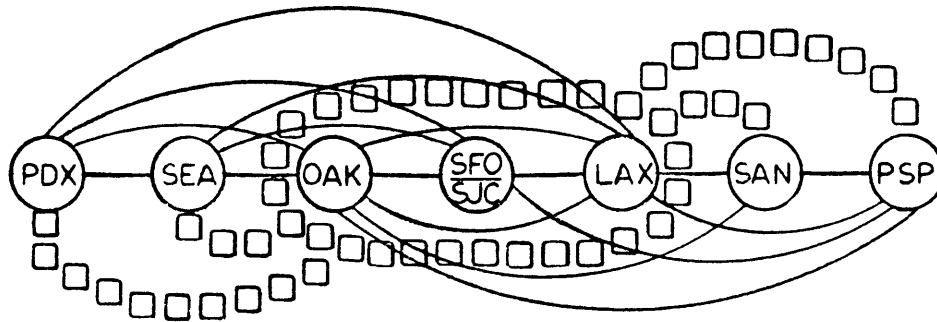
sets $\{A-b\}$ and $\{B\}$ have three junction points, OAK, SFO/SJC and LAX, the two sets are therefore biconnected, in the sense that connection between cities in $\{A-b\}$ and $\{B\}$ can be made at OAK, or SFO/SJC or LAX or any combination of those three cities. The contiguity matrix has a submatrix of "0"'s in the cells corresponding to the restricted city pairs. Notice that two subsegment sets are "connected" (in fact "strongly connected", since route authority specifications are non-directional) if the intersection set of junction points is non-empty.

representation of special qualifications

"NO SINGLE PLANE" SERVICE

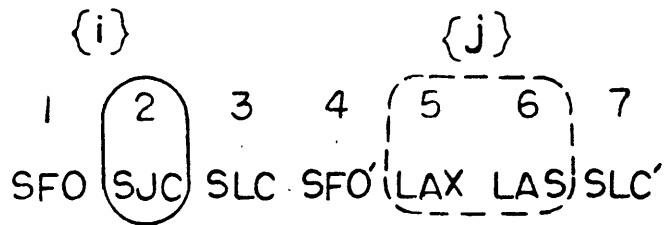
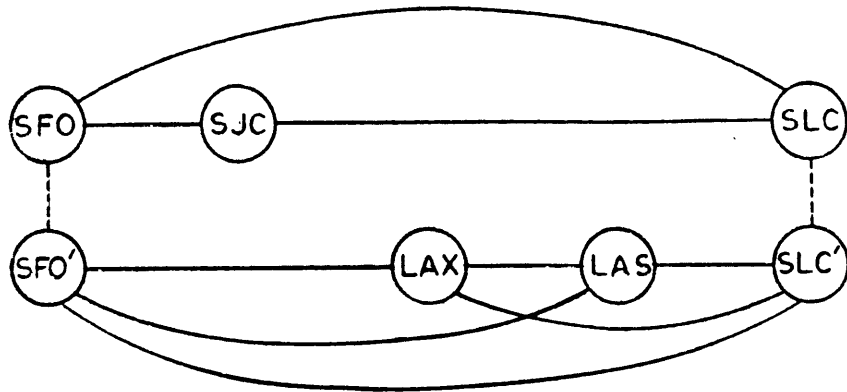
"No single plane" services can be readily handled by "splitting" a subsegment into two more basic building blocks. If no single plane service is allowed between SJC and LAX or LAS, splitting the subsegment in the way of Figure 3-1-7 can be shown to handle this route restriction nicely. In Figure 3-1-7 we have separated the two sets of restricted city pairs $\{i\}-\{j\}$ into two "split" subsegments $\{i\}$ and $\{j\}$, with the non-restricted cities in the original subsegment duplicated in both split subsegments. Notice the submatrix of zero's in the contiguity matrix clearly shows that these two split subsegments are disconnected from each other (Figure 3-1-7).

As an example, the "no single plane service" restriction for LAS-RNO (Clause (4) of the Certificate) and between SJC and SEA, PDX, LAS, LAX, LGB or SAN (Clause (6) of the Certificate) is represented in Figure 3-1-8.



	1	2	3	4	5	6	7
	OAK	SFO SJC	LAX	PDX	SEA	SAN	PSP
1 OAK	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)
2 SFO SJC	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)	(2,7)
3 LAX	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)	(3,7)
4 PDX	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	0	0
5 SEA	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	0	0
6 SAN	(6,1)	(6,2)	(6,3)	0	0	(6,6)	(6,7)
7 PSP	(7,1)	(7,2)	(7,3)	0	0	(7,6)	(7,7)

FIG. 3-1-6 THE INTERSECTION OF TWO SUBSEGMENTS AND THE CORRESPONDING CONTIGUITY MATRIX



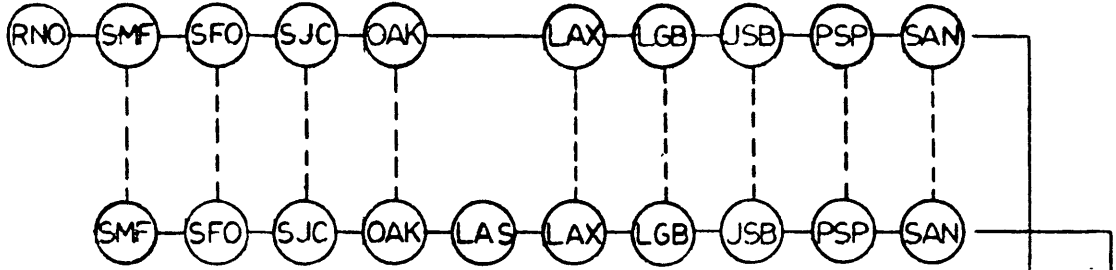
	$\{i\}$	1 SFO	(1,1)	(1,2)	(1,3)	0	0	0	0
		2 SJC	(2,1)	(2,2)	(2,3)	0	0	0	0
		3 SLC	(3,1)	(3,2)	(3,3)	0	0	0	0
		4 SFO'	0	0	0	(4,4)	(4,5)	(4,6)	(4,7)
	$\{j\}$	5 LAX	0	0	0	(5,4)	(5,5)	(5,6)	(5,7)
		6 LAS	0	0	0	(6,4)	(6,5)	(6,6)	(6,7)
		7 SLC'	0	0	0	(7,4)	(7,5)	(7,6)	(7,7)

FIG. 3-1-7 SPLITTING A SUBSEGMENT INTO TWO

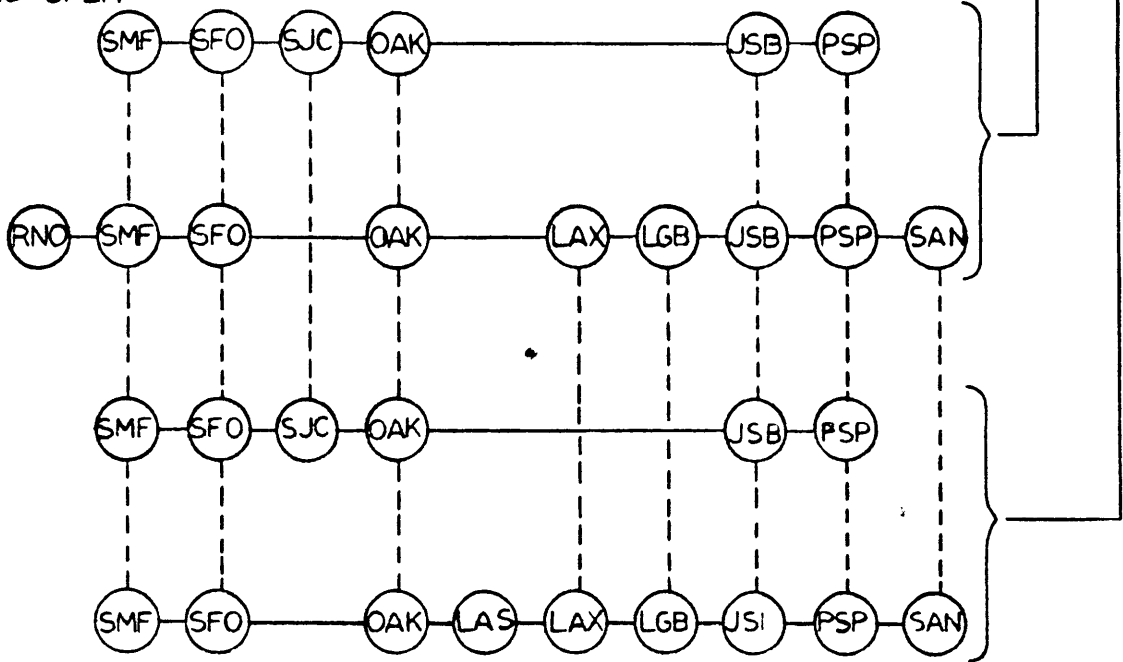
SUBSEGMENT {A-b}

140

1st SPLIT:



2nd SPLIT:



SEGMENT {B}

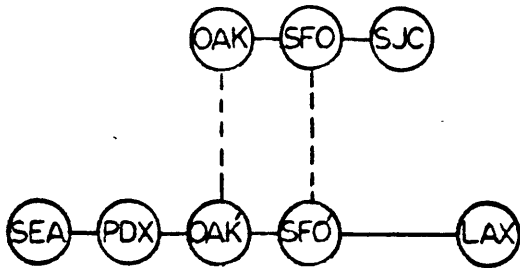


FIG. 3-1-8 SPLIT SUBSEGMENTS AND SEGMENTS

In general, for each $\{i\}$ - $\{j\}$ single plane restriction, the subsegment is split into two sets, one containing $\{i\}$ and the cities of the subsegment not in $\{i\}$ (call them $\{k\}$), and the other containing $\{j\}$ and $\{k\}$. Notice that by duplicating $\{k\}$ two times in the two disconnected split subsegments we have increased the number of "dummy cities" in our network and adjacency matrix by $\{k\}$, which may be a large number. A modified procedure is substituted for this approach in the software package RISE-I for computational efficiency.

"MUST ORIGINATE OR TERMINATE" SERVICE

"Must originate or terminate" services can be handled in the graph-theoretic manner only in an indirect and rather inconvenient manner. Let us say that any flight serving SAN must originate or terminate at SLC on subsegment $\{A-a\}$. From circuitry or other reasons, we identify the few cities that may logically be qualified as intermediate stops between SAN and SLC as LAS, PSP or both, PSP-LAS. In Figure 3-1-9, aggregate nodes X, Y, Z are defined. Viewing SAN-X, SAN'-Y, or SAN''-Z as links, it is rather obvious that all the flights serving SAN would originate or terminate at SLC.

The drawback about this method is the dimensionality. If there are m potential intermediate stops, the number of dummy aggregate nodes would be $m!$ If the Route Certificate says "must originate or terminate at SLC or east of SLC," then depending on the number of cities east of SLC being considered, say n of them (including SLC), the number of dummy aggregate nodes would be $n.m!$ Such a procedure is clearly not practical. A modified scheme will be adopted in the software package

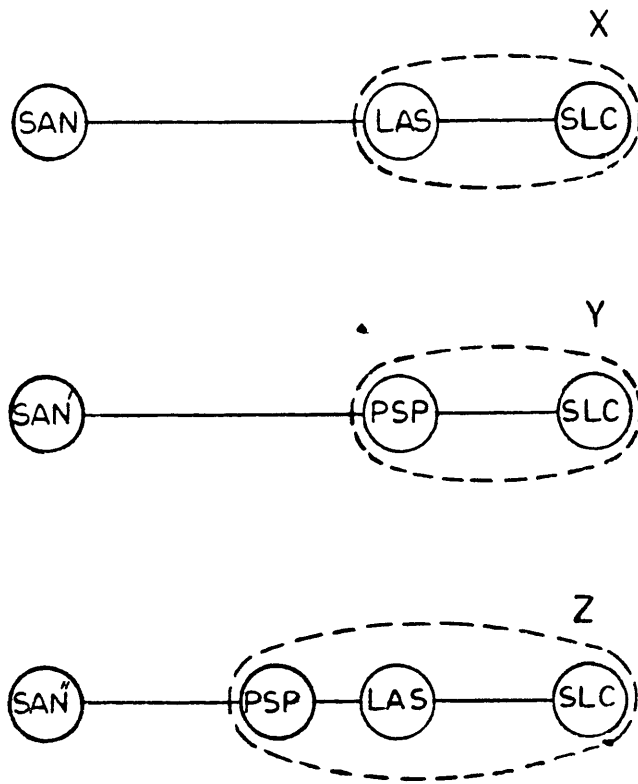


FIG. 3-1-9 "MUST ORIGINATE OR TERMINATE" RESTRICTION

RISE-I for computational efficiency.

Notice, the "no turn around" restriction is a special case of the above, in which SLC is removed from the aggregate nodes X, Y and Z.

- - -

To summarize the graph-theoretic way of representing route authority, we have coded the C.A.B. Route Certificate of Western Air Lines in a contiguity matrix (Figure 3-1-10). The blocks of filled entries represent subsegments. The blocks of "0"'s represent interchange between subsegments (segments, routes) or split subsegments. For the sake of clarity, "no single plane" service restrictions are not shown, except for that applied on segment B.

3.1.4 Generating Route Candidates via Graph-Theoretic Techniques

The readers have seen how a Route Certificate can be quantified graph-theoretically into a contiguity matrix. The current section will demonstrate exactly how route candidates and the associated passenger routings can be generated via raising the power of the matrix.

Figure 3-1-11 shows two basic subsegments {CHI-DAY-CMH-NYC} and {DAY-CMH-WAS} intersecting at the junctions points DAY and CMH. The contiguity matrix corresponding to this hypothetical route authority is shown below. We have assumed that there are no special qualification terms in the Route Certificate.

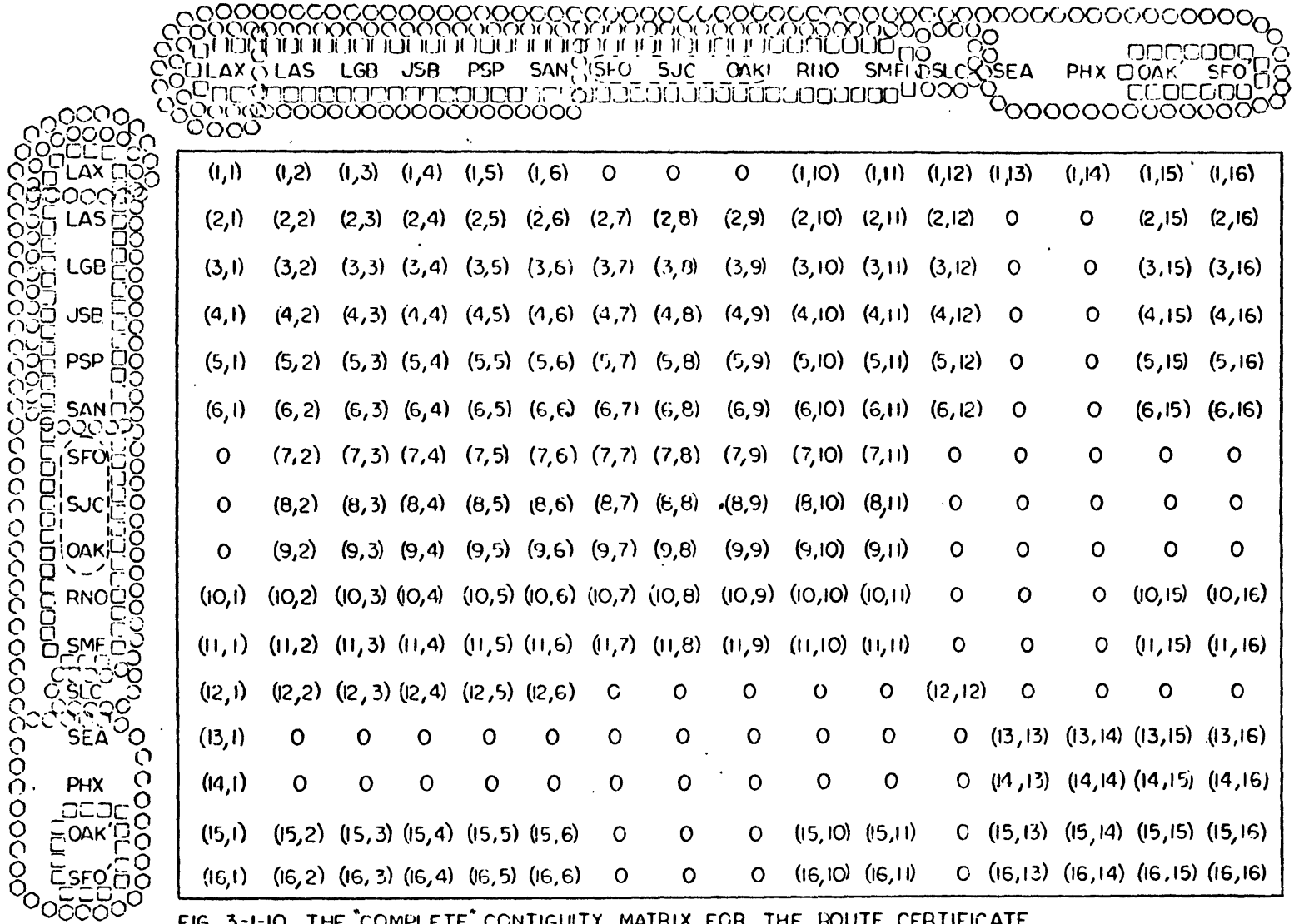


FIG. 3-1-10 THE "COMPLETE" CONTIGUITY MATRIX FOR THE ROUTE CERTIFICATE

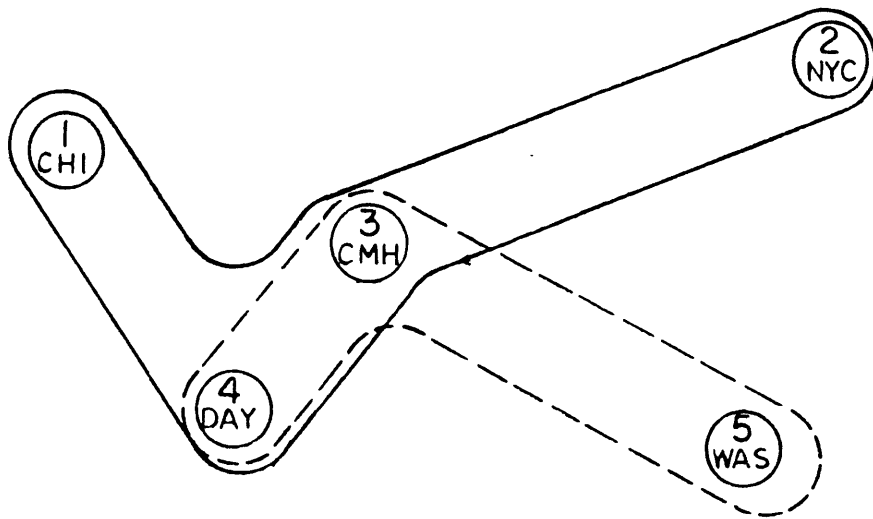


FIG. 3-1-11 ROUTE CERTIFICATE CONSISTING
OF 2 SUBSEGMENTS

$$\begin{array}{rcc}
 & & \begin{array}{ccccc}
 1 & 2 & 3 & 4 & 5 \\
 \text{CHI} & \text{NYC} & \text{CMH} & \text{DAY} & \text{WAS}
 \end{array} \\
 \begin{array}{r}
 1 \text{ CHI} \\
 2 \text{ NYC} \\
 3 \text{ CMH} \\
 4 \text{ DAY} \\
 5 \text{ WAS}
 \end{array} & = & \begin{array}{c}
 \left[\begin{array}{ccccc}
 (1,1) & (1,2) & (1,3) & (1,4) & 0 \\
 (2,1) & (2,2) & (2,3) & (2,4) & 0 \\
 (3,1) & (3,2) & (3,3) & (3,4) & (3,5) \\
 (4,1) & (4,2) & (4,3) & (4,4) & (4,5) \\
 0 & 0 & (5,3) & (5,4) & (5,5)
 \end{array} \right]
 \end{array}
 \end{array}$$

The above A matrix consists of two parts, the upper left-hand submatrix corresponds to subsegment {CHI-NYC-CMH-DAY} while the lower right-hand submatrix corresponds to subsegment {CMH-DAY-WAS}. The junction points CMH and DAY provide the connection between the cities in the two subsegments.

In its present form, A denotes the non-stop routes possible between all the city pairs in the system. It can be checked that city pairs within the same subsegment can be served by non-stops while city pairs in different subsegments, such as CHI-WAS, cannot.

Raising A to its second power generates all the one-stop routes. For the sake of clarity, we will show only the generated routes between CHI and WAS:

$$A^2 = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \text{CHI} \\ \text{NYC} \\ \text{CMH} \\ \text{DAY} \\ \text{WAS} \end{array} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ \text{CHI} & \text{NYC} & \text{CMH} & \text{DAY} & \text{WAS} \end{array} \begin{array}{c} (1,3) \cdot (3,5) + (1,4) \cdot (4,5) \\ - \\ - \\ - \\ - \end{array}$$

The two one-stop routes possible between CHI and WAS are therefore 1→3→5 or 1→4→5, corresponding to CHI→CMH→WAS and CHI→DAY→WAS respectively. Thus, cities in subsegment {CMH-DAY-WAS} can be reached from cities in subsegment {CHI-NYC-CMH-DAY} via either junction point city CMH or DAY.

From a combinational viewpoint, the number of legal routes that are possible between a terminal city pair is huge. For practical applications, usually the shortest route is of interest.

the shortest route

The contiguity matrix way of route generation actually provides more information than just the legal aircraft routes. Shortest route computations can readily be performed in the algebraic framework for each terminal pair. The minimum time route can be readily picked out among a number of legal candidates that may serve the terminal pair. Let us illustrate with an example.

Suppose we are given the following inter-city non-stop distances in clock minutes for the cities in the two subsegments:

	1	2	3	4	5
	CHI	NYC	CMH	DAY	WAS
1 CHI	0	108	57	51	0
2 NYC	108	0	79	87	0
3 CMH	57	79	0	31	60
4 DAY	51	87	31	0	68
5 WAS	0	0	60	68	0

We will define the following matrix operation:

$$T^2 = [t_{ij}^1] = \underset{k=1}{\overset{N}{X}} (t_{ik}^0 * t_{kj}^0) \quad (3:1:1)$$

$$i, j = 1 \dots N$$

where $\underset{k=1}{\overset{m}{X}} (x_k)$ is $\min (x_1, x_2, \dots, x_m)$.

The operator $a*b$ equals infinity if either a or b (or both) is (are) zero, or the scalar sum of a and b otherwise:

$$a*b = \begin{cases} \infty & \text{if } a \text{ or } b \text{ (or both) is (are) zero} \\ a+b & \text{otherwise} \end{cases} \quad (3:1:2)$$

To illustrate (3:1:1), let us compute t_{15}^1 , which is the minimum time one-stop distance between CHI (city 1) and WAS (city 5):

$$\begin{aligned}
t_{15}^1 &= \sum_{k=1}^5 (t_{1k}^0 * t_{k5}^0) \\
&= \min (t_{11}^0 * t_{15}^0, t_{12}^0 * t_{25}^0, \dots, t_{15}^0 * t_{55}^0) \\
&= \min (0 * 0, 108 * 0, 57 * 60, 51 * 68, 0 * 0) \\
&= \min (\infty, \infty, 117, 119, \infty) \\
&= 117
\end{aligned}$$

The minimum distance between CHI and WAS is therefore 117 minutes of block time. This corresponds to the routing CHI-CMH-WAS.

Minimum two-stop routes can likewise be computed:

$$\begin{aligned}
T^3 = [t_{ij}^2] &= \sum_{k=1}^N (t_{ik}^1 * t_{kj}^0) \\
& \qquad \qquad \qquad (3:1:3) \\
& \qquad \qquad \qquad i, j = 1 \dots N
\end{aligned}$$

In general, we write

$$\begin{aligned}
T^{n+1} = [t_{ij}^n] &= \sum_{k=1}^N (t_{ik}^{n-1} * t_{kj}^0) \\
& \qquad \qquad \qquad (3:1:4) \\
& \qquad \qquad \qquad i, j = 1 \dots N \\
& \qquad \qquad \qquad n = 1, \dots, M
\end{aligned}$$

where M-stop is the longest route to be considered. The (3:1:4) operation is actually a dynamic programming algorithm.

It has been shown, therefore, that the contiguity matrix representation of route authority cannot only generate all the legal multi-stop routes, but also yield the minimum M-stop time distance between terminal points. The actual route (i.e. in terms of the cities that made up the route, e.g. A-B-C-D) that yields the minimum time path between the terminal points (in this case between A and D) is obtained via sheer bookkeeping.

passenger routings associated with an aircraft route

An aircraft route such as A-B-C-D provides through passenger services between all the city pairs in the route, viz. between A-D, A-C, B-D, A-B, B-C and C-D. In other words, through passenger routings are available for all the paired combinations of city pairs in the aircraft route. For an M-stop aircraft route, there will be $\binom{m+1}{2}$ different passenger routings--1 of them is m-stop, 2 of them are (m-1)-stop, 3 of them are (m-2)-stop, ... , and (m+1) of them is 0-stop.

Besides through routings, there are connect routings. A connect passenger routing between an O-D pair is available when two or more aircraft routes provide a continuous path from O to D. For example, non-stops A-B and B-C would provide a one-stop connect routing from A to C: A-B-C, making a transfer at station B. Similarly, a non-stop A-B and a one-stop C-B-D would facilitate a one-stop connect routing between A and D via A-B-D, connecting at B.

There would be quite a number of through/connect routings between a city pair from a combinational viewpoint, but usually only the shortest m-stop routings are of practical interest. The matrix-algebraic

framework readily lends itself to shortest m-stop through/connect routing computation. The algebra is exactly the same as that of (3:1:4).

3.2 Route Selection: An Integer Programming Approach

The previous Section 3.1 discusses how we generate all the topologically feasible route candidates. It has identified the "feasible region" of our constrained optimization model R.I.S.E. This section will be devoted to the technique of searching among the alternative route candidates in the feasible region, resulting in a selection of the best subset of routes. The route selection problem will be formulated as an integer program.

3.2.1 Selecting Among Route Candidates

Strictly speaking, R.I.S.E. is an optimization model that cannot be formulated in an explicit way in terms of objective function and constraints. In our formulation, R.I.S.E. sequentially generates columns (which are the routes) of the tableau, and simultaneously select the 'best' subset of routes. As route candidates are generated, not only more columns are annexed to the tableau, but also rows. In this sense, we are not working with a static tableau defined before execution time (as is the case of the usual linear programming problems). The row and column dimensions of the tableau keep growing as the algorithm proceeds. In order to show the nature of our optimization problem for illustration purposes, (e.g. to show the general structure of the constraints and objective function), we have to assume that all routes that comply with the C.A.B. route authority have been generated (hence the number of rows and columns fixed). In practice, the number of legal

routes that could be generated is astronomical. R.I.S.E. cannot be viewed as a two part process, with generation being separate from selection, in the sense that the former defines the feasible region and the latter optimizes over the feasible region. Computationally, we cannot combinatorily generate all legal routes and include every route in the rows and columns of the tableau. This would mean literally thousands and thousands of rows and columns. Rather, we would have to decide sequentially which column to generate and include in the tableau (and the associated row). In other words, we generate and select simultaneously. We will come back to this point in Section 3.3.

For obvious dimensionality reasons, we show here, for illustration purposes, an example tableau describing only a three-city system where all legal routes (see the route map in Figure 3-2-1) have been included into the rows and columns. As represented in this example integer program tableau, R.I.S.E. is reduced to a pure selection problem. We have to keep in mind that R.I.S.E. is both a generation and selection problem in the general case.

A brief description of our three-city example is necessary (refer to Figure 3-2-1). There are only four routes in the route map--three non-stop and one one-stop. Notice city pair 1-3 is served both by a non-stop route and a segment of a one-stop route.

3.2.2 Constraints of the Integer Program

Figure 3-2-2 shows that the tableau we are dealing with is a very sparse tableau with a 'staircase' structure. The 'subproblem' blocks are only "loosely tied together". The solution strategy, as will

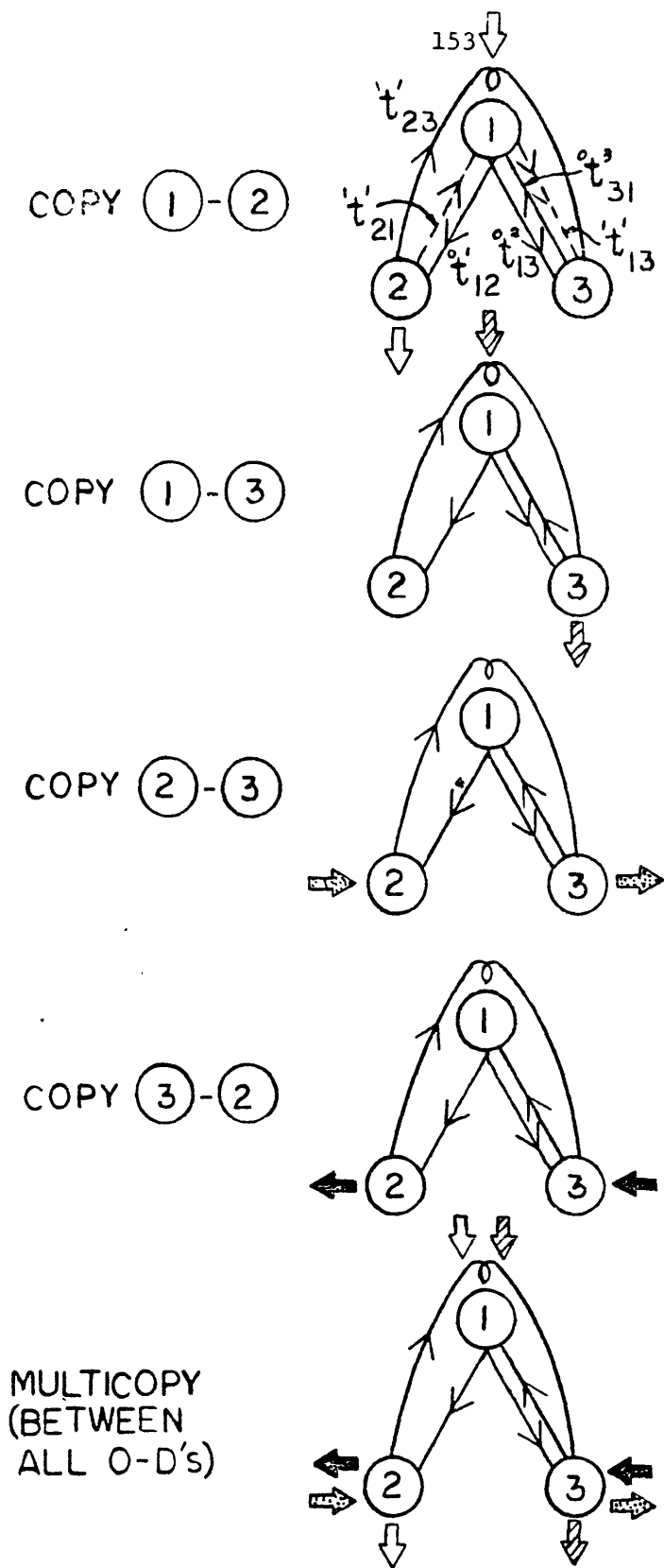
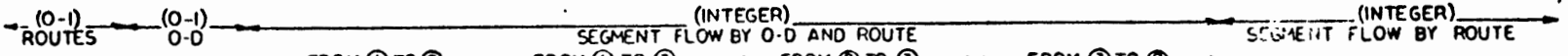


FIG. 3-2-1 EXAMPLE ROUTE NETWORK



$$\max \left\{ y_1, y_2, y_3, y_4, x_{12}, x_{13}, x_{23}, x_{32} \left[\min(\dots) + \min(\dots) + \min(\dots) + \min(\dots) - c_1 \max(\dots) \right] \right\}$$

		COMPETITION REQUIREMENT													
		NON-STOP ROUTES						I-STOP ROUTES							
SERVICE OF O-D FAX DEPENDS ON ACCEPTANCE OF CERTAIN ROUTES		1	2	3	1	2	3	1	2	3	1	2	3		
	X ₁₂	1													20
	X ₁₃		1												20
	X ₂₃			1											20
	X ₃₂				1										20
	FROM ③ TO ②		1												0
			2												0
			3												0
			1												0
			2												0
			3												0
	FROM ① TO ③		1											0	
2													0		
3													0		
FROM ① TO ②		1											0		
		2											0		
		3											0		
FROM ② TO ③		1											0		
		2											0		
		3											0		
FROM ③ TO ①		1											0		
		2											0		
		3											0		
FROM ② TO ①		1											0		
		2											0		
		3											0		
FROM ③ TO ②		1											0		
		2											0		
		3											0		
FROM ③ TO ①		1											0		
		2											0		
		3											0		
FROM ③ TO ③		1											0		
		2											0		
		3											0		
FROM ② TO ②		1											0		
		2											0		
		3											0		
FROM ① TO ①		1											0		
		2											0		
		3											0		

FIG. 3-2-2 INTEGER PROGRAMMING TABLEAU

be outlined in the next chapter, resort to a decomposition algorithm developed by the author.

A word about mathematical notations before we start writing equations: most symbols are defined in the way described in the report by Simpson [1969], except in a few cases where we have to define new notations for new variables not in his vocabulary. The writer prefers to follow a superscript and subscript notation. Instead of writing $P_{a pq r ij}$ to denote the passenger flow on segment ij of route r going from origin p to destination q on board aircraft type a , we write $y_{a ij}^{pq r}$.

competition requirement constraint

The first 'subproblem' block is represented by a "route-covering" matrix* triangular in appearance (see Figure 3-2-2). The rows are city pairs pq while the columns are route variables y_r^m . Each route is identified by the superscript m which denotes the number of intermediate stops in the route, and the subscript r , denoting the route number. For example y_2^0 means the second non-stop route while y_1^1 means the first one-stop route. y_r^m assumes a (0-1) value corresponding to whether the route is included in or rejected from the route network.

*This term is coined by the writer for convenience, since he is not aware of any official name given to this kind of matrix.

Notice that the city pairs are repeated in two 'copies' along the rows. In the "non-stop copy", all the city pairs are covered by non-stop routings only. Notice that city pair 1-3 is covered by both a non-stop route y_2^0 and a non-stop segment routing of a one-stop route y_1^1 . Notice also that city pair 2-3 is not covered by any non-stop routings, since the row entries corresponding to city pair 2-3 in the "non-stop copy" of the route covering tableau are all zeros. City pair 2-3 is covered only by a one-stop route y_1^1 —a "1" is found corresponding to the row 2-3 in the "one-stop copy" of the route covering tableau.

An m-stop route covers a number of city pairs via various routings. These routings are n-stop or less. For example, a one-stop route 2-1-3 covers O-D pairs 2-1, 1-3 and 2-3. The first two O-D pair is covered by a one-stop routing. Let ${}^n R_{pq}^m$ denote the set of m-stop routes covering city pair pq by n-stop routings. In the example just quoted, city pair 1-3 is covered by non-stop route y_2^0 via a non-stop routing (obviously), and we write ${}^0 R_{13}^0 = \{y_2^0\}$. The same city pair is covered by one-stop route y_1^1 via non-stop routing since 1-3 is a segment of y_1^1 and we write ${}^0 R_{13}^1 = \{y_1^1\}$. City pair 2-3 is only covered by a one-stop route y_1^1 via one-stop routing, we write ${}^1 R_{23}^1 = \{y_1^1\}$ (with the empty sets ${}^0 R_{23}^0 = \emptyset$ and ${}^0 R_{23}^1 = \emptyset$). Notice ${}^0 R_{pq}^0$ are always single element sets since there is only one non-stop route per city pair. ${}^0 R_{pq}^1$, however, can be multi-element since there may be more than one one-stop route with a common segment covering pq.

An important consideration in determining the route network is competition between airlines. If a competitive carrier is serving a

city pair non-stop, there is very little choice but to match up with a non-stop, if the airline is serious about the share of that city pair market. This kind of competitive pressure is recognized in our formulation. Let us say that city pair 1-3 is such a city pair market in our example. There are three alternatives open to the operator to cover 1-3 via a non-stop routing (refer to Figures 3-2-1 and 3-2-2):

- (i) He can cover city pair 1-3 by a non-stop route

$$y_2^0 \geq 1$$

- (ii) He can cover city pair 1-3 by a segment of the one-stop route

$$y_1^1 \geq 1$$

Note that each of the above two constraints can be written as

$$\sum_{r \in {}^0R_{pq}^m} y_r^m \geq 1 \quad \begin{array}{l} \text{for a city pair } pq \\ \text{and } m\text{-stop routes} \end{array}$$

where ${}^0R_{pq}^m$ are trivially* single element sets:

$${}^0R_{13}^0 = \{y_2^0\} \text{ and } {}^0R_{13}^1 = \{y_1^1\}.$$

*Our three-city example is so simple that the equations in (i) and (ii) appear trivial. Hopefully the oversimplification does not prevent the reader to see the general case where ${}^nR_{pq}^m$ is a multi-element set. We have considered using a five-city example for illustration. But that would "blow up" the size of our tableau in Figure 3-2-2.

(iii) In our case, 1-3 can be covered by either y_2^0 , or y_1^1 , or y_2^0 and y_1^1 :

$$y_2^0 + y_1^1 \geq 1 \quad \text{for city pair 1-3} \\ \text{and non-stop routing}$$

More formally,

$$\sum_{r \in O_{R_{13}}^1} y_r^1 + \sum_{r \in O_{R_{13}}^0} y_r^0 \geq 1 \quad \text{for city pair 1-3} \\ \text{and non-stop routing}$$

which is the same as

$$\sum_{m=0}^1 \sum_{\substack{r \in O_{R_{p_0 q_0}}^m \\ p_0 q_0}} y_r^m \geq 1 \quad \text{for } p_0 - q_0 \text{ and } n=0$$

In general, we write

$$\sum_{m=n}^M \sum_{\substack{r \in O_{R_{pq}}^m \\ pq}} y_r^m \geq 1 \quad \forall p-q \text{ and } n \quad (3:2:1)$$

where M is the number of intermediate stops in the longest route generated.

Equation (3:2:1) helps to explain the triangular structure of the route covering tableau. Since (3:2:1) is written for all pq and n , all city pairs in the network are repeated in $|n|$ copies. The n^{th} copy denotes the city pairs covered by n -stop routings. Thus the first copy depicts how city pairs are covered by non-stop routings. The second copy depicts how city pairs are covered by one-stop routings, etc. Suppose we look at the 2^{nd} copy. By definition, non-stop routes,

r^0 , would not be able to cover city pairs via one-stop routings. The corresponding entries under the set of non-stop routes R^0 in the route covering tableau is therefore zero. For the same reasons, with the 3rd copy all entries under R^0 and R^1 would be empty and so on. This explains the triangular structure of the route covering tableau.

Let us count the number of rows in this constraint. The upper bound of this number is $N(N-1)(M+1)$, where N is the number of cities in the system and M , as defined before, is the number of intermediate stops in the longest route generated. The set of O-D pairs is repeated in $(M+1)$ copies row-wise. For an 80-city system and the longest routes being two-stop, the number of rows in this constraint alone could be up to 18,960.

connectivity constraint

A city pair in the network is either connected on the route map (via through or connecting service) or a city pair is disjoint. For a disjoint city pair, there is no service between them (either connecting or through service). R.I.S.E. decides which city pair market is worth servicing and if so, whether the city pair should be served by a non-stop, multi-stop or connecting service. When a city pair pq is connected by the route network x_{pq} assumes the value of "1". otherwise $x_{pq} = 0$. Similar to y_r^m , x_{pq} is a (0-1) binary variable.

The connectivity constraints relate the Origin-Destination (O-D) variables x_{pq} and the route variables y_r^m . An O-D pair is connected if routes exist to make the connection. Take the simple case of city pair 1-2, these cities would not be connected (i.e., x_{12} would not be 1)

unless non-stop route 1-2 exists (i.e., $y_1^0 = 1$). To put this dependency between x_{12} and y_1^0 in a formal language

$$y_1^0 - x_{12} \geq 0 \quad (3:2:2)$$

The above inequality says that x_{12} cannot be 1 unless y_1^0 is unity, for if x_{12} were 1 when $y_1^0 = 0$, the inequality would be violated.

Take another example. City pair 1-3 will be connected if either route y_2^0 or route y_1^1 exists. This either/or relation is conveniently represented below

$$y_2^0 + y_1^1 - x_{13} \geq 0 \quad (3:2:3)$$

It can be checked that x_{13} can be 1 if either y_2^0 or y_1^1 are "1".

Finally, take a city pair which is connected via a change-of-plane service--O-D pair 3-2. City pair 3-2 would not be connected unless both routes y_1^0 and y_3^0 exist. This both/and relation appears as

$$1/2 y_1^0 + 1/2 y_3^0 - x_{32} \geq 0 \quad (3:2:4)$$

It can be checked that x_{32} cannot be "1" unless both y_1^0 and y_3^0 are "1".

Let us formalize what we have said so far about connectivity between O-D pairs. Define R_{pq} as the set of routes that cover or connect city pair pq . Equation (3:2:2) can be rewritten as

$$\sum_{m,r \in R_{12}} y_r^m - x_{12} \geq 0 \quad \text{where } R_{12} = \{y_1^0\} \quad (3:2:5)$$

Equation (3:2:3) now looks like

$$\sum_{m, r \in R_{13}} y_r^m - x_{13} \geq 0 \quad \text{where } R_{13} = \{y_2^0, y_1^1\} \quad (3:2:6)$$

Equation (3:2:4) looks like

$$1/2 \sum_{m, r \in {}_1R_{32}} y_r^m - x_{32} \geq 0 \quad \text{where } {}_1R_{32} = \{y_1^0, y_3^0\} \quad (3:2:7)$$

where we have defined ${}_1R_{32}$ as the set of routes connecting 3-2 in exactly one change of plane. The following general constraint includes all the equations (3:2:5), (3:2:6) and (3:2:7).

$$\sum_{m, r \in R_{pq}} y_r^m - x_{pq} \geq 0 \quad \forall pq \quad \text{for through service} \quad (3:2:8)$$

$$\sum_{\ell} \frac{1}{|\ell^R_{pq}|} \sum_{m, r \in {}_\ell R_{pq}} y_r^m - x_{pq} \geq 0 \quad \forall pq \quad (3:2:9)$$

for connecting service

In general we define ${}_\ell R_{pq}$ as the set of routes serving pq in exactly ℓ change of plane. Notice that pq may be served by both a one-connection and two-connection routing (although the latter would probably be negligible).

In case there are more than one connect routing between a connecting city pair pq , additional constraints have to be written. Take the following example: if city pair A-B can be served by both a one-connection at W and a two-connection at Y and Z (i.e., the two

alternative routings from A to B will be ~~A-W-B~~ and ~~A-X-Y-B~~,

$$y_1^0 \ y_2^0 \qquad y_3^0 \ y_4^0 \ y_5^0$$

(3:2·9) would look like

$$1/2 (y_1^0 + y_2^0) + 1/3(y_3^0 + y_4^0 + y_5^0) - x_{AB} \geq 0$$

Or more compactly

$$1/2 \sum_{r, m \in {}_1R_{AB}} y_r^m + 1/3 \sum_{r, m \in {}_2R_{AB}} y_r^m - x_{AB} \geq 0$$

The readers can check that x_{AB} could be "1" if some routes in set ${}_1R_{AB}$ and ${}_2R_{AB}$ are "1"'s (say $y_1^0 = 1$ and $y_3^0 = y_4^0 = 1$, making up a sum of $1/2 \cdot 1 + 1/3 \cdot 2 = 7/6 > 1$). Physically it means a passenger can be connected from A to B by a routing of A-W-A-X-Y. This is obviously meaningless because a passenger in general cannot take some segments in the ${}_1R_{AB}$ routings and some segments in the ${}_2R_{AB}$ routings to go from A to B. He has either to take the routing specified in ${}_1R_{AB}$ or that in ${}_2R_{AB}$. To prevent this from happening, we impose the additional constraint which would make sure that all the routes in ${}_1R_{AB}$ are taken together or not at all,

$$y_1^0 - y_2^0 = 0$$

$$y_3^0 - y_4^0 = 0$$

$$y_4^0 - y_5^0 = 0$$

In general, these additional constraints look like

$$(y_r^m - y_{r'}^m) = 0 \quad \forall y_r^m, y_{r'}^m, \in {}_l R_{pq} \quad (3:2:10)$$

$\forall pq$ served by more than 1 connect routing

and $\forall l$

The above constraints (3:2:9) and (3:2:10) are typically written for a class of indivisible, dependent "projects" called "contingent" projects*. A "project" in our case is defined as the acceptance or rejection of a route y_r^m which is part of a connection routing** for a city pair. Project A is contingent on project B if project A is ineffective unless project B is implemented. Constraints (3:2:9) and (3:2:10) say that for connecting traffic, the potential revenue from the O-D passengers cannot be realized unless all of the routes in each of the various connect routings from p to q have been accepted.

DISCUSSION:

The set of routes R_{pq} for the case of a connecting service has to be known before (3:2:9) can be written. This points out the fact that

*These terms are coined by mathematical programming minded applied economists like Weingartner [1963]. An 'indivisible' project yields no return unless the entire project is adopted. Projects are 'dependent' if the effectiveness of a particular project depends on whether certain other projects have been implemented.

**For the purpose of our discussion here, we have defined routes to be the string of cities visited by an aircraft under the same flight number. Routings, on the other hand, are paths followed by a passenger in going from O to D. The routing may therefore consist of segments from the same route or different routes.

it is not sufficient just to generate all the reasonable aircraft routes between all city pairs (which is a rather large number, as pointed out before), it is also necessary to trace out all the passenger connect routings (which again may be large in number) between a city pair not "satisfactorily" served by through service. This includes those city pairs not connected by through service and those city pairs, although connected by multi-stop through service (say a two-stop), can be better served by a one-stop connection service, for example. Again, the "curse of dimensionality" haunts us. The writer again comes to the conclusion that route generation and selection should be done simultaneously to make R.I.S.E. computationally soluble. More will be said about this point when we discuss the decomposition approach for R.I.S.E. in Section 3.3.

Let us count the number of rows. Constraints (3:2:8) and (3:2:9) sum up to the upper bound of $N(N-1)$, where N , as defined before is the number of cities in the system. Assuming that one quarter of the city pairs in the network are served by connecting service and each connecting service is a one-stop and one-change of plane routing, constraint (3:2:10) would amount to $1/2N(N-1)$. We write one equation for each city pair. For an 80-city system, the total number of rows in the connectivity constraint could be up to 9,480.

passenger flow constraint

Let us examine the passenger traffic between a city pair. A city pair pq can be served by (i) a non-stop route $p-q$, (ii) a multi-stop route $p-\dots-q$ or (iii) a through/connect routing of a route/routes.

Take the case where the city pair is served by a non-stop routing which is a segment of a multi-stop route. For example, city B-C is served by the non-stop routing B-C of the multi-stop route A-B-C-D. The passengers in segment (B,C) could be from diverse origins and destinations--they may originate from B and terminate at C (i.e., local traffic), or they may be through traffic from A to C or B to D, and there may even be connecting traffic making connections at A or B. The passenger traffic in (B,C) can therefore be identified by (i) the route they are traveling in and (ii) the O-D of the passenger. We say that the p passengers between i and j is identified by both the route y_r^m and the O-D pair pq -- $\begin{matrix} m & r \\ p & q \end{matrix} P_{ij}$. The passenger flow on route y_r^m is said to be composed of "multi-commodity" flows (a term first coined by the management scientists when they modelled the distribution of multi-commodity goods between warehouses and shops). In our context, it is more appropriate to call them "multi-O-D" flows because they are passenger flows from diverse O-D's. When an individual passenger wants to make a trip from O to D, he has a large number of alternative paths open to him. Theoretically, he could take any combination of segments in the whole network as long as they constitute continuous paths from O to D. He can take a rather circuitous multi-stop and multi-connection route if he wants to "see the country". Or he may want to go via non-stop if he is the average business traveller who has a schedule to meet. In our example, to model this path choice phenomenon, the traffic between an O-D pair is given a whole copy of the route map to choose from. Thus the O-D traffic from 2 to 3 is

flown over copy 2 - 3 of the four copies of the route network (see Figure 3-2-1).

At this point, we model builders have to make an assumption on how an average passenger would make his path choice. The widely accepted assumption is that he would take the shortest time path from O to D. Each passenger would take the path most convenient for himself. Seldom could he be persuaded to take a circuitous, inconvenient, multi-stop routing between the city pair so that the airline could save money in putting in a redundant non-stop service between the same city pair (hence a higher profit for the airline). From the point of view of the passenger, he just wants to go the most convenient path from O to D. Such is the difference between "descriptive" versus "prescriptive" traffic flows.

Referring back to Figure 3-2-1, for O-D flow 2 - 3, we have a min-path problem on the copy of network from 2 to 3. In the figure is shown four copies of the network corresponding to four O-D passenger groups making their path choice on their copy of the network. Each group of O-D passengers trying to take the minimum time path from O to D. We are dealing with four min-path problems. In this objective function we will find four minimization operators. This will be discussed more fully when we come to describe the objective function.

Each copy of the network is represented by a node-arc incidence matrix. This matrix has nodes along the rows and arcs (in our case route segments) along the columns. If we read row-wise for a given node, say node 1 in our example, and put a "1" under the arc if it

is incident from it (i.e., pointing away from the node) and a "-1" if it is incident on it (i.e., pointing into the node). We have

$$o_{P_{12}}^1 + o_{P_{13}}^2 - o_{P_{31}}^3 - l_{P_{21}}^1 + l_{P_{13}}^1$$

in the tableau (see Figures 3-2-1 and 3-2-2). One characteristic of the node-arc incidence matrix is that each column has only two entries-- a "+1" and a "-1".

The route network is copied four times. Each passenger flow variable m_{ij}^r (the flow in an arc) is tagged with different O-D designations in each copy -- m_{pq}^r . If we are talking about copy 2 - 3, we can write for node 1, where no passengers originate or terminate, the constraint

$$o_{12}^1 + o_{13}^2 - o_{31}^3 - l_{21}^1 + l_{13}^1 = 0$$

which simply says "flow in equals flow out", which is best seen in the following form

$$o_{12}^1 + o_{13}^2 + l_{13}^1 = o_{31}^3 + l_{21}^1$$

Since for this copy describing trips from 2 to 3, no passengers originate or terminate at node 1. Node 1 is neither a "source" or "sink". It is a "bypass node". Let us generalize the node-arc incidence equation for a bypass node k

$$\sum_{m,r} \left(\sum_i m_{pq}^r - \sum_j m_{pq}^r \right) = 0 \quad \forall pq$$

The first term above represents the sum of the flow into node k , the second term represents the sum of the flows from k . The flows in and out must be equal for the bypass node k .

While 1 is a bypass node for copy 2 - 3, it is an origin node for copy 1 - 3 (i.e., a source). Passengers originate at 1 and terminate at 3 in this copy. In this case, the bundle of flow into node 1 would include the originating traffic P_{13} . We write the node-arc incidence information for node 1 as

$$P_{13} + \sum_{m,r} \sum_i \overset{\text{flow in}}{13^r i3} m_{pr} = \sum_{m,r} \sum_j \overset{\text{flow out}}{13^r 3j} m_{pr}$$

which is the same as

$$\sum_{m,r} \sum_i \overset{m_{pr}}{13^r i3} - \sum_{mr} \sum_j \overset{m_{pr}}{13^r 3j} = -P_{13}$$

In general for k being an origin (source) p :

$$\sum_{m,r} \sum_i \overset{m_{pr}}{pq^r ip} - \sum_{m,r} \sum_j \overset{m_{pr}}{pq^r pj} = -P_{pq} \quad \forall pq$$

It can be shown that if k is a sink q , the above equation is the same except for a sign,

$$\sum_{m,r} \sum_i \overset{m_{pr}}{pq^r iq} - \sum_{m,r} \sum_j \overset{m_{pr}}{pq^r qj} = P_{pq} \quad \forall pq$$

To summarize the node-arc incidence relation for an origin (source), destination (sink) and bypass node, we write

$$\sum_{m,r} \sum_i m_{pq}^{pr}{}_{ik} - \sum_{m,r} \sum_j m_{pq}^{pr}{}_{kj} = \begin{cases} -P_{pq} & \text{for } k = \text{origin } p \\ P_{pq} & = \text{destination } q \\ 0 & = \text{bypass node} \end{cases}$$

The copies of node-arc incidence matrices are linked to the connectivity constraint via the following relation

$$\sum_{m,r} \sum_i m_{pq}^{pr}{}_{ik} - \sum_{m,r} \sum_j m_{pq}^{pr}{}_{kj} = \begin{cases} -P_{pq} x_{pq} \\ P_{pq} x_{pq} \\ 0 \end{cases} \quad (3:2:11)$$

for $k = \text{origin } p$
 $= \text{destination } q$
 $= \text{bypass node}$

As an example of the above equation, the reader is referred to the tableau in Figure 3-2-2. Notice that connecting traffic is taken care of in exactly the same way as through traffic. This can be seen in the fourth copy of the node-arc incidence matrix (copy 3 - 2) where we can check (Figure 3-2-1) that the shortest route (in fact the only routing) between 3 and 2 is to go connect at city 1 . The ease with which connecting traffic can be handled within this formulation is rather satisfying.

A word about how we define the travel time on the links of the route network: segment travel time is defined as the block time. For a multi-stop route through service, the travel time for the whole route

is the sum of the block time of all the segments. In general we write $t_{ij}^{m,r}$ as the travel time of the m -stop routing from i to j on route r . Such definitions would be a rather rough average figure over a cycle time. Unfortunately at the level of aggregation we are dealing with, a better method has not yet been found.

With our multi-copy formulation and the way link travel time is defined, it is not necessary to assign 'weights' to a non-stop versus multi-stop versus connecting route arbitrarily. The present formulation will automatically find the shortest time path from all O's to all D's.

The right side of constraint (3:2:11) shows in a nutshell how R.I.S.E. defines the potentially profitable city pair market to serve. x_{pq} is either "1" or "0". If $x_{pq} = 1$, city pair pq is served and routings have to be found via the min-path formulation in copy $p - q$. Otherwise (i.e., $x_{pq} = 0$), the copy is not linked to the rest of the tableau and we have one less min-path problem (from p to q) to worry about.

All four copies of the node-arc incidence matrices are linked together by 'flow bundle' constraints. By a 'flow bundle' I mean that the passenger flow in a route segment is composed of a collection of flows from diverse O-D pairs, as explained earlier. Referring to Figure 3-2-1, the effect of the flow bundle constraint is shown graphically as 'superimposing' all the copies of route maps into one (see the graph at the bottom). In non-stop route y_1^0 , which leads from 1 to 2, we will find flows from 1 to 2 (the local traffic),

plus the connecting traffic going from 3 to 2 connecting at 1 .

We write that the flow in route y_1^0 , ${}^0P_{12}$, could be composed of flows from all O-D pairs,

$${}_{12}^0P_{12} + {}_{13}^0P_{12} + {}_{23}^0P_{12} + {}_{32}^0P_{12} = {}^0P_{12}$$

Some of the ${}_{pq}^0P_{12}$'s may be zero. In our case, an inspection shows that the O-D flows ${}_{13}^0P_{12}$ and ${}_{23}^0P_{12}$ will not be found in segment (1,2).

In general we write the following segment flow bundle equation

$$\sum_{pq} m_{pq}^r = m_{ij}^r \quad \forall m, r \text{ and } ij \quad (3:2:12)$$

The diagonal structure of the tableau generated by this constraint is shown in Figure 3-2-2.

It is necessary to define a segment flow variable m_{ij}^r in estimating the frequency of a route. Enough frequency has to be scheduled to accommodate the traffic on a segment which carries the largest bundle of passenger flow. Take the example of route A-B-D-D. If there are 100 passengers from A to B, 200 from A to C, and 50 from B to D and there is no connecting traffic, segment (A,B) carries 100 passenger, (B,C) carries 250 and (C,D) carries 50. Enough frequency has to be scheduled to accommodate the peak segment flow, which in our case is (B,D). Defining $\langle \cdot \rangle$ as the upper integer of \cdot , we can say in general

$$\text{frequency} \geq \max_{ij} \left\langle \sum_{pq} \frac{m_{pq}^r}{f_L \cdot s_a} \right\rangle$$

where f_L = load factor, and

s_a = seat capacity of aircraft type a

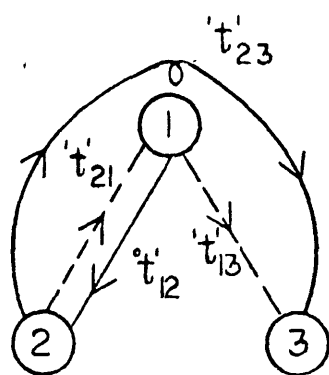
If we look at the objective function we find that route frequency is exactly determined in this manner.

DISCUSSION:

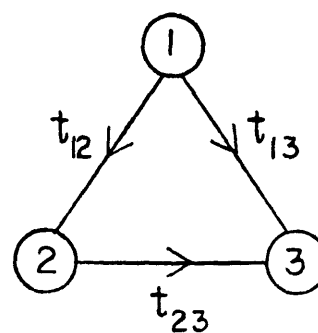
Multicopy formulation of traffic flow was originally suggested by Charnes and Cooper [1961], Robert and Funk [1964], Tomlin [1966], etc. We have extended the formulation to handle (i) 'descriptive' passenger flow by each O-D pair, (ii) an airline route network (instead of a highway network*) and (iii) a distinction between through versus connecting traffic (such distinction does not exist in ground traffic).

Let us count the number of rows in the passenger flow constraint. The number of nodes, N , in the network is replicated in as many copies as the number of O-D pairs: $N(N-1)$. Equation 3:2:11 could take up to

*Figure 3-2-3 puts both the 'route network' and 'highway map' of our three-city system side by side for a comparison. There is no such thing as a 'one-stop route' in a highway network. A route network, besides having one-stop routes, also shows the segments of the one-stop route, with travel times, e.g., ${}^1t_{21}^1$ and ${}^1t_{13}^1$. Furthermore, there are non-stop routes such as the one shown with travel time ${}^0t_{12}^1$. A highway map, on the other hand, only has links between adjacent city pairs.



ROUTE NETWORK



HIGHWAY MAP

FIG. 3-2-3 DIFFERENCES BETWEEN A ROUTE NETWORK AND A HIGHWAY MAP

$N(N-1)$. The linking constraint Equation 3:2:12 written to bundle up the various O-D flows in a segment, could take up to the number of segments in the network. Assuming 1 1/2 segments per route, there are $1/5 |R|$ segments in the whole network and hence that many rows in Equation 3:2:12. For an 80-city system, and assuming two route candidates generated per city pair (a rather conservative figure), there could be 524,560 rows in the passenger flow all together!

3.2.3 Objective Function of the Integer Program

The mathematical program seeks to maximize income to the airline operator. Income is revenue minus direct operating cost minus indirect operating cost.

While the operator is trying to maximize profit, the travelling public is concerned with minimizing their travel time from origin to destination. This presents us with a game-theoretic problem where there are two decision makers: the operator and the travelling public. The operator, in order to achieve high profitability, must align its aircraft routes with the preferred passenger routings so as to capture the highest market share of passenger patronage.

revenue

Revenue is O-D passenger, p_{pq} , times yield, y_{pq} . p_{pq} can be either a function of the type of routings (non-stop/multi-stop/connect) or simply a fixed number. In our formulation, p_{pq} is to be interpreted as the potential O-D demand between a city pair. p_{pq} is either connected or disjoint depending on the route structure. The potential O-D demand p_{pq} is served if the integer program

decides that it is a lucrative city pair market and a route network is configured to connect p and q. Otherwise, pq will be disjoint, and there will be no traffic between the city pair. The actual demand would be zero and there will be no revenue from that city pair.

For the sake of illustrative clarity, the O-D demand p_{pq} in our example formulation (shown in Figure 3-2-2) is a fixed number and not a function of non-stop, multi-stop or connect routings. The revenue in such a case is $\sum_{pq} y_{pq} p_{pq} x_{pq}$, where x_{pq} is either "1" or "0" corresponding to whether city pair market pq is served by the route map.

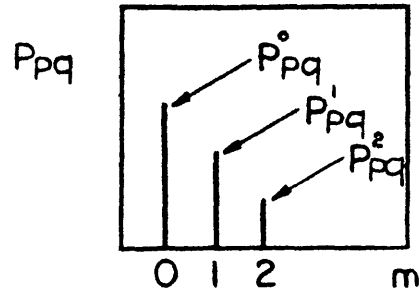
A MORE GENERALIZED DEMAND FUNCTION

A more general formulation of demand where p_{pq} is a function of the type of routings is shown in Figure 3-2-4. In the demand function shown, p_{pq}^0 is the passenger demand if a non-stop routing is provided between p and q. p_{pq}^1 is the passenger demand if a one-stop direct routing is provided, and so forth. Similarly, if a one-stop connect routing involving one transfer is scheduled, p_{pq}^{11} passengers will be attracted, etc.

The effect of O-D frequency on demand has not been considered in the demand functions of Figure 3-2-4. Frequency is assumed fixed at the practical upper limit of the demand-frequency curve as shown in Figure 3-2-5. The demand figures used in R.I.S.E. is therefore the passenger patronage corresponding to a "saturated" market in which no appreciable demand can be captured with additional frequencies.

The effect of O-D travel time on demand has also been neglected. Travel time is assumed to be fixed at the minimum time for the type of

THROUGH



CONNECTING

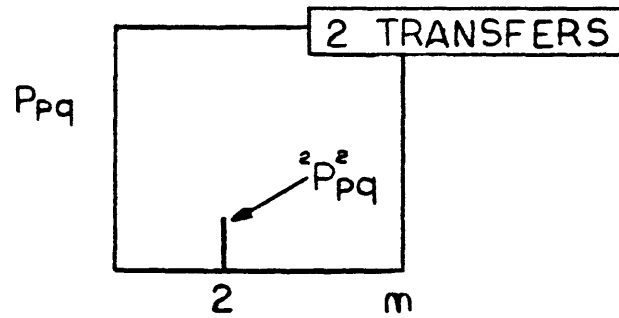
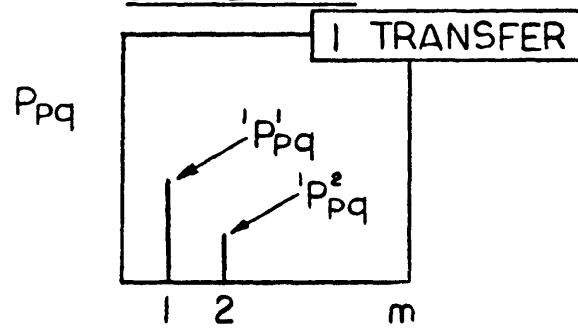


FIG. 3-2-4 O-D DEMAND FUNCTION

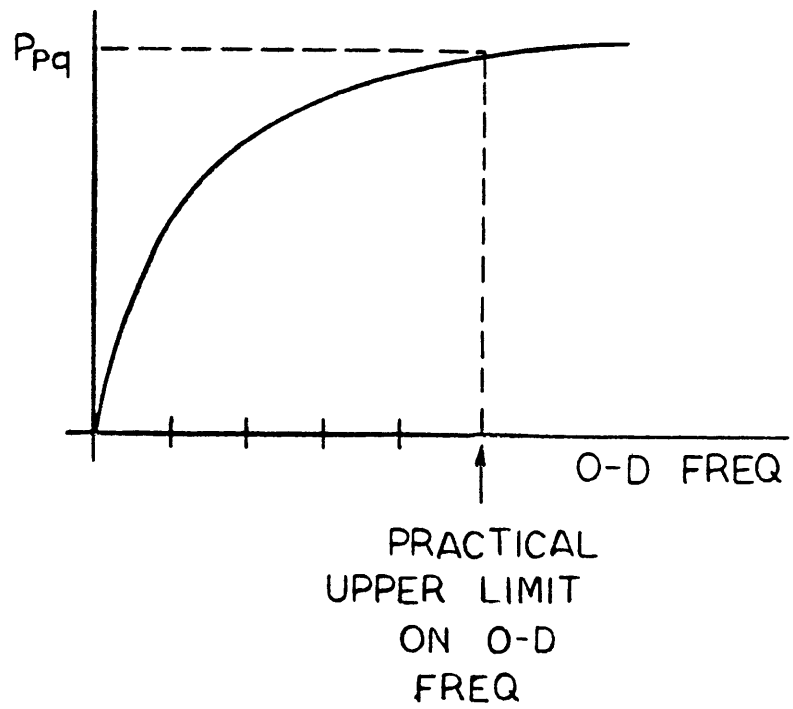


FIG. 3-2-5 DEFINITION OF P_{pq}

routing. For example, p_{pq}^1 would be the O-D demand corresponding to the shortest one-stop routing between p and q. Similarly, ${}^1p_{pq}^1$ would be the demand corresponding to the shortest two-stop connect routing between pq involving only one transfer.*

It is assumed that O-D demands are additive. For example, if both a non-stop and one-stop routings exist to serve city pair pq, the total induced demand is the sum of the demand corresponding to a non-stop routing and the demand corresponding to a one-stop routing, i.e., $p_{pq}^0 + p_{pq}^1$. Similarly, if both a non-stop routing and a one-stop connect routing serve the city pair, the total demand would be $p_{pq}^0 + {}^1p_{pq}^1$, etc.**

*The minimum-time assumption is made for computational efficiency. It can be bypassed in a straightforward manner at a much higher computational cost.

**By assuming that demands are additive, computational speed and storage requirements are greatly reduced. The assumption can be bypassed in a straightforward manner at a much higher computational cost.

The above more generalized demand function calls for some revision of some of the connectivity and passenger flow constraints. Constraint Equation 3:2:8 will now read:

$$\sum_{r \in R_{pq}^m} y_r^m - o_{pq}^m \geq 0 \quad \text{for all } m, pq \quad (3:2:13)$$

where R_{pq}^m is the set of routes connecting pq via m -stops

Constraint Equation 3:2:9 will read

$$\frac{1}{|R_{pq}^m|} \sum_{r \in R_{pq}^m} y_r^m - l_{pq}^m \geq 0 \quad \text{for all } l, m, pq \quad (3:2:14)$$

where R_{pq}^m is the set of routes connecting pq via m -stops in exactly l connections

Constraint Equation 3:2:10 becomes

$$\text{for all } y_r^m, y_r^m, \in R_{pq}^m$$

$$(y_r^m - y_r^m) = 0$$

for all pq served by more than 1 connect routing, and

$$\text{for all } l, m \quad (3:2:15)$$

And constraint Equation 3:2:11 becomes

$$\sum_m \sum_r \sum_i \sum_{pq} p_{ik}^{m,r} - \sum_m \sum_r \sum_j \sum_{pq} p_{kj}^{m,r} = \begin{cases} \sum_l \sum_m l_{pq}^m x_{pq}^m \\ 0 \end{cases} \quad (3:2:16)$$

for all pq and k

$k = p$ origin; $k = q$ destination; $k =$ Bypass node

Each row in Equation 3:2:8, Equation 3:2:9 and Equation 3:2:10 has now been expanded to M rows, where M, as defined previously, is the number of intermediate stops in the longest route. Equation 3:2:9 has to be further replicated L times, where L is the largest number of transfers allowed in a routing.

We summarize the general expression for revenue(REV):

$$REV = \sum_{pq} y_{pq} \sum_m \sum_{\ell} \ell_{pq}^m \ell_{pq}^m \quad (3:2:17)$$

Cost

Cost is composed of direct operating cost (DOC) and indirect operating cost (IOC). DOC is cost per block hour times the block hours per flight times the frequency (flights/cycle)

$$DOC = c_a t_r^m a_n^m$$

where a_n^m = frequency (flights/cycle) along the r^{th} m-stop route using aircraft a

Frequency can be estimated if the maximum segment flow, the average load factor on the segments, f_L , and the seat capacity of an aircraft type a, s_a , is known. Dividing $\max_{ij} ({}^m p_{ij}^r)$ by $f_L s_a$, and rounding off to the next higher integer gives the frequency:

$$a_n^m = \left\langle \frac{1}{f_L s_a} \max_{ij \in L_r^m} ({}^m p_{ij}^r) \right\rangle$$

where L_r^m = set of link segments (i, j) making
up route m-r

$\langle \cdot \rangle$ = operator to round off to the next higher
integer

For route r and aircraft type a, the direct operating per cycle
can be written as

$$a_{c_r}^m = c_a t_r^m \max_{ij \in L_r^m} \left\langle \frac{m_{p_{ij}}^r}{f_L s_a} \right\rangle$$

and DOC for the whole system is

$$DOC = \sum_a \sum_m \sum_r a_{c_r}^m = \sum_a c_a \sum_m \sum_r t_r^m \max_{ij \in L_r^m} \left\langle \frac{m_{p_{ij}}^r}{f_L s_a} \right\rangle \quad (3:2:18)$$

IOC is usually estimated by linear regression. The typical explanatory
variables are average passenger traffic per cycle, PAX, total system
departures per cycle, DEP, and revenue passenger miles per cycle, RPM.

$$IOC = c_o + c_p PAX + c_D DEP + c_R RPM$$

PAX can be written as

$$PAX = \sum_{pq} p_{pq} x_{pq} = \sum_{pq} \sum_m \sum_l l_{p_{pq}}^m l_{x_{pq}}^m$$

DEP can be written as

$$DEP = \sum_a \sum_{m,r} |L_r^m| a_{n_r}^m = \sum_a \sum_{m,r} |L_r^m| \left\langle \frac{1}{f_L s_a} \max_{ij \in L_r^m} (c_{a_{ij}}^{m,r}) \right\rangle$$

And RPM can be written as

$$RPM = \sum_{pq} \sum_m \sum_{r \in R_{pq}^m} \sum_{ij \in L_r^m} m_{ij}^r m_{pq}^r$$

where we have used time distance instead of mileage distance. The coefficient c_R is defined in a consistent manner.

DESCRIPTIVE TRAFFIC FLOW

The above revenue passenger mile expression is based on a particular passenger flow pattern $\{m_{pq}^r\}$. In R.I.S.E., passengers are modelled to minimize their travel time from O to D. To represent this descriptive traffic flow behavior, we define the operator $\min_t^{(p-q)}$ to denote that the passenger flow from p to q is distributed in the network along the minimum time path (as shown by the expression \min_t)

Hence

$$\min_t^{(p-q)} \left(\sum_m \sum_{r \in R_{pq}^m} \sum_{ij \in L_r^m} m_{ij}^r m_{pq}^r \right)$$

would mean that trips are executed from origin p to destination q in the descriptive manner. The expression for revenue passenger miles can now be put in its final form

$$RPM = \sum_{pq} \min_t^{(p-q)} \left(\sum_m \sum_{r \in R_{pq}^m} \sum_{ij \in L_r^m} m_{ij}^r m_{pq}^r \right)$$

which simply sums up all the O-D trips from p to q.

Having defined the expressions for PAX, DEP, and RPM, we can write the equation for indirect operating cost, IOC, as the sum of

these three terms:

$$\begin{aligned}
 \text{IOC} = & c_0 + c_p \sum_{pq} p_{pq} x_{pq} + c_D \sum_{\bar{a}} \sum_{m,r} |L_r^m| \max_{ij \in L_r^m} \langle a_{ij}^{m,r} / f_L s_a \rangle + \\
 & c_R \sum_{pq} \min_{t(p-q)^m} \sum_{r \in R_{pq}^m} \sum_{ij \in L_r^m} m_{t_{ij}}^{m,r} p_{pq}^{m,r} \quad (3:2:19)
 \end{aligned}$$

Notice that for illustrative clarity, we have shown a tableau (Figure 3-2-2) with only one aircraft type. It can be checked that if we rewrite constraint Equation 3:2:12 in several parts, the multi-aircraft type case can be handled easily. First, we write

$$\sum_{pq} m_{pq}^{m,r} p_{ij}^{m,r} - m_{pq}^{m,r} p_{ij}^{m,r} = 0 \quad \text{for all } m, r \text{ and } ij \quad (3:2:20)$$

Second, Equation 3:2:12 is written as many times as there are aircraft types,

$$\sum_{pq} m_{pq}^{m,r} p_{ij}^{m,r} - a_{pq}^{m,r} p_{ij}^{m,r} \geq 0 \quad \text{for all } a, m, r \text{ and } ij \quad (3:2:21)$$

Finally, we specify that only enough aircraft types are assigned to handle the traffic

$$\sum_{\bar{a}} m_{\bar{a}}^{m,r} a_{pq}^{m,r} - m_{\bar{a}}^{m,r} p_{ij}^{m,r} = 0 \quad \text{for all } m, r \text{ and } ij \quad (3:2:22)$$

Assuming there are 1 1/2 segments in a route on the average, we have added $3|\bar{a}||R|$ more rows to Equation 3:2:12, where $|\bar{a}|$ is the

average number of aircraft types assigned to a route.

Income

Income is revenue minus cost: $INC = REV - DOC - IOC$. Substituting the appropriate terms by the expressions 3:2:6, 3:2:17 and 3:2:18

$$\begin{aligned}
 INC &= \sum_{pq} (y_{pq} - c_p) \sum_m \sum_{\ell} \ell_{p,pq}^m \ell_{x,pq}^m - \sum_a \sum_m \sum_r (c_a^a t_r^m + |L_r| c_D) \\
 &\quad \max_{ij \in L_r^m} \langle \frac{m_{p,r}}{a_{ij}^r} / f_L s_a \rangle - c_R \sum_{pq} \min(\sum_{r \in R_{pq}^m} \sum_{ij \in L_r^m} m_{t,ij}^r m_{p,ij}^r) \\
 &\quad - c_o
 \end{aligned}$$

$$\begin{aligned}
 &= \sum_{pq} y'_{pq} \sum_m \sum_{\ell} \ell_{p,pq}^m \ell_{x,pq}^m - \sum_a \sum_m \sum_r m_{c_a}^r \max_{ij \in L_r^m} \langle \frac{m_{p,r}}{a_{ij}^r} / f_L s_a \rangle \\
 &\quad - c_R \sum_{pq} \min(\sum_{r \in R_{pq}^m} \sum_{ij \in L_r^m} m_{t,ij}^r m_{p,ij}^r) - c_o
 \end{aligned}$$

$$\text{where } y'_{pq} = y_{pq} - c_p \text{ and } m_{c_a}^r = c_a^a t_r^m + |L_r| c_D$$

The objective function seeks to maximize income (or profit, or net revenue) per cycle. c_o , being a constant, is dropped from the expression for INC.

$$\begin{aligned}
\max f &= \sum_{pq} y'_{pq} - \sum_{\bar{m}} \sum_{\ell} \ell_{pq}^{\bar{m}} \ell_{pq}^{\bar{m}} \\
&- \sum_{R_{pq} \in \mathcal{P}^{\bar{m}}} m_{pq} \ln \left(\sum_{r \in R_{pq}^{\bar{m}}} \sum_{ij \in L_r^{\bar{m}}} m_{ij}^r p_{pq}^r \right) \\
&- \sum_a \sum_{\bar{m}} \sum_r m_{ca}^r \max_{ij \in L_r^{\bar{m}}} \left\langle \frac{m_{ij}^r}{f_L s_a} \right\rangle \quad (3:2:23)
\end{aligned}$$

discussion

The objective function takes into account the equilibrium computation of demand and supply. On the demand side is the potential O-D passenger $\ell_{pq}^{\bar{m}} \cdot \ell_{pq}^{\bar{m}}$ is a demand function since it is responsive to the level of service expressed in terms of non-stop versus multi-stop versus connecting routes. On the supply side is the airline trying to offer the various kinds of connectivity level of service (non-stop, multi-stop, connecting or disjoint) between the O-D pairs in order to capture the passenger demand. The equilibrium point represents a finalized route map providing the connectivity to serve an economically profitable portion of the potential O-D demand market.

Let us count the number of variables. (0-1) variables $\ell_{pq}^{\bar{m}}$ could be $3M N(N-1)$, where M and N , as defined previously, are the maximum number of multi-stops and the number of cities in the network. Passenger flow variables by O-D, route and segment, m_{pq}^r , could be as large as $3N^2(N-1)^2(M+1)$ where we have assumed that, on the average, there are only two route candidates generated per city pair (a rather

conservative figure) and there are 1-1/2 segments per route. Segment flow variables, ${}^m p_{ij}^r$, could take up to $3(M+1)N(N-1)$. For an 80-city system, and two-stops being the longest route considered, the mathematical program can take up to 341,602,800 variables!

We have only counted the variables that appear in the objective function. The route variables, y_r^m 's, have not been considered. The y_r^m 's are 'coordinating variables' and there are as many as the number of route candidates included in the mathematical program. The number is large, as mentioned numerous times before. The route candidates define the route network. The route network provides the connectivity between city pair markets, and it facilitates passenger flow. Although y_r^m 's do not appear in the objective function, it is through $\{y_r^m\}$ (i.e., the route network) that ${}^l x_{pq}^m$, ${}^m p_{pq}^r$ and ${}^m p_{ij}^r$ are associated with each other (see the structure of the tableau in Fig. 3-2-2, particularly see how y_r^m link the variables ${}^l x_{pq}^m$ and ${}^m p_{pq}^r$). y_r^m 's coordinate the variables on the demand (revenue) side ${}^l x_{pq}^m$, and those on the supply (cost) side, ${}^m p_{pq}^r$ and ${}^m p_{ij}^r$, to bring about a demand/supply equilibrium (profit maximization). We are continuously coordinating the revenue and cost (demand and supply) sides as we generate route candidates y_r^m .

3.2.4 Discussion on the Route Selection Integer Program

Let us look back on the whole integer program (summarized in Fig. 3-2-6) and try to answer these questions:

- (a) What aspects of the 'real world' have we included in the model?

Figure 3-2-6 ORIGINAL INTEGER PROGRAMMING FORMULATION

Function

$$\sum_{pq} y'_{pq} \left(\sum_m \sum_{\ell} \ell_{pq}^m x_{pq}^m - c_R \sum_{pq} \min \left(\sum_m \sum_{r \in R_{pq}^m} \sum_{ij \in L_r^m} m_{ij}^{tr} \right) \right) - \sum_a \sum_m \sum_r m_{ca}^r \max_{ij \in L_r^m} \left\langle \frac{m_{ij}^r}{f_L \cdot s_a} \right\rangle \quad (3:2:23)$$

its

petition requirements

$$\sum_{m=n}^M \sum_{r \in R_{pq}^{n,m}} y_r^m \geq 1 \quad \text{for all } n \text{ and the specified } pq \quad (3:2:1)$$

Connectivity

$$\sum_{r \in R_{pq}^{o,m}} y_r^m - x_{pq}^m \geq 0 \quad \text{for all } m, pq \quad (3:2:13)$$

$$1/|\ell_{pq}^m| \sum_{r \in \ell_{pq}^m} y_r^m - x_{pq}^m \geq 0 \quad \text{for all } \ell, m, pq \quad (3:2:14)$$

d - Figure 3-2-6 ORIGINAL INTEGER PROGRAMMING FORMULATION)

$$(y_r^m - y_r^m) = 0 \quad \text{for all } y_r^m, y_r^m, \epsilon, \lambda_{pq}^m \quad (3:2:15)$$

for all pq served by more than one connect routing

and for all ℓ, m

enger flow

$$\sum_m \sum_r \sum_i \sum_{pq} \lambda_{pq}^{m,r} p_{ik}^{m,r} - \sum_j \sum_{pq} \lambda_{pq}^{m,r} p_{kj}^{m,r} = \begin{cases} -\sum_{\ell} \sum_m \lambda_{pq}^{\ell,m} x_{pq}^{\ell,m} & k - \text{origin} \\ \sum_{\ell} \sum_m \lambda_{pq}^{\ell,m} x_{pq}^{\ell,m} & k - \text{destination for all pq and k} \\ 0 & k - \text{bypass node} \end{cases} \quad (3:2:16)$$

$$\sum_{pq} \lambda_{pq}^{m,r} p_{ij}^{m,r} - \lambda_{pq}^{m,r} p_{ij}^{m,r} = 0 \quad \text{for all } m, r \text{ and } ij \quad (3:2:20)$$

$$\sum_{pq} \lambda_{pq}^{m,r} p_{ij}^{m,r} - a_{pq}^{m,r} p_{ij}^{m,r} \geq 0 \quad \text{for all } a, m, r \text{ and } ij \quad (3:2:21)$$

$$\sum_a \lambda_{pq}^{m,r} a_{pq}^{m,r} - \lambda_{pq}^{m,r} a_{pq}^{m,r} = 0 \quad \text{for all } m, r \text{ and } ij \quad (3:2:22)$$

$\lambda_{x_{pq}}^m$ and y_r^m - (0-1) variable $\lambda_{pq}^{m,r}$ and $p_{ij}^{m,r}$ - integer variable (positive)



- (b) What type of mathematical program are we dealing with?
Linear or nonlinear, convex or nonconvex, single or
multiple objective?

An airline operates in a regulatory environment provided by the C.A.B. The model takes as input a given fare structure and route authority. The route authority restrictions are not explicitly shown in constraints of the integer program, since the formulation is only for the selection phase of R.I.S.E. The generation phase of R.I.S.E. has already guaranteed that every route candidate is allowed by the C.A.B. route authority.

An airline operates in a competitive market. The competition requirement constraint takes into account the route competition between a city pair. Given that competitor X is serving city pair A-B by a one-stop route the airline concerned can match up the competitive service by specifying in the constraint that A-B be served by a one-stop or better.

An airline faces a passenger O-D demand that is responsive to the level of service. In the integer program, O-D passenger demand for an airline has been modelled as a function of the quality of service the airline offers. A non-stop service between a city pair would attract a different number of O-D passenger patronage than say a connecting service. Given this demand function facing an airline, the integer program decides on the best level of service to offer between a city pair.

An airline tries to satisfy their customers--the travelling public. Passengers, as individual consumers, try to minimize their travel time from O to D. Given a schedule, a passenger would take the flight and connection most convenient for himself. This 'descriptive' behavior is simulated by the passenger flow constraint and the expression in the objective function for each O-D pair:

$$\min_{t(p-q)} \left(\sum_m r \in R_{O_{pq}}^m \sum_{ij \in L_R^m} m_{ij}^r m_{pq}^r \right)$$

Among the prime objectives of an airline is profit maximization. The objective function of our model seeks to maximize profit to the operator given the regulatory/competitive environments and the passenger behavior.

We will identify the type of integer formulated above. The constraints are all linear. It follows that the constraints define a convex hull. The complication comes from the objective function. Conceptually, the objective function is of the form (refer to Figure 3-2-6 for the actual objective function)

$$\max \{ z = h(x) - \min_{\sum_i a_i} \left(\sum_i a_i b_i^{(x)} \right) - \max_j \langle g_j(x) \rangle \} \quad (3:2:24)$$

Notice there are $\min_{\sum_i a_i}$ and \max_j operators 'nested' within a max operator. The nexted $\min_{\sum_i a_i}$ operator comes from a descriptive passenger flow. The nested \max_j operator comes from frequency estimation. The objective function is therefore a nonlinear, nonconcave, and nonconvex function. It is best described as being discontinuous, or 'zigzag'.

The above Equation 3:2:24 assumes a dual objective. While the airline operator is maximizing system profit, each transportation user (i.e., the passenger) is minimizing his/her own travel time. There are two decision makers, typical of a game-theoretic formulation. The equilibrium solution to the "game" played between the operator and passengers would be for the operator to align their routes to the preferred minimum-time passenger routings, so as to capture the maximum system profit.

The biggest hurdle of solving R.I.S.E. is not so much the nonlinear, nonconcave and nonconvex properties of the objective function of the integer program. The problem lies in trying to carry out an optimization problem before the constraints and the objective functions are defined. Remember we are making an assumption when we show the above integer programming formulation. Namely, that all the route candidates have been generated. The integer program in such a case is reduced to a route selection problem. The set of route candidates that can be generated is finite. But it is a typical combinatorial problem where finiteness also means large. Literally thousands and thousands of routes that are allowed by the C.A.B. route authority can be generated. It is impossible, in terms of the dimension we will be dealing with, to include all the legal routes (or even those 'reasonable', e.g., noncircuitous ones) in the integer program for selection. However, the objective function and constraints will not be defined unless all the qualified route candidates y_r^m have been generated. As discussed before, the generation and selection problems

cannot be separated. The only approach is to do generation and selection simultaneously. In other words, we are dealing with an optimization problem where the objective function and the constraints are not known explicitly a priori. The objective function and constraints are incrementally developed as we carry out the optimization problem. We are exploring the shape of an unknown objective function and constraints as we carry out the solution algorithm. The R.I.S.E. solution algorithm carries out column and row generation at the same time that it improves on the current 'feasible solution' in a 'hill climbing' fashion typical of an ill-behaved optimization problem (see Wilde and Beightler [1967]).

Let us, for a moment, pretend that a magic route screening criterion have reduced the potential route candidates per city pair to two as input to the selection integer program. (Notice that by limiting ourselves to only such a small subset of route candidates we may have rejected the global optimum to start with.) Let us further assume that passenger O-D demand is perfectly inelastic, i.e., P_{pq} is not a function of whether the route is non-stop/one-stop/two-stop or connecting, and that we are dealing with only one aircraft type. We will still have $N(N-1)(M+N+2) + 1.5|R|$ rows and $3N(N-1)[N(N-1)(M+1) + 2M + 1]$ variables. In an 80-city system with two-stops being the longest route, it would mean 553,000 constraints and 341,571,200 variables--a formidable problem!

Hopefully by now, the type and complexity of optimization problem we are dealing with in R.I.S.E. has been identified. A method will

be discussed in the next section which the author personally believes would solve the problem. It combines generation and selection in a decomposition type approach.

3.3 Generation and Selection: A Decomposition Approach

The first section of this chapter deals with the generation of route candidates. The second section deals with the selection among these candidates. The current section will combine these two processes and we will see how they work together simultaneously.

3.3.1 Simultaneous Route Generation and Selection

Sections 3.1 and 3.2 have emphatically pointed out that it is not feasible to generate all 'reasonable routes' and then select among these routes in an integer program. Route generation and selection have to be carried out simultaneously. Routes should be generated only as needed. This means that the conventional notion of defining a feasible region and then searching among the region for the optimum does not apply in R.I.S.E. Because the complete set of feasible routes are too expensive to generate computationally, we generate only those few routes which hold the greatest promise to be selected. In the R.I.S.E. optimization problem the feasible region is so expansive that we cannot define the boundary of it a priori. Only those few spots where the chance of obtaining the optimum is the greatest will be identified and search is carried out only among these few spots. Put in another way, the exact expressions for the constraint equations and the objective function cannot be put down at the start of the algorithm. More constraint equations are generated and more variables (routes) are

included in the objective function and the mathematical program as the algorithm proceeds. We refer to this process as column and row generation. In order to facilitate column/row generation in the simultaneous generation and selection algorithm, the original integer program (Figures 3-2-2 and 3-2-6) has to be transformed into a decomposable formulation. The current section is devoted to discussing how to rewrite the integer program, as given in the last section, into a decomposable form. It serves as a transition from the model formulation discussion of the present chapter to the model solution discussion of the next chapter.

3.3.2 The Primal Decomposition Formulation

When dealing with large mathematical programs with a 'block diagonal' or 'staircase' structure (as is the case of our integer program), we often have to resort to decomposition. Some of the classical papers on decomposable systems are Dantzig and Wolfe [1961], Tomlin [1966], Jarvis [1969], and Benders [1962]. Simpson [1969] discussed a decomposable fleet planning model. Manheim [1966] viewed the highway location problem as a multi-level decision process. Chan [1969] addressed a special case of the network investment problem as a two level problem--the 'aggregate' and the 'detailed' levels. The more recent works in the subject include Geoffrion [1970] and Mesarovic, et. al. [1970].

In this section, we will transform our original integer program tableau (Figure 3-3-1) into a decomposable form (Figures 3-3-2 and

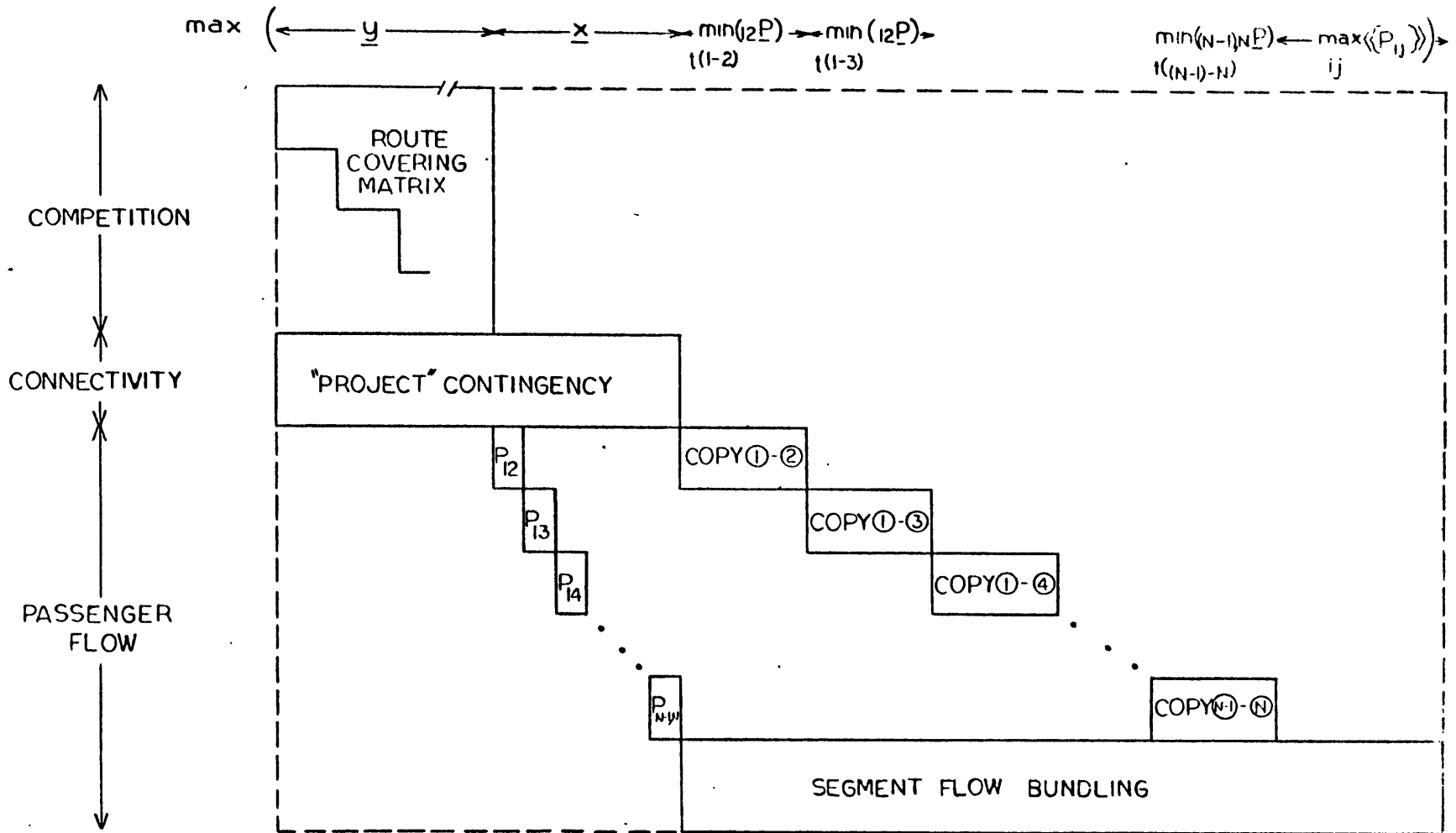


FIG. 3-3-1 THE ORIGINAL FORM OF THE INTEGER PROGRAM

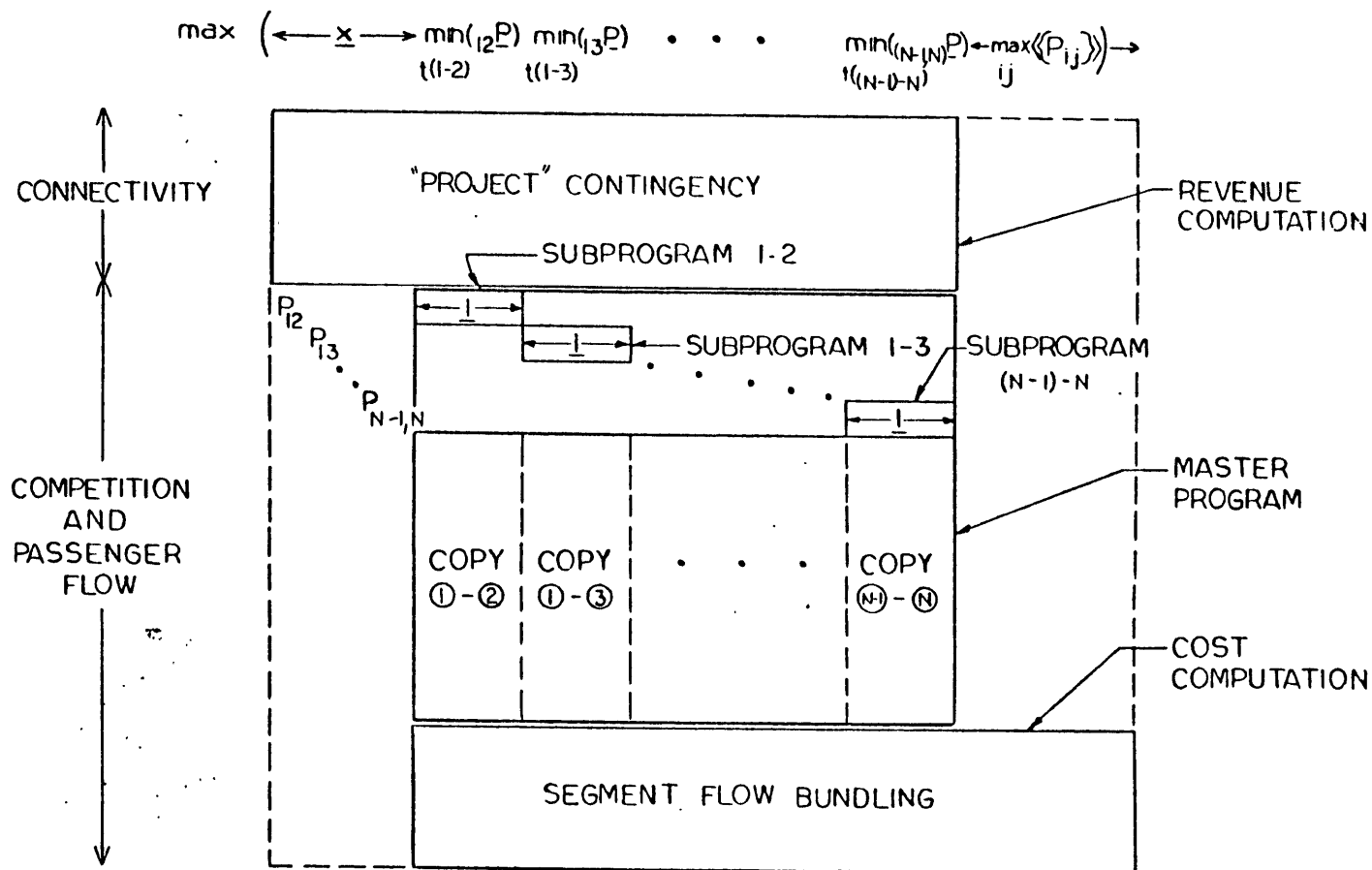


FIG. 3-3-2 THE DECOMPOSABLE FORM

Figure 3-3-3 THE DECOMPOSABLE FORMULATION

Objective Function

$$\max z = \sum_{pq} y'_{pq} \sum_m \sum_{\ell} \ell_{pq}^m x_{pq}^m - c_R \sum_{pq} \min_{\substack{m,r \\ \epsilon_{pq}^C}} t_{(p-q)} \left(\sum_{m,r} m_{pq}^{tr} p_{pq}^{pr} + \sum_{\substack{m,r,r' \\ \epsilon_{pq}^{C'}}} m_{pq}^{tr'} p_{pq}^{pr'} + \sum_{\substack{m',m_1,m_2,\dots,r' \\ \epsilon_{pq}^{C''}}} m'_{pq} t_{r',m_1,m_2,\dots}^{m_1,m_2,\dots} p_{pq}^{m_1,m_2,\dots} \right)$$

$$-\sum_a \sum_{m,r} m_{pq}^{cr} \max_{ij \in L_r} \left\langle \frac{m_{pq}^{r}}{a_{pq}^{ij}} / f_L \cdot s_a \right\rangle \quad (3:3:9)$$

Constraints

Master program

$$\sum_{\substack{m,r \\ \epsilon_{pq}^R}} m_{pq}^{pr} + \sum_{\substack{m,r,r' \\ \epsilon_{pq}^{R'}}} m_{pq}^{pr'} + \sum_{\substack{m,r',m_1,m_2,\dots \\ \epsilon_{pq}^{R''}}} m_{pq}^{m_1,m_2,\dots} = \sum_{\ell} \sum_m \ell_{pq}^m x_{pq}^m \quad \text{for all } pq \quad (3:3:1)$$

$$\sum_{m=n}^M \sum_{r \in R_{pq}^m} (m_{pq}^{pr}) \geq 1 \quad \text{for all } n \text{ and the specified } pq \quad (3:3:2)$$

(Continued - Figure 3-3-3 THE DECOMPOSABLE FORMULATION)

Revenue computation

$$\sum_{\substack{m', r \\ \in R_{pq}^m}} (p', q', p^r) - o_{x_{pq}}^m \geq 0 \quad \text{for all } m, pq \quad (3:3:3)$$

$$1/|\ell_{pq}^{R^m}| \sum_{\substack{m', r \\ \in \ell_{pq}^{R^m}}} (p', q', p^r) - \ell_{x_{pq}}^m \geq 0 \quad \text{for all } \ell, m, pq \quad (3:3:4)$$

$$(p', q', p^r) - (p'', q'', p^{r'}) = 0 \quad \text{for all } m'-r, m''-r' \in \ell_{pq}^{R^m} \quad (3:3:5)$$

for all pq served by more than one
connecting service

and for all ℓ, m

(Continued - Figure 3-3-3 THE DECOMPOSABLE FORMULATION)

Cost computation

$$m_{ij}^r + \sum_{pq} \sum_{\substack{r' \in \\ m_{Cij}^r}} p_{pq}^{m_{r'}} + \sum_{pq} \sum_{m', m_1, m_2, \dots, r'} \epsilon_{pq}^{m_{Cij}^r} \frac{m_{pq}^{m_1, m_2, \dots, r'}}{p_{pq}^{m_{r'}}} - \frac{m_{ij}^r}{a_{ij}^r} \geq 0 \quad (3:3:6)$$

for all a, m, r, ij

$$m_{ij}^r + \sum_{pq} \sum_{\substack{r' \in \\ m_{Cij}^r}} p_{pq}^{m_{r'}} + \sum_{pq} \sum_{m', m_1, m_2, \dots, r'} \epsilon_{pq}^{m_{Cij}^r} \frac{m_{pq}^{m_1, m_2, \dots, r'}}{p_{pq}^{m_{r'}}} - \sum_a \frac{m_{ij}^r}{a_{ij}^r} = 0 \quad (3:3:7)$$

for all m, r, ij

x_{pq}^m 0-1 variable $p_{pq}^{m_{r'}}$, $p_{pq}^{m_{r'}}$, $\frac{m_{pq}^{m_1, m_2, \dots, r'}}{p_{pq}^{m_{r'}}}$, $\frac{m_{ij}^r}{a_{ij}^r}$ integer

3-3-3). The original problem is cast into a 'master program' and the associated 'subprograms' in the spirit of Dantzig and Wolfe, Tomlin and Benders. First, let us briefly review the above three pieces of literature.

In view of the large dimensions of the original tableau, the above mentioned authors transform the original problem into an equivalent, but much smaller, master program as a start. Columns or rows are annexed one at a time to the master program only 'as needed'. The philosophy is that hopefully the algorithm converges to optimality before too many columns or rows are generated. The column/row generation is carried out as follows. After one iteration of the master program, each subprogram receives new input from the master. With these inputs (e.g., in Dantzig and Wolfe/Tomlin, these inputs are the dual variables of the linking constraint of the master, which modify the cost coefficients of the subprogram), the subprograms are solved. The subprogram solution(s) suggest(s) the most promising column/row to be annexed to the master. (This step is analogous to route selection in R.I.S.E.). The master is resolved and the next iteration begins. The algorithm terminates when optimality condition is achieved in the master. (The check on optimality is analogous to the evaluation step in R.I.S.E.)

R.I.S.E. decomposition works in a similar manner except that a primal approach, instead of the above dual method, is employed. Instead of using a dual price to communicate between the master program and subprograms, its primal analog—marginal profit—serves the role.

Through a three-node example network, we show here how the original 'staircase' integer program tableau (see Figures 3-2-6 and 3-3-1) is transformed into its equivalent problem (Figures 3-3-2 and 3-3-3), with its master program and subprograms.

master program

The transformation boils down to combining the route covering matrix and the different copies of node-arc incidence matrices (shown in Figure 3-3-1) into an arc-chain tableau (shown as the master program in Figure 3-3-2). By doing so, we greatly reduce the dimensionality. Such a transformation is an extension of the papers by Tomlin [1966] and Jarvis [1969], who gave a physically appealing network flow interpretation of the Dantzig-Wolfe decomposition. In a graphic form, we show how the node-arc incidence matrix formulation is cast into an arc-chain formulation in Figure 3-3-4.

Notice that the transformation results in fewer rows but at the expense of more columns. The row reduction is a result of the fact that each O-D copy of node-arc incidence matrix is 'squeezed' down to a single row of 1's, while the linking constraint has the same number of rows in both tableaux. Let us refer to our three-node example tableau in Figure 3-3-5. This is an equivalent formulation of the original tableau of Figure 3-2-2. The constraints on connectivity and constraint Equation 3:2:12 on flow bundling are essentially unchanged. The constraints on competition requirements and constraint Equation 3:2:11 on O-D passenger flow have been cast into an arc-chain formulation shown in the part of the tableau outlined by heavy lines

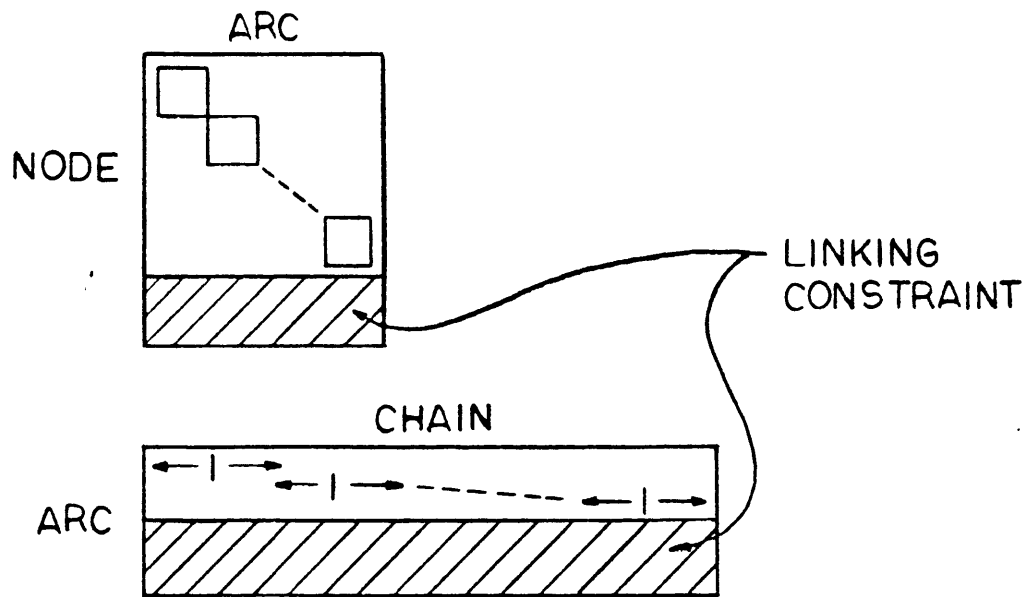


FIG. 3-3-4 NODE - ARC vs ARC-CHAIN FORMULATION

(Figure 3-3-5). This arc-chain tableau has the 'arcs', or more precisely the O-D pairs, on its rows and the chains/routes on its columns. It has 'multi-copies' by O-D pair. The left-most copy is for city pair 1-2. The next copy for 1-3 ... up to the last copy for 3-2.

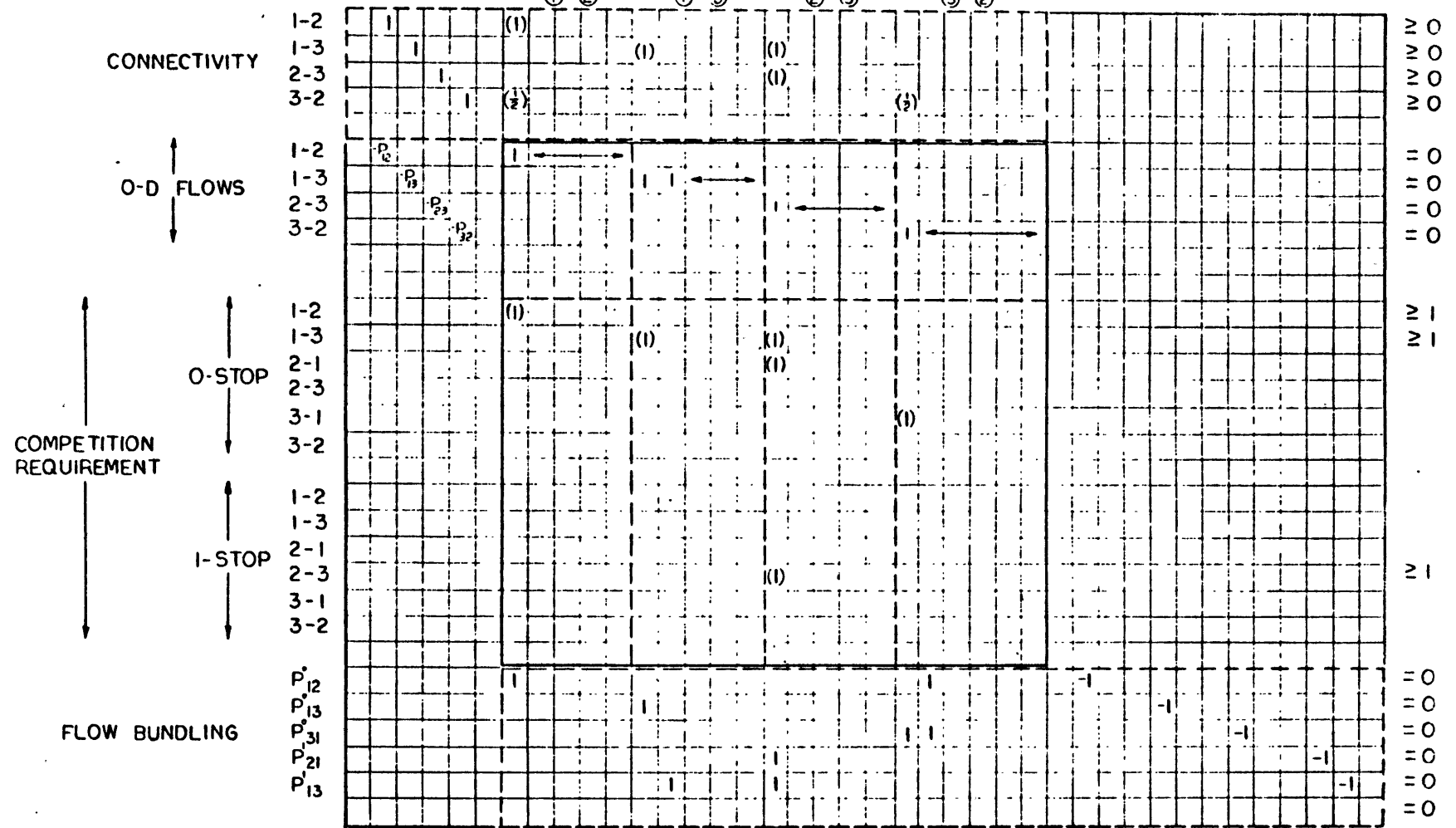
In our notation, a route is identified by (i) the number of intermediate stops m , (ii) a sequence number, and (iii) the origin city p and termination city q . As an example, $m \cdot r: p - \dots - q$ denotes the r^{th} m -stop route originating at p and terminating at q . Notice also that the route $m \cdot r = p - k_1 - \dots - k_m - q$ may provide routings for through and connect passengers. Also we denote ${}_{pq}^m p^r$ as the passenger from p to q travelling in the r^{th} m -stop route $p - \dots - q$, ${}_{p'q'}^m p^r$ as the flow from p' to q' in the through service routing r' of route $m \cdot r = p - \dots - q$, and ${}_{p''q''}^m m_1 m_2 \dots$ as the flow from p'' to q'' in the m -stop connect routing r'' utilizing a number of different routes and making intermediate transfers at $m_1, m_2 \dots$. Examples of these notations will be given in the paragraph below.

We pointed out that an r^{th} m -stop route makes it possible to carry passengers from diverse O-D pairs via through or connect routings. These routings, which are made possible because of the existence of r^{th} m -stop route, would appear in the corresponding O-D copies. As an example (Figure 3-3-5) the first of the one-stop routes 2-1-3 can carry passengers from 2 to 1, 1 to 3 and 2 to 3, we find the four variables ${}_{21}^1 p^1$, ${}_{13}^1 p^1$, ${}_{23}^1 p^1$ in the corresponding O-D copies for 2-1, 1-3 and 2-3. (Due to the fact that no demand exists between 2-1, the copy

FIG. 3-3-5
ARC-CHAIN
FORMULATION

$$\max \left\{ \sum_{i_2} y_{i_2}^i p_{i_2}^i + \sum_{i_3} y_{i_3}^i p_{i_3}^i + \sum_{i_2, i_3} y_{i_2, i_3}^i p_{i_2, i_3}^i + \sum_{i_2, i_3} y_{i_2, i_3}^i p_{i_2, i_3}^i - C_R \left[\frac{\min(t^i)}{t(1-2)} + \frac{\min(t^i + t^i)}{t(1-3)} + \frac{\min(t^i)}{t(2-3)} + \frac{\min(t^i + t^i)}{t(3-2)} \right] - c' \left(\frac{1}{f_{i_2}} \right) + c' \left(\frac{1}{f_{i_3}} \right) + c' \left(\frac{1}{f_{i_2, i_3}} \right) + c' \max \left(\frac{1}{f_{i_2, i_3}}, \frac{1}{f_{i_3, i_2}} \right) \right\} \quad 204$$

$x_{i_2}^i \quad x_{i_3}^i \quad x_{i_2, i_3}^i \quad x_{i_2, i_3}^i \quad p_{i_2}^i \quad p_{i_3}^i \quad p_{i_2, i_3}^i \quad p_{i_2, i_3}^i \quad p_{i_2}^i \quad p_{i_3}^i \quad p_{i_2, i_3}^i \quad p_{i_2, i_3}^i \quad p_{i_2}^i \quad p_{i_3}^i \quad p_{i_2, i_3}^i \quad p_{i_2, i_3}^i$



2-1 is not shown). Notice that besides the route (represented by flow variable $\frac{1}{21}p^1$), we include the routings $\frac{1}{13}p^1$ and $\frac{1}{23}p^1$, in the column of the arc-chain tableau. Another example of a routing is the connection made at 1 in going from 3 to 2 (i.e., 3-1-2) which shows up in copy 3-2 as $\frac{1}{32}p^1$. Let us explain the clumsy adscripts of $\frac{1}{32}p^1$. $\frac{m}{pq}p^r$ is read as the r^{th} m -stop connection route from p to q . The routing is made up of two portions, connecting at the transfer station m_1 . An example would make this clear. $\frac{1}{32}p^1$ denotes that the first one-stop connect routing 3-1-2 between 3 and 2 is consisted of two trips--3-1 and 1-2, making a connection at 1. Each of the trips, 3-1, 1-2 is a non-stop segment. In general, the connection may involve more than one transfer station, $\frac{m}{pq}p^r$, should be generalized to $\frac{m}{pq}p^{r,m_1,m_2 \dots}$.

The former 0-D copies of node-arc incidence matrices are 'squeezed' into rows of 1's along the diagonal of the arc-chain tableau. There is a physical interpretation of this "squeezing" transformation. Take copy 1-3, the row of 1's constitute the following constraint:

$$\frac{0}{13}p^2 + \frac{1}{23}p^1 = P_{13} x_{13}$$

where x_{13} is a 0-1 variable denoting whether city pair 1-3 is served.

which reads that the 0-D passengers from 1 to 3 can travel both on the second non-stop route 1-3 and a segment (1,3) of the first one-stop route 2-1-3. It can be thought of as saying that a fraction of the solution 0-D flow goes on route 0.2 and the other fraction goes

via route 1.1. These two routings constitute two smaller problems. The rows of 1's combine these small problems solutions to arrive at the solution for an O-D problem for city pair 1-3. These rows of 1's appear in a similar context in the arc-chain formulations of Tomlin [1966] and Jarvis [1969].

Theoretically, if we assume passengers do take circuitous routings to 'see the country', there is a high number of routings that carry passengers between a city pair p and q . For example, between city pair 1-3, it is possible that a passenger would backtrack by going to city 2 and then make a connection via the one-stop route 2-1-3 to 3 (refer to Figure 3-3-6). If this circuitous routing is considered, we would have another 1 in the row of 1's for copy 1-3, thus increasing the number of columns. To avoid the 'curse of dimensionality', only the most reasonable route or routing (e.g., the shortest route/routing) is annexed to the column of the tableau. In this way we only generate the column (route/routing) as needed, in the full spirit of the column generation scheme of Dantzig and Wolfe. We will come back to this point later.

Let us formally write down the constraint equation corresponding to the rows of 1's.

$$\sum_{\substack{m,r \\ \epsilon_0 R_{pq}}} m_{pq}^{r} + \sum_{\substack{m,r,r' \\ \epsilon_0 R'_{pq}}} m_{pq}^{r,r'} + \sum_{\substack{m,r' \\ \epsilon R''_{pq}}} m_{pq}^{m_1, m_2, \dots} = p_{pq} x_{pq}$$

for all pq

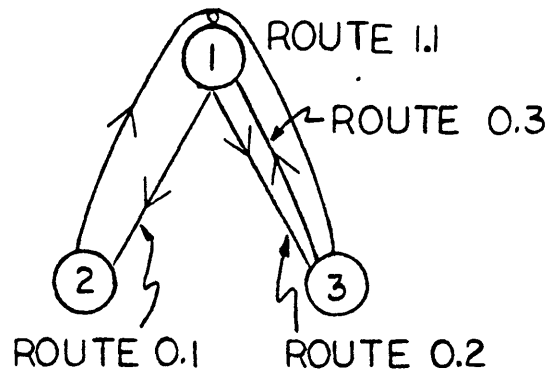


FIG. 3-3-6 EXAMPLE NETWORK

where: $R_{o pq}$ is the set of routes connecting pq.
 $R'_{o pq}$ is the set of through routings connecting p-q
 R''_{pq} is the set of connect routings connecting p-q.

For O-D demands which are a function of the type of routing (i.e., whether non-stop, one-stop, two-stop, through or connect), the above constraint has to be replicated M by L times, where M is the number of intermediate stops in the longest route and L is the largest number of transfers allowed in a connect routing (as defined in Section 3.2). The passenger flow node-arc incidence matrices represented by Equation 3:2:16 have now been transformed into an equivalent arc-chain format

$$\sum_{\substack{m,r \\ \in R_{o pq}}} x_{pq}^{m,r} + \sum_{\substack{m,r,r' \\ \in R'_{o pq}}} x_{pq}^{m,r,r'} + \sum_{\substack{m,r',m_1,m_2 \\ \in R''_{pq}}} \dots x_{pq}^{m,r',m_1,m_2} \\ = \sum_{\ell,m} \ell_{pq}^m x_{pq}^{\ell,m} \quad \text{for all } pq \quad (3:3:1)$$

The rest of the arc-chain tableau is similar to the route covering matrix of our original integer programming formulation (see constraint Equation 3:2:1 of Figure 3-2-6). Essentially, if an O-D pair is covered by a route, a 1 is entered under the route in the row corresponding to the city pair. For example, the route 2-1-3 covers 2-1, 1-3 and 2-3, a 1 appears under $\begin{matrix} 1 & 1 \\ 2 & 3 \end{matrix}$ in the rows corresponding to 2-1, 1-3 and 2-3, (refer to Figure 3-3-5). The competition require-

ment that a city pair is to be covered by an m -stop or better routing is specified by writing a " ≥ 1 " on the row corresponding to the city pair concerned. The only complication in the arc-chain formulation is that instead of dealing with the former (0-1) binary route variable y_r^m , we are dealing with a pure integer variable. We introduce the operator (\cdot) , where we define (\cdot) to be unity if the variable exists and $(\cdot) = 0$ if it does not. This converts the entries of the arc-chain covering matrix ${}_{pq}^m p^r$ we are dealing with back to the familiar 0-1 variable y_r^m found in the route covering matrix of our original integer programming formulation. This operator (\cdot) however, converts what used to be a set of linear constraints to a set of non-linear ones. Let us formally write the competition requirement constraint.

$$\sum_{m=n}^M \sum_{r \in \bar{R}^m} ({}_{pq}^m p^r) \geq 1 \quad \text{for all } n \text{ and the } p \text{ (3:3:2)}$$

specified pq

(To refresh the reader's memory on notation, n is the number of intermediate stops in a routing. Also the adscript " n " was there in the original integer program formulation to denote the different copies of the O-D pairs repeated along the rows, the first copy for non-stop routings, the second for one-stop routings, etc., hence accounting for the triangular shape of the route covering matrix (refer to Figure 3-3-1).)

In the above equation we are summing over the routes only. The routings (through or connect) do not enter into the picture in this

constraint at all. One final note—by specifying the competition requirement within the arc-chain tableau, we have eliminated one set of binary route variables y_r^m , thus decreasing the column dimension of the tableau.

We will show that unlike the way we indicated in the original integer programming formulation, it is actually not necessary to replicate the copy of O-D pairs $(M+1)$ times along the rows of the route covering matrix. Only one copy will suffice. The following is an intuitive argument why this is so. Recall that the route covering format allows for the specification of competition requirements on the route map. By putting a '1' corresponding to an O-D pair in row copy n of the route covering matrix, the user specifies that the city pair must be served by a n -stop routing or better. The key words are 'or better'. If the user of R.I.S.E. specifies that a city pair A-B must be served by two-stop or better services due to competitive pressure, it would be logically inconsistent if he again specified that the same city pair must be served by one-stop or better. It is either that city pair A-B must be served by two-stops or better or one-stop or better. The user simply does not know what he is doing if he specifies A-B to be served by both two-stops or better and one-stop or better. This means if a '1' appears in the row corresponding to A-B in the two-stop row copy of the route covering matrix, there would not be a '1' for AB in the one-stop row copy, or vice-versa. To rephrase it, the '1'

constraint can only be written for city pair A-B once in all the (M+1) row copies of the route covering matrix. The finalized master program with each city pair appearing only once in the route covering matrix is shown in Figure 3-3-7.

DISCUSSION

Let us count the number of rows and columns in the master program. Constraint Equation 3:3:1 takes $N(N-1)$ rows, which can be approximated as N^2 , and in the case of a symmetric demand matrix function, $N^2/2$ (where N is the number of cities and M the number of intermediate stops in the longest route). Similarly, constraint Equation 3:3:2 takes at most $N(N-1)$ rows. For a symmetric route map, we can approximate by $N^2/2$. This number $N^2/2$ can be further reduced since not every city pair in the system is connected by through service. As an example, in the 26-city system of Northeast Airlines, only 64 city pairs are connected by through service (as of February 1971, and counting only one way). This represents about 20% of the $N^2/2$ city pairs. This reduction factor of 20% can theoretically be applied towards $N^2/2$ to arrive at $N^2/10$. However, this reduction factor will vary from airline to airline. The R.I.S.E. software package has to be coded to cater for the worst case. Assume (i) a 50% reduction factor, (ii) symmetric O-D demand matrix function, (iii) symmetric route map, there would be $N^2/2 + .5 \times N^2/2 = 4800$ rows for an 80-city system.

While we have 'squeezed down' the number of rows considerably in our present master program (notice we have not discussed the dimensions

12^v_o 13^v_o 23^v_1 32^v_o 32^v_1
 12^p_o 13^p_o 13^p_1 23^p_1 32^p_o 32^p_1

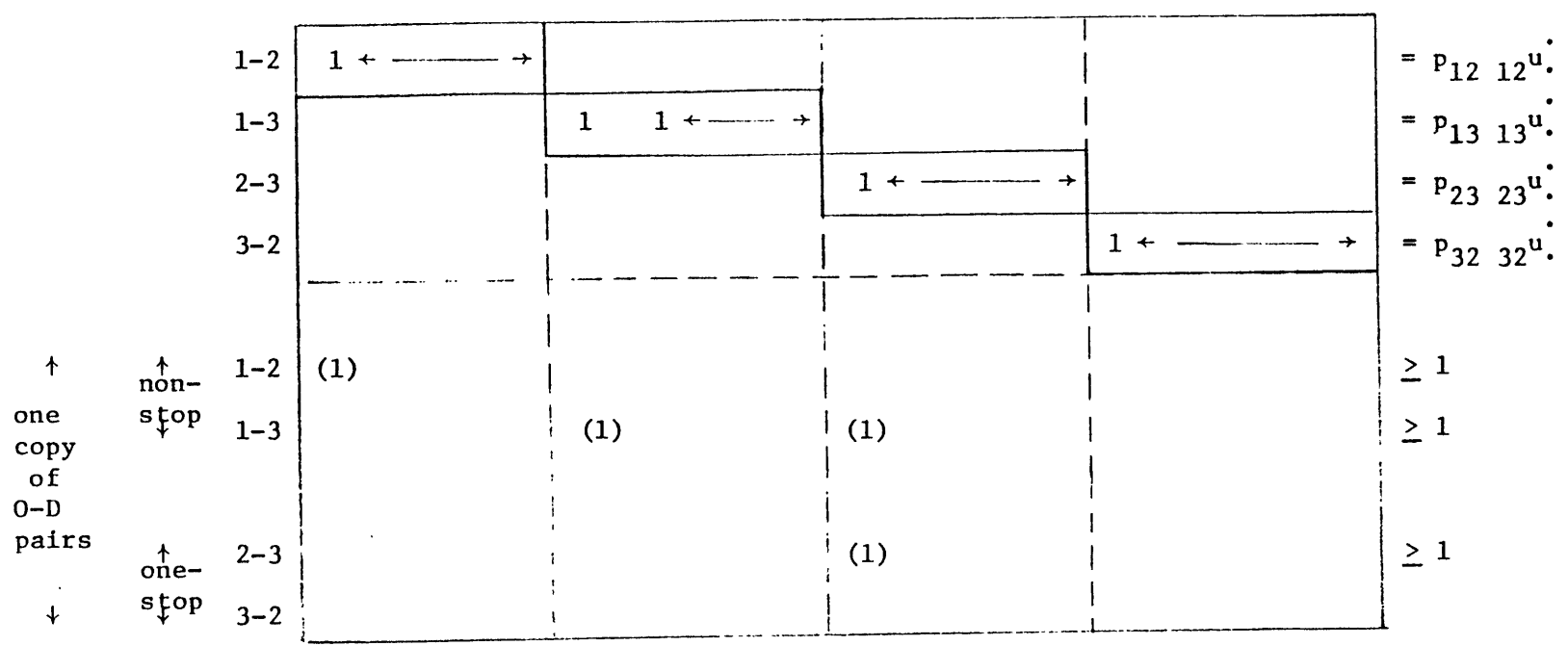


Figure 3-3-7 THE MASTER PROGRAM

of the subprograms yet), the number of columns (routes and routings) remains large, if not larger than before. As we mentioned earlier, the decomposition procedure reduces the number of rows at the expense of the columns. The whole merit of the decomposition approach is that routes/routings will be generated judiciously only 'as needed', thus limiting the number of columns to a minimum.

Let us count the number of columns. The columns are route and routing variables. By passenger routing we mean the path a passenger follows in executing his trip from O to D. The routing, as defined in R.I.S.E., may consist of segments from different routes if the passenger makes connections. Assuming (i) 50% of the city pairs are connected by through service routes, and the remaining 50% by connections, (ii) each route has 1.5 segments on the average, (iii) each route generates 1.5 through service routings and (iv) all routes and routings are symmetric, we expect $1.5 \times 1.5 \times N^2/2 \times .5 + N^2/2 \times .5 = .8125 N^2$ columns. For an 80-city network we could expect 5200 columns.

subprograms

Each origin-destination pair constitutes a subprogram. For an N-city system, we have subprograms for city pairs 1-2, 1-3, ... , (N-1) - N. For each O-D pair, there exists a corresponding row of 1's in the master tableau (see Figure 3-3-5). Since each O-D subprogram is solved sequentially, we expect to increase the number of the rows of 1's in the tableau as the solution algorithm proceeds. In this sense, more and more rows are generated as additional O-D

subprograms are tackled.

The solution of the subprogram for city pair p-q involves generating one or more aircraft routes, or connect passenger routing between p and q. With the demand function as defined in Section 3.2.2 (Figure 3-2-4), only the shortest route (and its associated 'through' routings) and/or the shortest connect routing between p and q need to be found. The route/routings constitute the chains to be considered as columns for the master program. The routes/connect-routings with the largest marginal profit will be chosen to enter the master tableau in our column generation procedure, as will be explained immediately below.

marginal profit computation

In a dual decomposition method such as that of Tomlin [1966] and Dantzig/Wolfe[1961], the selection of route/routing columns to be annexed to the master program is based on dual prices. The primal decomposition of R.I.S.E. does not employ dual prices. Instead routes/connect-routings are selected according to its marginal profit during the column generation procedure.

There are two parts to the marginal profit calculation--the revenue part and the cost part as shown in Figure 3-3-3. The marginal profit of a route is evaluated by solving for the revenue potential and cost potential corresponding to the particular route. The route with the best marginal profit is annexed to the master program. This, in a nutshell, is the column generation scheme of R.I.S.E.

1. REVENUE COMPUTATION

The revenue potential of a route is computed by solving the 'connectivity' constraint of the original formulation (refer to Figs. 3-2-2 and 3-2-6). The revenue computation determines whether a city pair is connected (served) and hence revenue can be expected from the passengers that travel between that city pair. The revenue potential of a route is a complex function of the network structure. The addition of a route not only brings about new revenue from all the routings between all the city pairs, within that route, but it might also be the 'missing link' which makes a connect routing between a formerly disjoint city pair possible. If this is the case, additional revenue is expected from the connect traffic.

If we replace the route variables y_r^m in connectivity constraint of the original integer program by $(p'q'p^r)^m$, we have the constraint equations for revenue computation. We will work out the case of fixed demand (i.e., perfectly inelastic demand) first and then generalize to a demand which is a function of the routing:

$$\sum_{\substack{m,r,p'q' \\ \epsilon_{o^R}^{pq}}} (p'q'p^r)^m - x_{pq} \geq 0 \quad \begin{array}{l} \text{for all } pq \\ \text{for through service} \end{array}$$

$$\sum_{\ell} \frac{1}{\ell^R_{pq}} \sum_{\substack{m,r,p'q' \\ \epsilon_{\ell^R}^{pq}}} (p'q'p^r)^m - x_{pq} \geq 0 \quad \begin{array}{l} \text{for all } pq \\ \text{for connect service} \end{array}$$

$$\begin{aligned}
 (p',q',p^r) - (p'',q'',p^{r'}) &= 0 && \text{for all } m \cdot r, m \cdot r' \in \ell R_{pq} \\
 &&& \text{for all } pq \text{ served by more than} \\
 &&& \text{one connection service, and} \\
 &&& \text{for all } \ell
 \end{aligned}$$

In the last equation, we again use (\cdot) to convert p',q',p^r to the former y_r^m .

It is rather straightforward to extend to the case of a demand function:

$$\sum_{\substack{m',r \\ \in \ell R_{pq}^m}} (p',q',p^r) - o_{x_{pq}}^m \geq 0 \quad \text{for all } m, pq \quad (3:3:3)$$

$$\frac{1}{|\ell R_{pq}^m|} \sum_{\substack{m',r,p',q' \\ \in \ell R_{pq}^m}} (p',q',p^r) - \ell x_{pq}^m \geq 0 \quad (3:3:4)$$

for all ℓ, m, pq

$$(p',q',p^r) - (p'',q'',p^{r'}) = 0 \quad (3:3:5)$$

for all $m \cdot r, m'' \cdot r' \in \ell R_{pq}^m$
for pq served by more than
one connection service
for all ℓ, m

DISCUSSION

Let us count the number of constraints. We will assume (i) a symmetric route network and (ii) half of the city pairs are served by

through service while the remaining half by connections. Equation 3:3:3 takes up $N^2/2 \cdot |\bar{R}_{pq}|$, where $|\bar{R}_{pq}|$ is the average number of direct routes serving city pair pq . Equation 3:3:4 takes up $1/2 \cdot N^2/2 \cdot |\bar{C}_{pq}|$, where $|\bar{C}_{pq}|$ is the average number of connection routings serving a city pair. If we have assumed that out of the city pairs served by connections 50% can be connected by more than one routing. Equation 3:3:5 takes up $1/4 \cdot N^2/2 \cdot |\bar{C}_{pq}^1| \cdot \bar{\ell}$, where $\bar{\ell}$ denotes the average number of transfer stations in a connect routing. For the case of

$|\bar{R}_{pq}| = 1$, $|\bar{C}_{pq}| = 1$, $|\bar{C}_{pq}^1| = 2$ and $\bar{\ell} = 2$, we have $N^2/4 + N^2/8 + N^2/2 = 5600$ rows for an 80-city system.

The columns involved in revenue computation consist of the route and routing variables x_{pq}^m 's and the connectivity variables λ_{pq}^m . Since the x_{pq}^m 's have been counted in the master program, we need to count only the λ_{pq}^m 's. If we assume (i) two-stops being the longest route, (ii) only one transfer station in connecting service, and (iii) a symmetric demand matrix function, we would have $5/2 N^2$ columns. For an 80-city system, we have 16,000 columns.

It has to be noted that it is not necessary to write the full set of revenue constraints for all the routes in the network. Only several rows and columns, instead of the whole set need to be solved for each O-D route subprogram.

II. COST COMPUTATION

The second part of calculating the route marginal profit is cost computation. Cost computation involves estimating the direct

operating cost (DOC) and the part of the indirect operating cost (IOC) associated with system departures. DOC of a route is evaluated by multiplying the DOC per block hour, c_a , by the length of the route in block hours, t_r^m , and by the route frequency n_r^m . The IOC associated with system departures is the unit cost per departure, c_D , times the number of departures, D . And we recall that D is the product of the average number of segments in a route $|\overline{L}_r^m|$ and the route frequency n_r^m . Therefore, once route frequency is known, computing route cost is a matter of multiplication.

The main task in computing cost is frequency estimation. We recall that the route frequency is the frequency of the route segment carrying the largest flow volume in the route. We further recall that in our multicopy traffic flow formulation, a segment flow could be made up of through or connect traffic from diverse O-D pairs. In our original integer programming formulation, the various copies of node-arc incidence matrices are linked together by a flow bundling constraint. The flow bundling constraint 'bundles up' the through and connect traffic on a segment, allowing segment frequency (and thus route frequency) to be computed. Once route frequency is known, DOC and the part of IOC related to system departure for the route is directly obtainable. The cost subprogram in our present arc-chain formulation is essentially the flow bundling constraints 3:2:20, 3:2:21 and 3:2:22 of the original formulation.

The following constraint says that segment flow is a bundle of

through and connect traffic from diverse O-D's.

$$\begin{aligned}
 & m_{p_{ij}}^r + \sum_{pq} \sum_{r' \in \epsilon_{pq}^{m,r}} m_{pq}^{m,r} + \sum_{pq} \sum_{m', m_1, m_2, \dots, r' \in \epsilon_{pq}^{m', m_1, m_2, \dots}} m_{pq}^{m', m_1, m_2, \dots} \\
 & - \sum_a m_{a_{ij}}^r \geq 0 \quad \text{for all } a, m, r, ij \quad (3:3:6)
 \end{aligned}$$

In the above equation, the first term is the local traffic from i to j and segment (i, j) , the second term is the through traffic within the same route while the third term is the connect traffic from different connection routes. $\epsilon_{pq}^{m,r}$ stands for the set of chains (through or connect routings) that facilitate O-D flow from p to q in a segment ij of route $m \cdot r$.

The following additional equation says that only enough aircraft types are assigned to handle the traffic on the route:

$$\begin{aligned}
 & m_{p_{ij}}^r + \sum_{pq} \sum_{r' \in \epsilon_{pq}^{m,m}} m_{pq}^{m,r} + \sum_{pq} \sum_{m', m_1, m_2, \dots, r' \in \epsilon_{pq}^{m', m_1, m_2, \dots}} m_{pq}^{m', m_1, m_2, \dots} \\
 & - \sum_a m_{a_{ij}}^r = 0 \quad \text{for all } m, r, ij \quad (3:3:7)
 \end{aligned}$$

DISCUSSION

Let us count the number of constraints involved in cost computation. Equations 3:3:6 and 3:3:7 take up $|R| |\bar{L}_r^m| \bar{a}$ and $|R| |\bar{L}_r^m|$ rows respectively,

where $|\bar{L}_r^m|$ is the average number of segments in a route and \bar{a} is the average number of aircraft types assigned to a route. Assuming that on the average, there is one route per city pair, each route is 1.5 segments long and flown by 1.1 aircraft types. For an 80-city system where the route map is symmetric the total number of rows amounts to $|R| |\bar{L}_r^m| (\bar{a}+1) \approx N^2/2 \times 1.5 \times 2.1 = 10,080$.

The number of columns involved in cost computation are made up of two types of variables--the route and routing variables, $\overset{m}{p}_{pq}$ and the segment variable $\overset{m}{a}_{ij}^r$. The $\overset{m}{p}_{pq}$'s have been counted while we discuss the master program. We will count only the $\overset{m}{a}_{ij}^r$'s here. Given the same assumptions about the number of routes per city pair, segments per route and aircraft types, there are $.825N^2$ columns. For an 80-city system, this amounts to 5280.

It has to be noted that it is not necessary to write the full set of cost constraints for all the routes in the network. Only several rows and columns, instead of the whole set, need to be solved in order to select the next route to enter the master program.

objective function

The objective function is profit maximization, where profit is revenue minus DOC minus IOC. DOC is cost per block hour, c_a , times block hour, t_r^m , times frequency, n_r^m . IOC is related by regression to the explanatory variables passenger traffic, PAX, system departure, DEP, and revenue passenger miles, RPM.

$$IOC = c_o + c_p PAX + c_D DEP + c_R RPM \quad (3:3:8)$$

The expression for maximizing the system income (profit) is very similar to that in our original integer program formulation.

$$\begin{aligned}
 \max z = & \sum_{pq} y'_{pq} \sum_{m,l} \ell_{pq}^m \ell_{pq}^m \\
 -c_R \sum_{pq} \min_{t(p-q)} & \left(\sum_{m,r} t_r^m \ell_{pq}^{m,r} + \sum_{m,r,r'} t_r^m \ell_{pq}^{m,r'} + \sum_{m,r,m',m_1,m_2 \dots m_1,m_2, \dots} t_r^m \ell_{pq}^{m',m_1,m_2, \dots} \right) \\
 & \epsilon_{pq}^C \qquad \qquad \qquad \epsilon_{pq}^{C'} \qquad \qquad \qquad \epsilon_{pq}^{C''} \\
 - \sum_{a,m,r} m C_a^r & \max_{ij \in L_r^m} \left\langle \frac{m}{a} \ell_{ij}^r / f_L \cdot s_a \right\rangle \qquad (3:3:9)
 \end{aligned}$$

where C_{pq} = the set of route chains serving pq
 C'_{pq} = the set of through routings serving pq
 C''_{pq} = the set of connect routings serving pq

To refresh our reader's memory, y'_{pq} is a 'discounted' yield per passenger figure because $y'_{pq} = y_{pq} - c_p$ where we have taken the coefficient of P in Equation 3:3:8 from the actual yield. $m C_a^r$ is an aggregate unit cost figure where we have combined the cost per block hour figure and the coefficient of DEP in Equation 3:3:8:

$$m C_a^r = c_a^a t_r^m + |\bar{L}_r^m| c_D$$

And the operator $\min_{t(p-q)} (\cdot)$ denotes that traffic is assigned according to the descriptive flow criterion among the chains that connect p to q.

Let us give some physical interpretations to the three terms of

Equation 3:3:9. The first term gives a discounted system revenue figure where the IOC per passenger has been taken off from the yield figure. The second term gives a system IOC figure related to the revenue passenger miles served. The last term gives a system cost figure that consists of both the DOC and the IOC per departure. The first term is evaluated by revenue computation, and the third term is evaluated by cost computation. If we examine the structure of the tableau in Figure 3-3-2 the first and second terms of the objective function lies directly above the cost computation constraints. The master program is situated in between the two sets of constraints, allowing communication between 'the revenue side of the picture' and 'the cost side of the picture'. The master program is an arc-chain formulation of the route network and it provides passenger flow information on the network. From the route network the revenue constraints evaluate the O-D connectivities to see whether a city pair is served. If a city pair is served, revenue could be expected from that city pair. From the passenger flow information on the route network, the cost constraints evaluate the route frequency and system departures. Once route frequency and system departure are obtained, we can estimate the cost of operating a route. The master program can be thought of as a coordinating mechanism between the revenue side (demand) and the cost side (supply) to bring about profit maximization (demand/supply equilibrium).

3.3.3 Route Network Configuration via Primal Decomposition

The R.I.S.E. optimization model, with its simultaneous route generation and selection feature, has been formulated in a decomposition

framework consisting of O-D subprograms and a master program.

The current section will discuss how the subprograms are coordinated with the master program through the use of 'marginal profit'. The marginal profit concept is the primal analog of the shadow prices of the dual decomposition method of Tomlin [1966] or Dantzig and Wolfe [1961].

We will examine in detail the coordination role assumed by the master program. In Figure 3-3-7 is shown the tableau of the master program. The lower half of the tableau is the familiar route covering matrix, in which the inter-airline route competition information is expressed as constraints. The upper half of the tableau is actually the interesting part. We will name it the revenue-cost matrix. On the rows of this matrix are the O-D pairs. On the column are the routes and routings. To each O-D route we assign a 'primal revenue' variable u_{pq}^m . To each route on the column we assign a 'primal cost' variable v_{pq}^r . A route $m \cdot r$ (which denotes the r^{th} m -stop route) is accepted as a new column for the master arc-chain tableau only if its associated primal revenue minus primal cost satisfies certain 'entry criteria'. The primal revenue and cost of a route are determined by solving several rows and columns of the revenue and cost constraints in conjunction with the master program.

The introduction of a route into the route network has its revenue implications and cost implications. On the revenue side, the introduction of a new route means that more city pairs are connected and/or served better. This means more revenue. Take a hypothetical

example from our three-city network shown in Figure 3-3-8. Let us say that the existing route network had only non-stop routes 1-2 and 3-2. We are considering the addition of one-stop route 3-1-2. The addition would mean opening up the revenue potential of the following city pair markets via through service: 2-1, 3-2. The primal revenue of the route u_1^1 is the sum of the potential revenue from all these new routings. Let us write down the shadow revenue of route 1-1.

$${}_{32}u_1^1 = y'_{21} \circ p_{21}^0 + y'_{32} \circ p_{32}^1$$

where the first term is the revenue from the non-stop passengers of the demand curve for city pair 2-1 and the second term is from the one-stop through service passengers of the demand curve for city pair 3-2. (The reader is reminded once again the y'_{pq} 's are the discounted yield figures, where the IOC associated with passenger traffic is subtracted from the yield per passenger.) Formally, we define the primal revenue for route $m \cdot r$ as:

$${}_{pq}u_r^m = \sum_{p'q',m} y'_{p'q'} \circ p_{p'q'}^m \quad (3:3:10)$$

$$\in S_{pq}^m$$

where S_{pq}^m = the set of cities served by through routings as a result of the introduction of route $m \cdot r$: $p - \dots - q$. If we define $\circ u_{p'q'}^m$ as

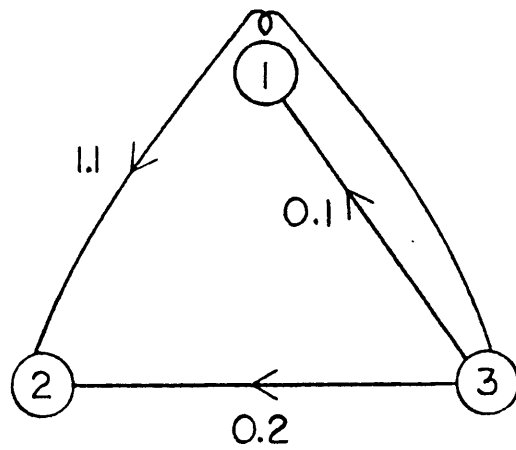


FIG. 3-3-8 3-CITY NETWORK

$y'_{p,q}$, $o_{p,q}^m$, the above expression becomes

$$y'_{p,q} = \sum_{p,q,m} \epsilon_{pq}^m o_{p,q}^m \quad (3:3:11)$$

The $o_{p,q}^m$'s are exactly the primal revenue variables we assign to each row of the revenue-cost matrix of the master program (refer to Figure 3-3-7).

On the cost side, the introduction of a new route means more system block hours of flying time and more system departures. This means more cost. Take the same hypothetical example from above. Enough frequency has to be provided to accommodate all the attracted through and connecting traffic. There is one complication, however. City pair 3-1 is now served by both the original non-stop route 0.1 and a segment of the one-stop route 1.1. The non-stop passenger demand from 3 to 1 is diverted into two routings, one via route 1.1 and the other via the first segment of route 1.1. What this means is that we can cut down on the frequency on route 0.1 due to the decrease traffic volume on this route. The cost of introducing route 1.1 is in part offset by the savings from decreasing the frequency on route 0.1. We write the following expression for the cost of operating route 1.1.

$$32 v_{11}^1 = \sum_a l_{c_a}^1 l_{n_a}^1 - \sum_{a'} o_{c_{a'}}^1 \Delta o_{n_{a'}}^1$$

where the first term is the cost of providing route 1·1 flown by aircraft types a and the second is the savings obtained by cutting back on the frequency of route 0·1 flown by aircraft types a'. We can formalize the above expression:

$$pq v_{r'}^m = \sum_a m_{c_a}^r m_{n_a}^r - \sum_{m',r'} \sum_{a'} m'_{c_{a'}}^{r'} \Delta m'_{n_{a'}}^{r'} \quad (3:3:12)$$

$$\epsilon_{pq} H_{r'}^m$$

where H_r^m is the set of routes whose frequency is reduced due to the introduction of route m·r and $m_{c_a}^r$ is an aggregate cost coefficient that includes both DOC per block hour and the IOC unit cost per departure.

If we write $pq v_{r'}^m$ as $\sum_a m_{c_a}^r m_{n_a}^r$ and $v_{r'}^{m'}$ as $\sum_{a'} m'_{c_{a'}}^{r'} \Delta m'_{n_{a'}}^{r'}$, we can

rewrite the above expression as

$$pq v_{r'}^m = pq v_{r'}^m - \sum_{m',r'} v_{r'}^{m'} \quad (3:3:13)$$

$$\epsilon_{pq} H_{r'}^m$$

Still, we have left out a cost item in the above route cost expression. We have not included the IOC associated with the revenue

passenger miles RPM. This inclusion is rather straight forward. The gross cost associated with RPM is

$$c_R (t_{pq}^{m,r} + \sum_{r' \in \epsilon_{pq}^m} C_{r'}^m \sum_{p',q' \in S_{r'}^m} t_{p',q'}^{m,r}) \quad (3:3:14)$$

where ϵ_{pq}^m is the set of routings introduced as a result of route $m:r: p- \dots -q$ and $S_{r'}^m$ is the set of O-D pairs served as a result of routings $m:r:r'$

The gross cost is offset by the savings resulting from the decreased traffic on the old routings.

$$c_D \left(\sum_{m',r'} \sum_{p',q'} t_{p',q'} \Delta_{p',q'}^{m,r} + \sum_{m',r',r''} \sum_{p',q'} t_{p',q'} \Delta_{p',q'}^{m,r} \right. \\ \left. \epsilon_{pq}^{H',m} \epsilon_{r'}^{S',m'} \quad \epsilon_{pq}^{H',m} \epsilon_{r''}^{S',r'} \right) \\ + \sum_{m',m_1,m_2,\dots,r''} \sum_{p',q'} t_{p',q'} \Delta_{p',q'}^{m,m_1,m_2,\dots} \epsilon_{pq}^{H',m} \epsilon_{r''}^{S',m_1,m_2,\dots} \quad (3:3:15)$$

where $\epsilon_{pq}^{H',m}$ = set of routings from where passenger flow has been diverted to new routings resulting from the introduction of route $m:r: p- \dots -q$.

$S_{r''}^{m'}/S_{r''}^{r'}/S_{r''}^{m_1,m_2,\dots}$ = set of city pairs whose O-D flows are diverted as the result of the introduction of route $m:r$.

Call Equation 3:3:14 and Equation 3:3:15 the net IOC cost associated with RPM for route $m \cdot r$, ${}_{pq}c_r^m$.

In our example, the old routings on route $0 \cdot 1$ and the new routing on segment $(3,1)$ of route $1 \cdot 1$ are equidistant in block hours. The IOC associated with ROM for the passenger flow from 3 to 1 is therefore the same before and after the introduction of route $1 \cdot 1$. The route primal cost is

$${}_{pq}v_r^m = {}_{pq}v'_r{}^m + {}_{pq}c_r^m \quad (3:3:16)$$

The ${}_{pq}v_r^m$'s are exactly the primal cost variables we assign to each route column of the revenue cost matrix of the master program (see Figure 3-3-7).

A similar set of primal cost and revenue equations can be written for each connect routing, such as the one-stop connect routing $3-1-2$. For the connect routing $m' \cdot m_1, m_2, \dots \cdot r''$ (which reads as the r'' th m' -stop connect routing making transfers at $m_1, m_2 \dots$) between O-D pair $p-q$, Equation 3:3:11 reads

$${}_{pq}u_{r''}^{m'} = \ell_{pq}^{m'} \quad (3:3:17)$$

Notice revenue is expected from a single O-D pair $p-q$, since by definition the connect routing carries only traffic from p to q . A generalized expression for Equation 3:3:11, including the case of a connect routing would be

$$p_{qr}^u{}^m = \sum_{p'q',m,\ell} \epsilon_{pq}^s{}^m \lambda_{p'q'}^m \quad (3:3:18)$$

Equation 3:3:13 reads the same except $p_{qr}^{v''m}$ stands for

$$p_{qr}^{v''m'} = \sum_{m,r} \epsilon_{pq}^c{}'' \sum_a m_c^r \Delta m_n^r$$

which sums up the cost associated with the increase in frequency on the routes that provide the connect routing.

Equations 3:3:14 and 3:3:15 remain unchanged if we read $m \cdot r \cdot r'$ as $m' \cdot m_1, m_2 \dots \cdot r''$.

The route marginal profit, defined below,

$$p_{qr}^w{}^m = (p_{qr}^u{}^m - p_{qr}^v{}^m) \quad (3:3:19)$$

will help to determine whether a route/connect-routing will enter the master tableau. The actual entry criterion for route/connect-routing entry is to make sure system profit is increased by the introduction of the candidate:

$$p_{qr}^w{}^m = \max \left(p_{qr}^w{}^m, \sum_{m',r'} \epsilon_{pq}^Q{}^m \Delta p'_{q'} w_{r'}^{m'} \right) \quad (3:3:20)$$

where Q_{pq} is the set of routes/connect-routings that serve a comparable set of demands as route $m \cdot r$, and Δ_{p,q,w_r^m} is the change in marginal profit in route $m' \cdot r'$ due to the introduction or removal of route $m \cdot r$.

The primal revenue $u_{pq,r}^m$ is supplied by the revenue computation. The primal cost $v_{pq,r}^m$ is supplied by the cost computation. The primal revenue and cost help in the selection of a route to enter the master tableau. The revenue and cost constraints, in return, receive the passenger flow information from the master program without which the primal revenue and cost cannot be determined. Such is the coordination between the master and subprograms in our primal decomposition procedure.

3.4 Summary

This chapter represents the formal discussion of the optimization formulation of the R.I.S.E. model. First, a graph-theoretic method is put forth to generate the comprehensive set of C.A.B. authorized routes. An integer programming formulation is then devised to select the optimal subset of routes to be included in the route network. Due to the huge combinatorial dimensionality, it is not feasible to separate the route generation and the route selection as two disjointed problems. It is computationally possible only if a few routes are generated at a time, from which the best is selected immediately. In other words, routes have to be generated and selected simultaneously.

To facilitate simultaneous generation and selection, the integer program has to be transformed into a decomposable format. Each handful of routes (and routings) generated graph-theoretically can then be readily appended as columns in the decomposable tableau for selection.

This chapter lays the algebraic background for an enumerative solution algorithm to be discussed in the next chapter.

REFERENCES

1. Avondo-Bodino, G. Economic Applications of the Theory of Graphs. Gordon and Breach (1962).
2. Bellman, R. E., K. L. Cooke and J. A. Lockett. Algorithms, Graphs and Computers. Academic Press (1970).
3. Benders, J. F. "Partitioning Procedures for Solving Mixed-Variables Programming Problems," Numerische Mathematik, vol. 4, pp. 238-252 (1962). (A more readable interpretation of this paper is found in Balinski's paper, "Integer Programming: Methods, Uses, Computation," Management Science, vol. 12, no. 3 (1965).
4. Chan, Y. P. "Optimal Travel Time Reduction in a Transport Network: An Application of Network Aggregation and Branch and Bound Techniques," M.I.T. Department of Civil Engineering Research Report R69-39 (July 1969).
5. Charnes, A. and W. W. Cooper. "Multicopy Traffic Network Models," Theory of Traffic Flow. (R. Herman ed.), Elsevier, Amsterdam (1961).
6. Dantzig, G. B. and P. Wolfe. "The Decomposition Algorithm for Linear Programs," Econometrica, vol. 29, pp. 767-778 (1961).
7. Dreyfus, S. E. "An Appraisal of Some Shortest-Path Algorithms," Operations Research, vol. 17, pp. 395-412 (May 1969).
8. Fulkerson, D. R. "Flow Networks and Combinatorial Operations Research," American Mathematical Monthly, vol. 73, no. 2, (February 1966).
9. Geoffrion, A. M. "Elements of Large-Scale Mathematical Programming," Management Science, vol. 16, no. 11 (July 1970).
10. Ichbiah, J. D. "Connectivity Analysis and Branch and Bound Methods," M.I.T. Department of Civil Engineering Ph. D. thesis (1967).
11. Jarvis, J. J. "On the Equivalence between the Node-Arc and Arc-Chain Formulations for the Multi-Commodity Maximal Flow Problem," Naval Research Logistics Quarterly, vol. 16, no. 4, pp. 525-529 (December 1969).

12. Levin, A. "Single Fleet Routing and Scheduling Problems for Air Transportation Systems," Third Israel Operations Research Conference (July 1969).
13. Manheim, M. L. Hierarchical Structure: A Model of Design and Planning Processes, Cambridge, Massachusetts. M.I.T. Press (1966).
14. Mesarovic, M., D. Macko, and Y. Takahara. Theory of Hierarchical, Multilevel Systems. Academic Press (1970).
15. Robert, P. O. and M. L. Funk. "Towards Optimum Methods of Link Addition in Transportation Networks," M.I.T. Department of Civil Engineering (September 1964).
16. Simpson, R. W. "Scheduling and Routing Models for Airline Systems," FTL Report R68-3, Department of Aeronautics and Astronautics (December 1969).
17. Sobel, R. A. "Domestic Route Authority Program," Lockheed California Company Commercial Transportation Research Department, CTR 1007, Issue 3 (June 1969).
18. Tomlin, J. A. "Minimum-Cost Multicommodity Network Flows," Operations Research, vol. 14, no. 1 (1966).
19. Weingartner, H. M. Mathematical Programming and the Analysis of Capital Budgeting Problems. Prentice-Hall (1963).
20. Wilde, D. J. and C. S. Beightler. Foundations of Optimization. Prentice-Hall (1967).

CHAPTER 4

A SOLUTION ALGORITHM FOR THE MODEL

In the last chapter, we have presented the R.I.S.E. optimization formulation. Route/routing generation was formulated in a graph-theoretic framework while the route selection problem was cast into an algebraic integer program format. Finally, the decomposable arc-chain formulation married the generation and selection problems, in the sense that the algebraic tableau used in route selection is structured to readily accept routes/routings generated graph-theoretically as columns. Still, a solution algorithm needs to be devised to solve the decomposable program. This will be the subject of the current chapter.

4.1 A Dynamic Programming Solution Scheme

The difficulty with solving the R.I.S.E. optimization model centers around three problems. First, it has a dual objective function in which system profit is maximized by the airline operator while individual travel time is minimized by the transportation user. We find min and max operators nested within a max operator. Second, an all integer solution is required. Third, the combinational dimension of R.I.S.E. is huge. With these ill-behaved properties, the author was in vain in the search of an "elegant" algebraic solution method. He comes to the conclusion that

clever enumerative techniques would be the practical, feasible approach*:

In 1964, Nemhauser [1964] published a dynamic programming approach to decompose block diagonal linear programs. Instead of having to solve each subprogram iteratively for more than once as in the original method of Dantzig and Wolfe, subprograms are solved sequentially, one at a time in each stage of the dynamic programming algorithm. Each subprogram is solved only once by parametric programming. The linking constraint of the block diagonal subprograms is handled by an artificial "state" vector, which coordinates the parametric subprograms in the various "stages" of dynamic programming.

If we look back on the master program of R.I.S.E. in Figure 3-3-2 we recall that each of the rows of 1's along the diagonal corresponds to an O-D subprogram. Drawing an analogy from Nemhauser, we put forth a dynamic programming approach to solve the series of O-D subprograms. The problem context of R.I.S.E. has an inherent sequential property. We tend to think of non-stop routes first, one-stop routes second, two-stop routes third and so on. In dynamic programming terms, we have the non-stop stage, the one-stop stage, two-stop stage Within each stage, we look at each O-D pair. Each O-D pair therefore constitutes a "sub-stage".** Each O-D substage is solved for the non-stop stage, the one-stop

*This point of view is shared by Balinski [1965].

**The term substage is coined here simply for convenience. It does not belong to the set of standard terminology in dynamic programming.

stage, two-stop stage

The linking constraint of the R.I.S.E. master tableau is the set covering matrix, which ensures that each city pair be "covered" by an m -stop or better routing. On the row dimension of the set covering matrix are city pairs, while on the column dimension we find routes/routings for each O-D pair. An artificial state vector, with each entry corresponding to a city pair, will be introduced in our dynamic program (D.P.) to play the role of the linking constraint—i.e., to coordinate the routes/routings of the various O-D subprograms.

Unlike Nemhauser's problem, the "network effects" in the R.I.S.E. optimization model necessitate each O-D subprogram to be solved iteratively for more than once. When two successive solutions to R.I.S.E. are identical, the optimal solution is obtained. We call this class of problem dynamic program with an unbounded horizon. The method of successive approximation is used to converge towards the final answer.

The plan of this chapter is as follows. Section 4.2 describes how the arc-chain formulation of the master tableau from last chapter is represented as a state-stage diagram, and how route/routing generation is carried out in the diagram. Section 4.3 discusses how each route/routing is evaluated in terms of the cost/revenue computations. Section 4.4 then re-examines the selection criteria for entering a route/routing in the state-stage diagram. Section 4.5 summarizes the whole D.P. algorithm via an example. Finally, Section 4.6 throws an overview on top of the total primal decomposition approach from a theoretical framework.

4.2 Route/Routing Synthesis

It will be shown in this section that a state-stage diagram of dynamic programming (D.P.) can be readily defined for the master program of the arc-chain formulation of R.I.S.E. Further, the readers will see how the "contiguity matrix" way of route/routing generation can be conveniently represented in the state-stage diagram. The diagram facilitates the synthesis of route/routing candidates for the subsequent evaluation stage of the solution algorithm.

4.2.1 A State-Stage Space Diagram

Let us describe the state-stage space of our D.P. The states are the city pairs 1-2, 1-3, ..., (N-1)-N. The stages correspond to non-stop, one-stop, two-stop, ..., which we denote as $m = 0, 1, 2, \dots, M$. The artificial state vector $\underline{b}^m = (b_{12}^m, b_{13}^m, \dots, b_{N-1,N}^m)$ denotes the various city pair states for stage m . Graphically, we show in Fig. 4-2-1 an example of such a state-stage for a three-city network with only the non-stop and one-stop stages.

A careful comparison of the grid space of the state-stage diagram and the structure of the decomposition master tableau (in Fig. 3-3-7) would reveal certain similarities. They both have city pairs on the vertical dimension and non-stop/one-stop/two-stop/etc. route/routings on the horizontal dimension. Notice also that the set covering way of specifying that a city pair must be covered by an m -stop or better routing can be readily handled by the state vector \underline{b}^m . If b_{pq}^m is assigned a value of "1"

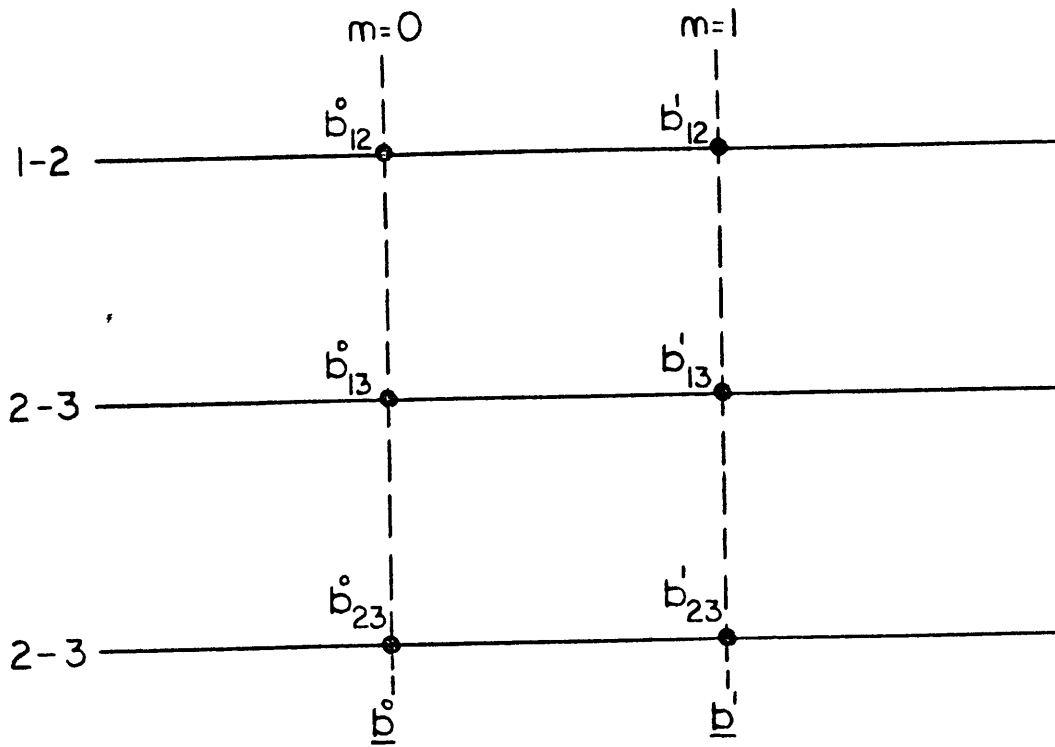


FIG. 4-2-1 AN EXAMPLE STATE-STAGE DIAGRAM

when city pair p-q is covered by an m-stop routing and a value of "0" otherwise, the requirement that city pair p-q be covered by a m_0 -stop or better routing can be written as

$$\sum_{m=0}^{m_0} b_{pq}^m \geq 1 \quad (4:2:1)$$

The state-stage diagram therefore serves as a compact representation of the master tableau. The equivalence between the state-stage diagram and the master program will be even more obvious as the discussions proceed in the next few sections.

4.2.2 Synthesizing Routes/Routings in the Diagram

The graph-theoretic way of synthesizing routes/routings (Section 3.1) can be readily represented in the state-stage diagram. Take the three-city example where cities 1 and 2 form one subsegment and cities 2 and 3 form the other. According to the C.A.B. route authority as explained in Section 3.1.2, non-stops can be scheduled between 1 and 2, and between 2 and 3. To represent these authorized non-stops, let us put a circle around the corresponding grid points in Figure 4-2-2. To serve city pair 1-3, an intermediate stop must be made at city 2 in compliance with the route authority. The one-stop route serving 1-3 is shown in Figure 4-2-2 by a circle and two arcs, which denote that the one-stop route 1-2-3 is built up from segments 1-2 and 2-3.

To extend the route building concept, two-stops, three stops can be successively synthesized in the diagram. Take Figure 4-2-3, the two-stop route 1-2-3-4 is shown to be built upon segments 1-2, 2-3 and

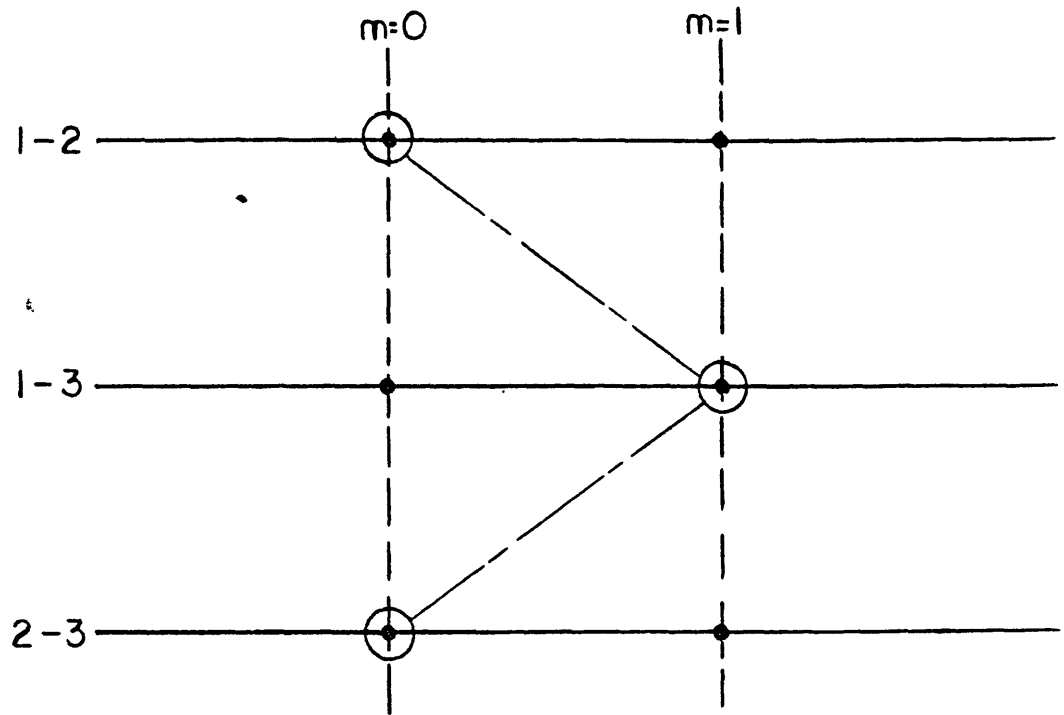


FIG. 4-2-2 ROUTE/ROUTINGS IN THE STATE STAGE DIAGRAM

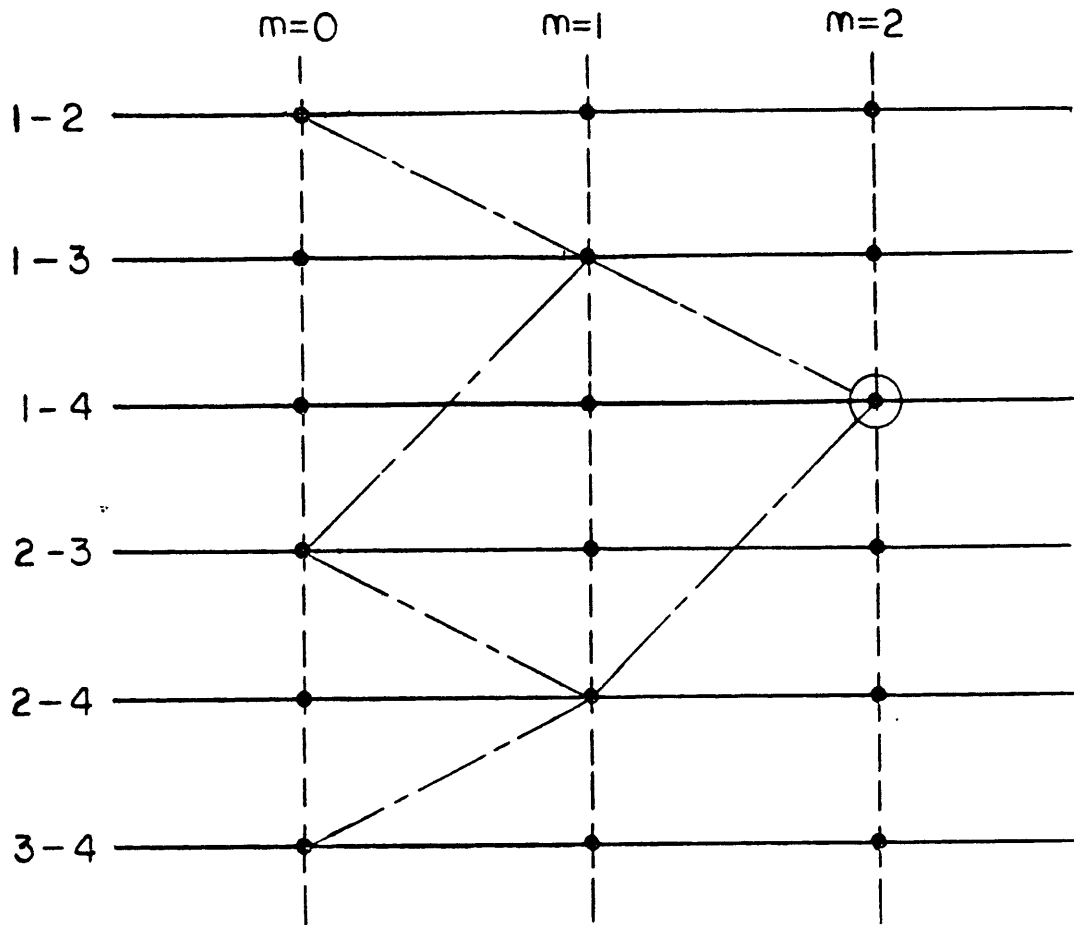


FIG. 4-2-3 A 2-STOP ROUTE IN THE STATE-STAGE
DIAGRAM

3-4. At the same time, all the through routings in the route are graphically shown in the diagram: one-stop routings 1-2-3/2-3-4, and non-stop routings 1-2/2-3/3-4.

A connect routing made up of two or more routes can also be drawn in the state-stage diagram. Take the example of the two-stop connect routing 1-2-3-4, made up of the one-stop route 1-2-3 and a non-stop route 3-4, with city 3 being the transfer station. Its graphic representation is shown in Figure 4-2-4. As another example, let us suppose that the connect routing 1-2-3-4 is made up of route 1-2-3 and the non-stop segment 3-4 of the one-stop route 2-3-4. Such a connect routing is displayed in Figure 4-2-5.

Through the use of the state-stage diagram, we have conveniently represented the dimensionalities of the master program in a compact form. The state-stage diagram also shows graphically the way a multi-stop route is built up from segments, and the through routings contained within the route.

4.3 Route/Routing Evaluation

In the last section, the reader has seen how routes/routings can be synthesized and be included in state-stage diagrams, similar to the way a route/routing column can be inserted into the decomposition master tableau. It will be seen in this section how each route/routing candidate can be evaluated in terms of its primal cost and revenue following the method of cost/revenue computation given in Section 3.3.2 of the last chapter. Furthermore, a route/routing candidate will be

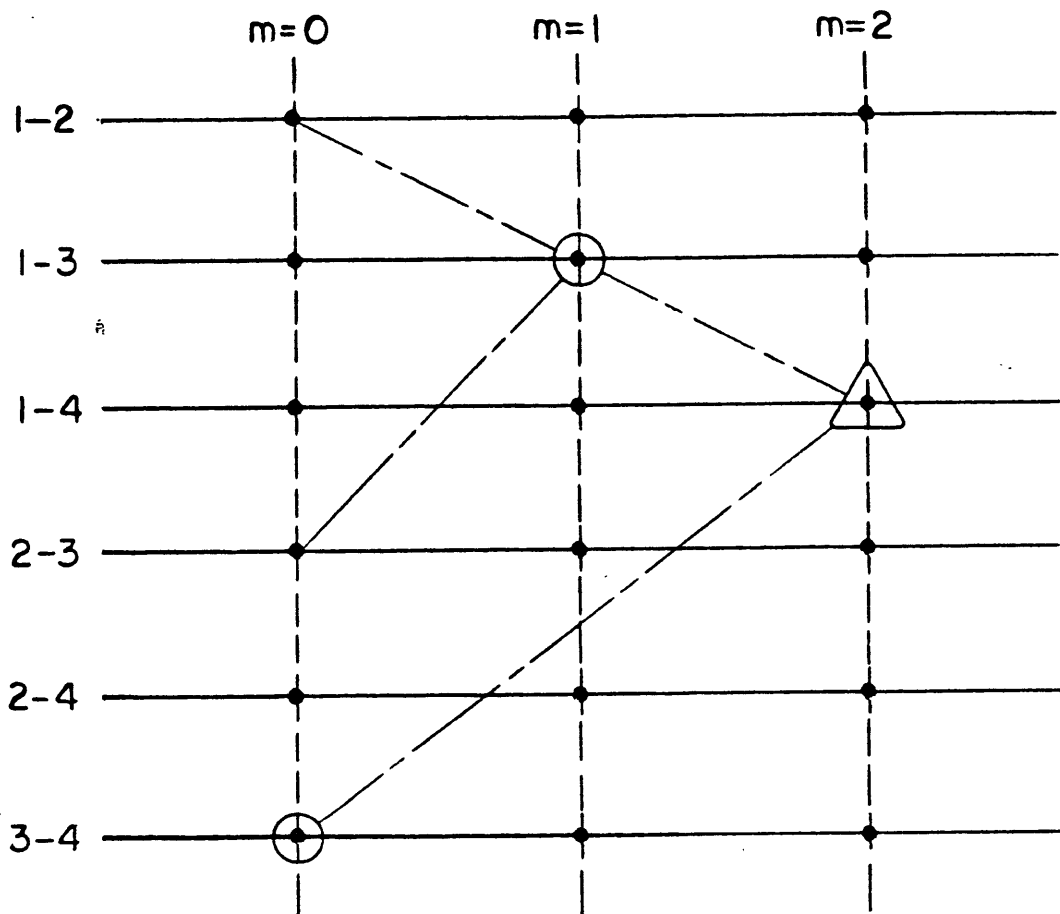


FIG. 4-2-4 A 2-STOP CONNECT ROUTING

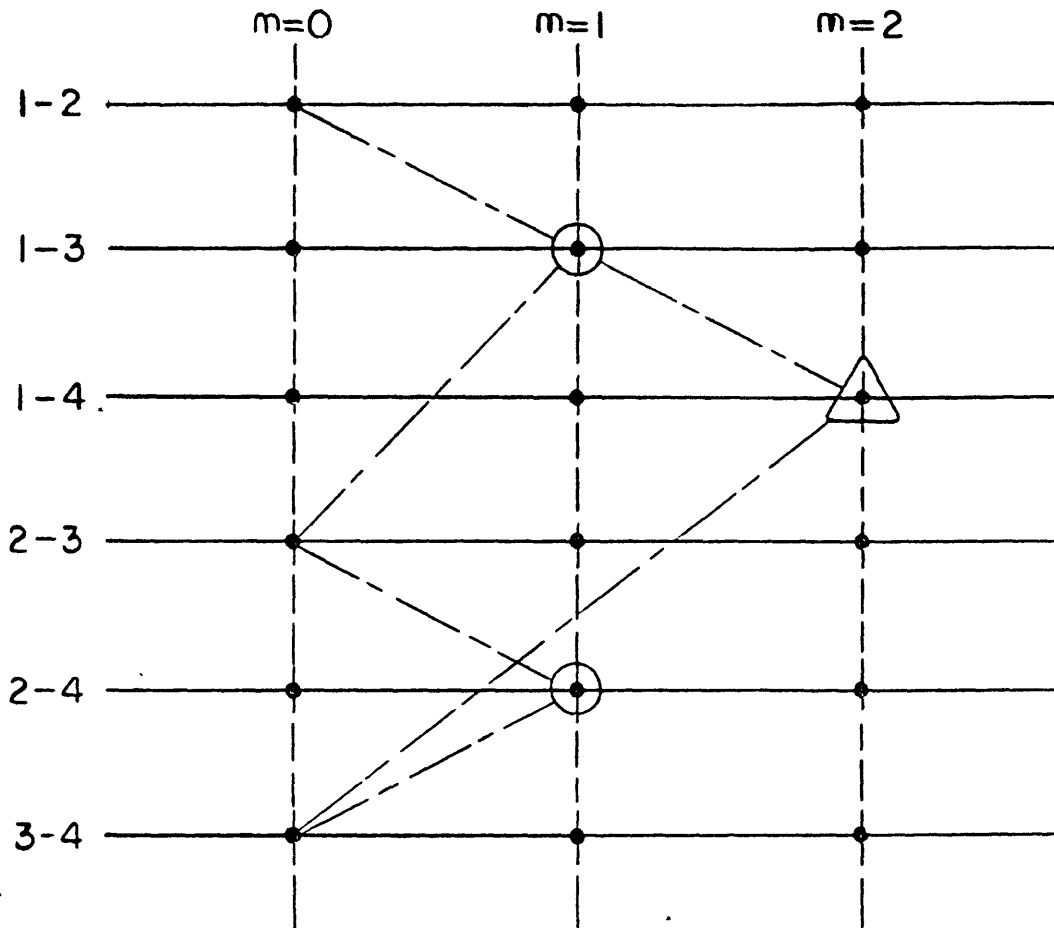


FIG. 4-2-5 A CONNECT ROUTING SYNTHESIZED FROM A ROUTE AND A ROUTING

evaluated to ensure that it contributes to provide the required m-stop or better routing between certain city pairs.

4.3.1 Passenger Flow Evaluation

It has been pointed out in Chapters 1, 2, and 3 that for each structural change in the route network topology, there is a corresponding change in the passenger flow pattern and the type of connectivity between city pairs. We have named the flow and connectivity characteristics the quantitative attributes in a network (to distinguish them from the structural or topological network features). In this section, we will see how the passenger flow information can be recorded in the state-stage diagram.

Let us overlay the passenger demand function on the state-stage diagram in Fig. 4-3-1. The demand function shown represents only the demand corresponding to through service. Additional sets of arrows should be drawn if connect routings are considered. For the sake of clarity, we will ignore connect routings for the time being.

To demonstrate how the traffic flow pattern is represented in the state-stage diagram, consider the example shown in Fig. 4-3-2. Say in the stagewise D.P. algorithm we instituted a non-stop route 1-2 in stage $m = 0$. The non-stop passenger demand between 1 and 2 would then be "discharged" onto the non-stop route. The depletion of the non-stop demand from 1 to 2 is shown graphically by shading the corresponding demand arrow. Suppose at the $m=1$ stage, a one-stop route 1-2-3 is introduced. The state-stage diagram conveniently shows that p_{13}^1 can be discharged onto the route (shown by a different type of shading in the p_{13}^1 arrow).

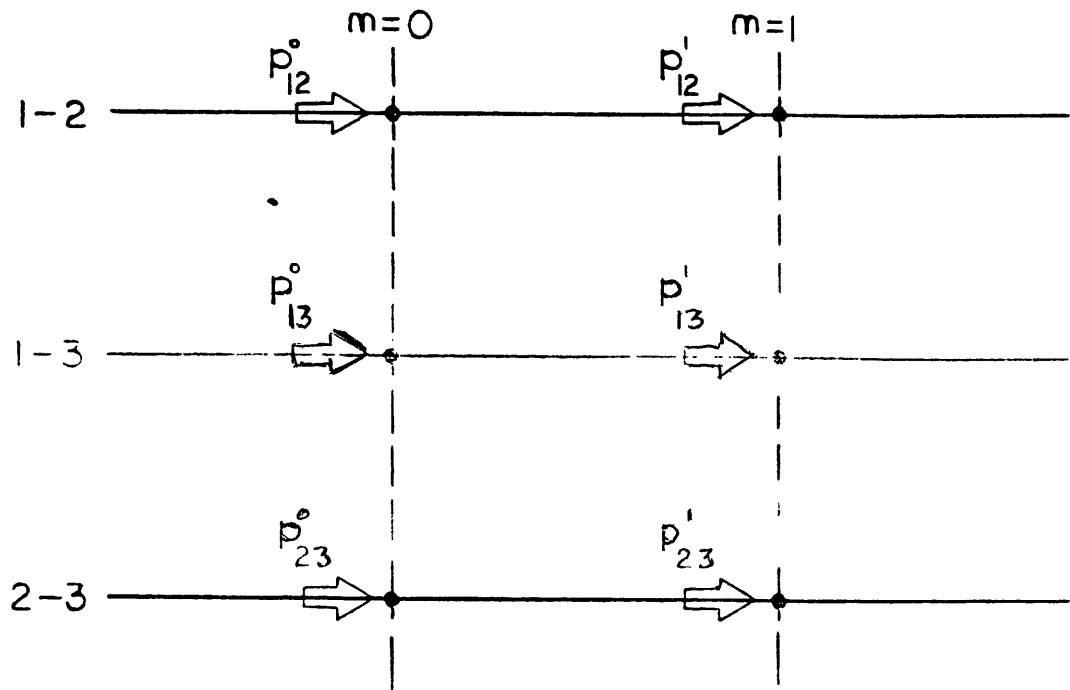


FIG. 4-3-1 PASSENGER DEMAND FUNCTION AS REPRESENTED IN THE STATE-STAGE DIAGRAM

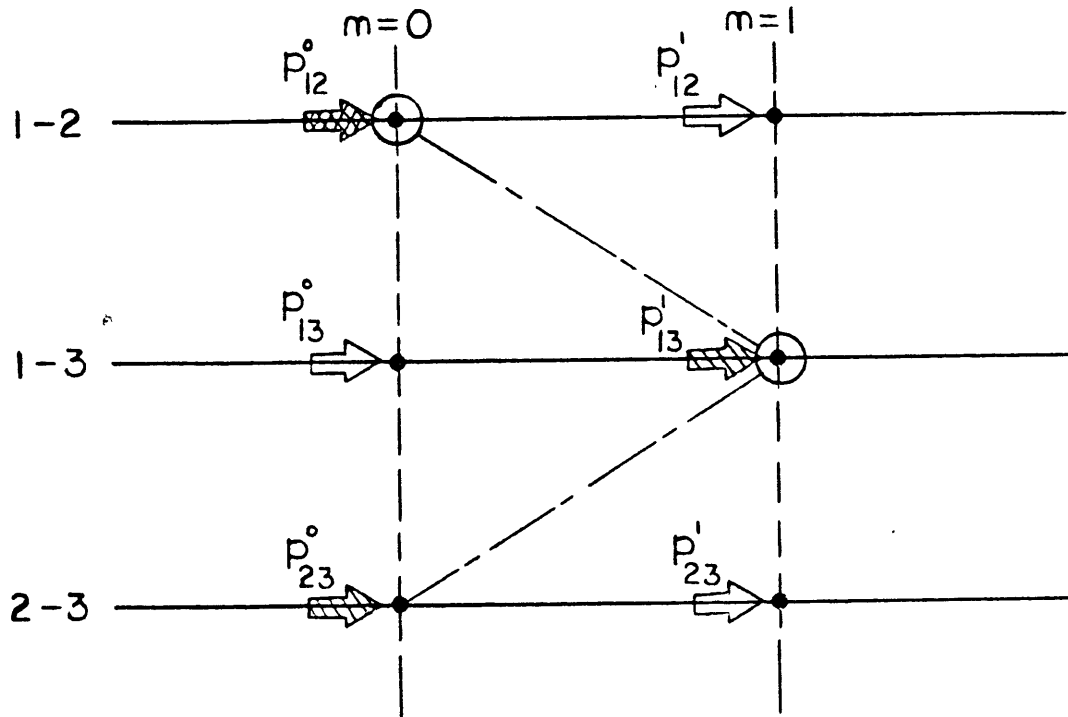
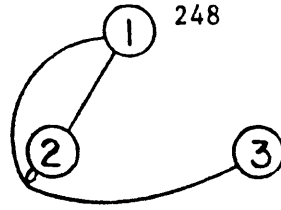


FIG. 4-3-2 TRAFFIC FLOW PATTERN AS REPRESENTED IN THE STATE-STAGE DIAGRAM

Furthermore, the non-stop demands p_{12}^0 and p_{23}^0 can be carried. City pair 1-2 is now redundantly served by both the former non-stop route 1-2 and a segment of the one-stop route 1-2-3 (the p_{12}^0 arrow is shown shaded twice). A certain portion of the 1-2 non-stop traffic could have been diverted to segment 1-2 of the route 1-2-3. The frequency on the former non-stop 1-2 can conceivably be decreased.

The traffic flow is distributed among the different routes according to the descriptive (or user optimizing) criterion. With the demand function as defined in Section 3.2.3, only the shortest routes/routings need to be synthesized to serve the O-D demands, since passengers are assumed to follow only the shortest routing.

cost

Cost for a route/connect-routing is computed in exactly the way as in Section 3.3.3. Briefly, route frequency is first computed to provide enough seats for the segment with the heaviest traffic load. The direct operating cost (including financial cost) is calculated by taking the product of route frequency m_a^r ; the number of block hours in the route $m_{t_a}^r$; and the cost per block hour unit cost c_a . As an example, the route cost for the one-stop route 1-2-3 will be (refer to Figure 4-3-3)

$$c_{13}^r = c_a \cdot l_{t_a}^r \cdot \max \left\langle \frac{l_{p_{12}}^r}{f_{L a}}, \frac{l_{p_{23}}^r}{f_{L a}} \right\rangle \quad (4:3:1)$$

$$= m_{c_a}^r \cdot m_{t_a}^r$$

for aircraft type a, where $m_{c_a}^r$ is the route cost per departure.

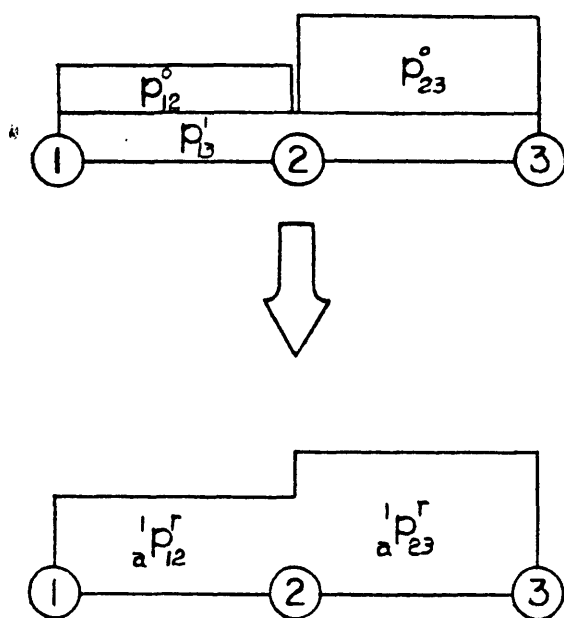


FIG. 4-3-3 THE COMPOSITION OF SEGMENT TRAFFIC
IN TERMS OF LOCAL AND THROUGH TRAFFIC

Since fleet availability is not a constraint, the least cost aircraft type with the suitable range should be chosen to fly the route in order to maximize system profit:

$${}_{13}v_r^1 = \min_a ({}_{13}v_r^1) \quad (4:3:2)$$

For the sake of illustrative clarity, we have, for the time being, ruled out the possibility of a mixed fleet to serve the same route.

In the case where a city pair is served redundantly by more than one route (such as the example in Figure 4-3-2), traffic would be distributed among the two routes so as to achieve the lowest cost. In our example, it means diverting just that much traffic from the non-stop route to the one-stop route so that the route frequencies of both routes combined would cost the least. Notice that the cost minimization is carried out with no infringement on the user travel time minimization (descriptive) criterion, since the passengers will be traveling in the shortest time paths in all cases.

revenue

Revenue from a route/connect-routing is simply the sum of all revenues for all the O-D demand carried by the route. In the state-stage diagram, the revenue computation is a straight-forward counting procedure. For example, revenue from route 1-2-3 in Figure 4-3-2 comes from the one-stop passengers p_{13}^1 , and the non-stop passengers p_{23}^0 , but only the appropriate fraction of revenue from the non-stop demand p_{12}^0 , since the remaining fraction of the revenue from those passengers

have been counted towards the system revenue as we went through the non-stop route 1-2 in the $m=0$ stage. It is to be noted that the demand function formulation used in this example assumes that passengers show no preference among aircraft types.*

4.3.2 Tabulation of Cost and Revenue

A table corresponding to the state-stage space would be used to keep account records of cost/revenue/frequency information. For example, in conjunction with the example of Figure 4-3-2 is a table shown in Figure 4-3-4. The table accounts for the basic "unit costs" and "unit revenues". For example, route 0.1 (i.e., the first of the non-stop routes) costs ${}^0c_a^1$ to fly per departure using aircraft type a. It is expected that if city pair 1-2 is served by a non-stop routing, a total revenue of ${}_{12}u^0$ will be forthcoming. These unit cost/revenue figures can be easily computed at each stage by the addition of unit cost/revenue figures from the previous stages following the arcs between the grid points. For example, the unit cost figures for route 1-2-3 in the $m=1$ stage can be obtained by (refer to Figures 4-3-4 and 4-2-2)

$${}^1c_a^2 = {}^0c_a^1 + {}^0c_a^3$$

since the block time for the one-stop route is simply the sum of the block times for the two non-stop segments. The unit revenue figure for the one-stop route 1.2 can likewise be calculated

*This assumption can be relaxed at a rather high computational cost.

	$m=0$	$m=1$
1-2	${}^{\circ}C_a^1$ ${}_{12}U^{\circ}$	${}^1C_a^1$ ${}_{12}U^1$
1-3	${}^{\circ}C_a^2$ ${}_{13}U^{\circ}$	${}^1C_a^2$ ${}_{13}U^1$
2-3	${}^{\circ}C_a^3$ ${}_{23}U^{\circ}$	${}^1C_a^3$ ${}_{23}U^1$

FIG. 4-3-4 A TABLE TO BOOKKEEP THE BASIC UNIT COSTS AND REVENUES

$${}_{13}u^1 = {}_{13}u^1 + {}_{12}u^0 + {}_{23}u^0$$

which says that the revenue to be expected from the route is the sum of the revenue from all O-D pairs served by the route. This table shows the additional advantage obtained from representing routes/routings on the state-stage diagram in the manner described in Section 4.2.2. The additivity property is certainly a big computational advantage.

4.3.3 Connectivity Evaluation

The other quantitative attributes associated with a network topology, besides traffic flow, is city pair connectivity. The state-stage diagram allows for a convenient way to check city pair connectivity. If we want to see what type of routings serve a city pair p-q, we need only to count the number of grid nodal points along the state p-q which are either circled or on which an arc is incident. For example, if we want to check the type of routings which serve city pair 2-3 in Figure 4-3-2, a quick scan along the state 2-3 reveals that the city pair is served only by a non-stop routing, since only the grid point corresponding to b_{23}^0 has an arc incident on it. Similarly, city pair 1-3 is covered by a one-stop routing since a circle is found around the grid point corresponding to b_{13}^1 .

The reader should recall that a circle around a grid point b_{pq}^m denotes that a route $m \cdot r : p \dots q$ (read: the r^{th} m-stop route beginning at p and ending at q) serves between the terminal points p-q, and b_{pq}^m assumes the value of "1".

When a grid point b_{ij}^m has an arc incident on it, it means that the city pair $i-j$ is served via a routing of a certain route, in which case b_{ij}^m also assumes a value of "1". The scanning operation along state $p-q$ merely ensures that $p-q$ is to be covered by an m_0 -stop or better routing, i.e., such that

$$\sum_{m=0}^{m_0} b_{pq}^m \geq 1$$

4.4 Route/Routing Improvement

Section 4.2 discusses how route/routing candidates are synthesized. Section 4.3 points out how a route/connect-routing is evaluated in terms of its cost/revenue potentials and the city pair connectivities it provides. The current section will marry these concepts to come up with a route network improvement scheme. The marginal profit for a route/connect-routing will be computed from the primal cost and revenue figures derived in the last section. It is based upon the marginal profit and the connectivity evaluation that a route/connect-routing is selected to enter the state-stage diagram to bring about the best incremental improvement towards the route network system.

4.4.1 Marginal Profit Computation

It has been pointed out in Section 3.3 of the last chapter that a primal decomposition method is used to tackle the route/routing improvement problem. Instead of using dual prices, primal costs/

revenues are used. A route/connect-routing is selected to enter the master tableau if it yields the highest marginal profit among the alternative route/connect-routing candidates.

From the way that the basic unit cost and revenue figures are kept (Section 4.3.2) the marginal profit for a route can be calculated in a straightforward manner. The appropriate basic unit figures, after adjustment for traffic flow distribution, yield the cost/revenue for a route. Once route cost and revenue are obtained, marginal profit is simply revenue minus cost. An example will make this clear.

Refer to the one-stop route 1-2-3 in Figure 4-2-2 and the associated table in Figure 4-3-4. Route primal cost ${}_{13}v_2^1$, is just frequency times the basic cost for a departure:

$${}_{13}v_2^1 = \frac{{}_1n_a^2}{{}_a} \quad \frac{{}_1c_a^2}{{}_a} \quad (4:4:1)$$

where route frequency $\frac{{}_1n_a^2}{{}_a}$ is the result of passenger traffic assignment on the route, as described in Section 4.3.1 (Equation 4:3:1). (Notice that aircraft type a is the least expensive aircraft to fly the route, in the way specified in Equation 4:3:2.)

Route primal revenue, on the other hand, is essentially the unit revenue figure as documented in the table in Figure 4-3-4 adjusted for the distribution of traffic among the two non-stop routings that serve city pair 1-2. We recall that city pair 1-2 is served both by a non-stop route 1-2 and a segment of the one-stop route 1-2-3. When the question is raised regarding the total revenue expected from the one-stop route,

the revenue from city pair segment 1-2 must be first resolved.

To the passengers, both the non-stop route and the segment are non-stop routings; they have no preference of one over the other according to the demand function as defined in Section 3.2.3. The passengers, in either case, are making their trips from O to D by the shortest time path, which satisfies the descriptive (or user optimizing) criterion set forth as one of the dual objectives (Section 3.2.3). The airline operator, therefore, has the freedom to assign passenger traffic among the non-stop route and the segment of the one-stop route so as to minimize operating cost--hence maximizing system profit. The least expensive alternative would determine the fraction of passengers from 1 to 2 (p_{12}^0) carried on the non-stop versus the segment of the one-stop route. It may be found that by carrying $x\%$ of the traffic on the non-stop and $y\%$ on the segment of the one-stop, it will require the least number of departures to satisfy the demand p_{12}^0 . The formal expression for computing the route primal revenue ${}_{13}u_2^1$ is given in the last chapter as Equation 3:3:10.

Route marginal revenue can now be computed as the difference between primal revenue and cost:

$${}_{13}w_2^1 = {}_{13}u_2^1 - {}_{13}v_2^1 \quad (4:4:2)$$

The best incremental improvement in the route network is achieved by selecting the best "project" among two comparable "projects" to serve the equivalent set of demands, as discussed in Section 3.3.3 (Equation 3:3:20)

For our example,

$${}_{13}w^1 = \max \left({}_{13}w_2^1, \sum_{m,r'} \Delta p'_{q,r'} w_{r'}^{m'} \right) \quad (4.4:3)$$

$$\in {}_{13}Q_2^1$$

where ${}_{13}Q_2^1$ is the set of existing routes/connect-routings which would serve an equivalent set of demands as route 1.2:1-2-3. After substituting the appropriate quantities, the above equation reduces to

$${}_{13}w^1 = \max \left({}_{13}w_2^1, \Delta {}_{12}w_1^0 \right) \quad (4.4:4)$$

where ${}_{13}Q_2^1$ contains only one element 0.1:1-2, since the only other authorized route which would serve any of the set of demands p_{13} , p_{12}^0 and p_{23}^0 is route 0.1:1-2. Recall $\Delta {}_{12}w_1^0$ is the change in marginal profit of 0.1 due to the introduction of route 1.2. Please take note that p_{pq}^m is written for each grid point of the state-stage diagram.

4.4.2 Labelling the State-Stage Diagram

The marginal profit computed from the last section will be "labelled" on the state-stage diagram during the execution of our algorithm. For example, the marginal profit ${}_{12}w_1^0$ will be labelled on the grid point b_{12}^0 as ${}_{12}w^0$, as shown in Figure 4-4-1.

In conjunction with the labelling process, a table (aside from the basic unit cost table) is used for bookkeeping. For example, in the table shown in Figure 4-4-2 are (i) the route frequency, ${}^0n^1$; (ii) the

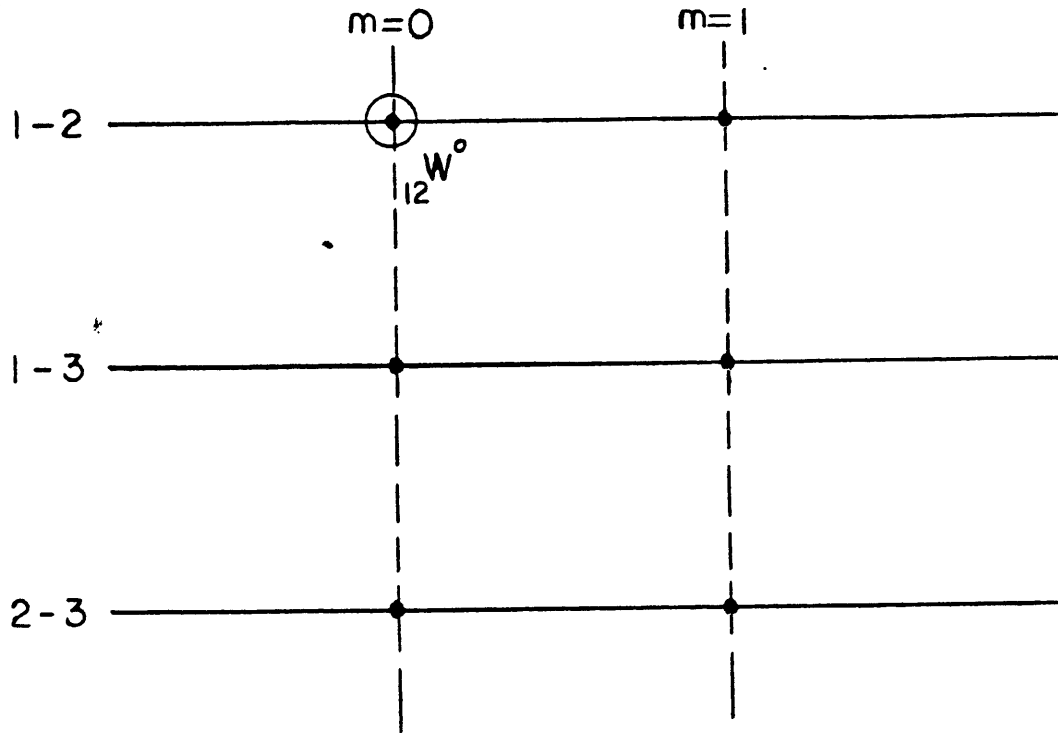


FIG. 4-4-1 LABELLING THE STATE-STAGE DIAGRAM

aircraft types that fly the route, a ; and (iii) the primal revenue for the route, p_{qr}^m . The table is needed because the information stored is required in subsequent steps of the algorithm.

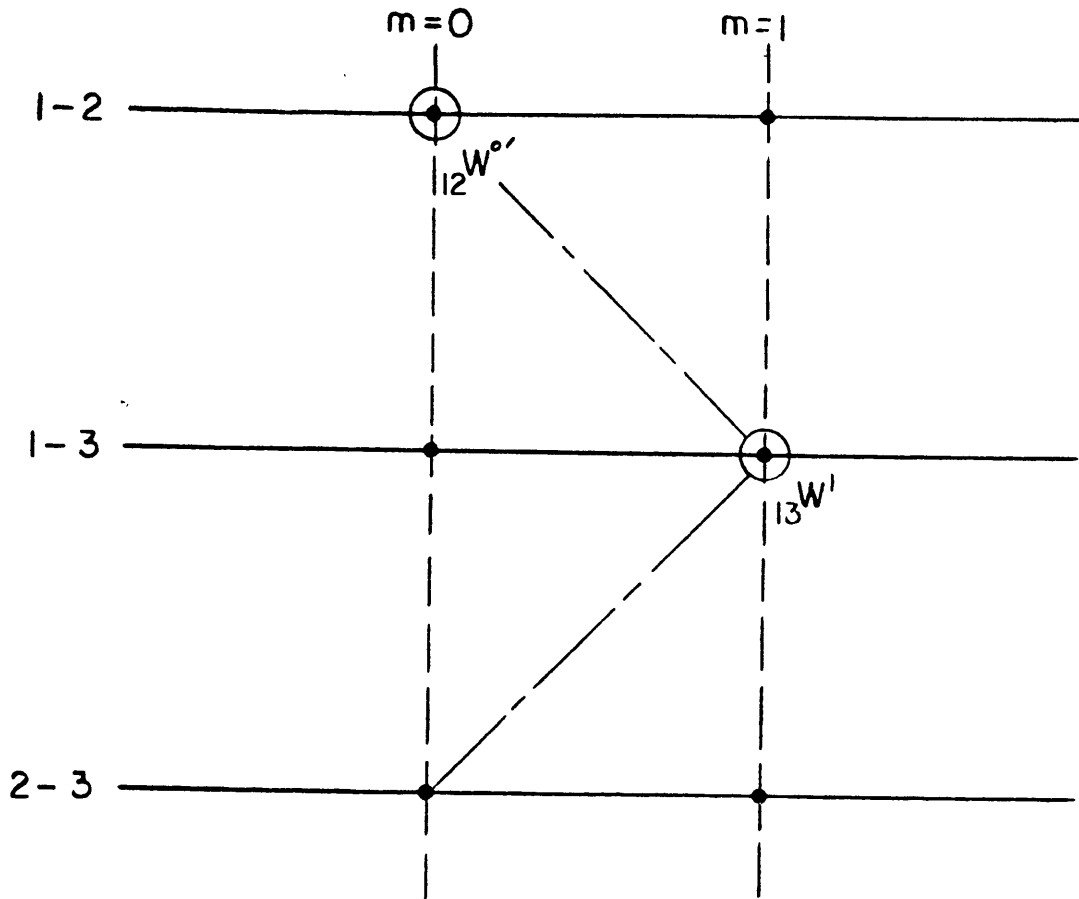
Let us bring out a few more points via the example of labelling route 1·2:1-2-3. Suppose we come to grid point b_{13}^1 in the algorithm with an existing set of labels and an existing table as shown in Figures 4-4-1 and 4-4-2. The question at hand is whether to include the one-stop route 1-2-3 in the route network. Following Equation 4:4:4, we examine the marginal profit of route 1·2 and the marginal profit of route 0·1. Notice that the marginal profits are computed based on a redistribution of p_{12}^0 between 0·1 and segment 1-2 of 1·2 which costs the least for the carrier, as explained in the last section. Very likely, this redistribution of traffic would mean a new ${}_{12}w_1^0$ (and a new label ${}_{12}w_1^{0'}$), since the number of passengers carried in 0·1 is now changed due to the introduction of 1·2. Defining $\Delta {}_{12}w_1^0$ to be $({}_{12}w_1^{0'} - {}_{12}w_1^0)$, we have Equation 4:4:4:

$${}_{13}w_2^1 = \max ({}_{13}w_2^1, \Delta {}_{12}w_1^0) \quad (4:4:5)$$

Suppose ${}_{13}w_2^1$ is the larger of the two, the one-stop route will be included in the route network. In the state-stage diagram, we label the grid point b_{13}^1 , relabel b_{12}^0 and make the appropriate entry and correction in the table for the two routes. The resulting state-stage diagram and table are shown in Figure 4-4-3. The above one-stop route example illustrates the relabelling and retabulating procedures, as well as the mechanics of making new labels and new tabulations.

	$m=0$	$m=1$
1-2	$\overset{\circ}{h}_a^1$ ${}_{12}u_1^{\circ}$	
1-3		
2-3		

FIG. 4-4-2 BOOKKEEPING A ROUTE/
CONNECT-ROUTING



	m=0	m=1
1-2	${}^0n_a'$ ${}_{12}u_a^0$	
1-3		${}^1n_a^2$ ${}_{13}u_2^1$
2-3		

FIG. 4-4-3 THE RESULT OF RE-LABELLING AND RE-TABULATING

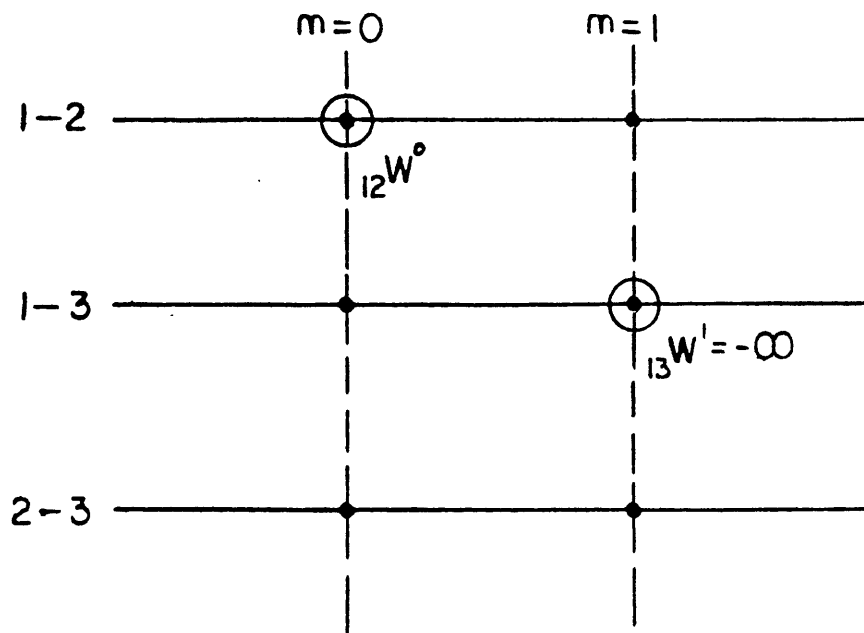
4.4.3 Phase I Versus Phase II of the Algorithm

Over the previous sections, the basic improvement operation of the algorithm has been introduced. This basic improvement operation based on Equation 3:3:20 is the crux of the whole algorithm. Before we actually go ahead with going through the total algorithm with an example, the following question should be answered: "How does the algorithm get an initial feasible solution?" The current section is devoted to providing an answer to the question.

Take the example we have been using. The first "pass" or the first "loop" through the two stages $m=0,1$ would be to obtain an initial feasible solution. The subsequent passes through $m=0,1$ would then incrementally improve on the solution, until finally the optimal route network is obtained.

Suppose we impose the constraint that city pair 1-3 must be covered by a one-stop or better routing. How could we cope with this constraint in arriving at an initial feasible solution? The method used is the commonly known "penalty method". A large negative label is put over the grid point b_{13}^1 , say ${}_{13}w^1 = -\infty$, which corresponds to a route with infinite cost. Imagine we are given a labelled state-stage space and the bookkeeping table as appeared in Figure 4-4-4 and the algorithm now comes up to grid point b_{13}^1 . The question at hand is to determine whether route 1-2 should be included into the route network. Substituting the appropriate quantities in Equation 4:4:3, we have

$${}_{13}w^1 = \max ({}_{13}w_2^1, -\infty) \quad (4:4:6)$$



	$m=0$	$m=1$
1-2	${}^0n^1$ a	
1-3		${}^0n^1 = \infty$ a_0
2-3		

FIG. 4-4-4 PENALTY METHOD TO OBTAIN AN INITIAL FEASIBLE SOLUTION

which automatically would include route 1-2 into the route map, hence satisfying the constraint that city pair 1-3 be covered by a one-stop routing or better.

The question that logically arises at this point is, "Under what circumstances should the $-\infty$ label be put at a grid point b_{pq}^m ?" The answer is: only when both of the following two conditions are true. First, the city pair $p-q$ is specified to be "covered" by an m -stop or better routing; and second, the city pair is not covered by any route or routing up to the $(m-1)^{\text{th}}$ stage. Take the example we just used-- since we specify that city pair 1-3 be covered by a one-stop or better routing, and no route or routing at the non-stop stage cover 1-3, a $-\infty$ is labelled over the grid point b_{13}^1 . To generalize, we say that a $b_{pq}^m = -\infty$ is put over grid point b_{pq}^m if both

$$(i) \quad \sum_{m_0=0}^m b_{pq}^{m_0} \geq 1$$

and (4:4:7)

$$(ii) \quad \sum_{m_0=1}^{m-1} b_{pq}^{m_0} = 0$$

The $-\infty$ label would then force the inclusion of a route/connect routing to satisfy the "city pair coverage" constraint.

We call the first pass (or first loop) around the various stages to obtain an initial feasible solution Phase I of the algorithm. The subsequent passes (or loops) to improve on the solution constitute

Phase II.

4.5 Route Improvement, Synthesis and Evaluation

The various components of the algorithm: (i) route/routing synthesis, (ii) route/routing evaluation, and (iii) route/routing improvement have been introduced in the earlier sections of this chapter. Here in this section they will be put together as a coordinated set of solution steps, which we call RISE (short for Route Improvement, Synthesis and Evaluation). The RISE algorithmic procedure will be illustrated via an example.

4.5.1 An Example

A simple three-city example network with all the essential features of the route improvement problem is given below. The reader will notice that it is essentially the example we have been using all along in this chapter.

The demand function $\{l_{pq}^m\}$ which will be used is shown in Figure 4-5-1.* For the sake of illustrative clarity, no distinction is made between through or connect service, which means that both an m-stop through routing and an m-stop connect routing induce the same passenger demand (hence l_{pq}^m is reduced to p_{pq}^m). For the same reason, demands are assumed to be additive, i.e., referring to Figure 4-5-1, the demand from 1 to 2 when both non-stop and one-stop services are offered would be the sum of 150 and 75, giving 225 passengers. Finally, please note that

*The reader should recall that l_{pq}^m is read: the passenger demand from p to q via an m-stop routing making l transfers.

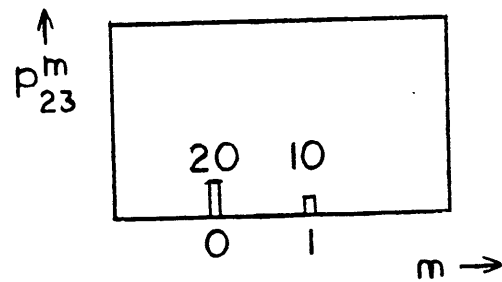
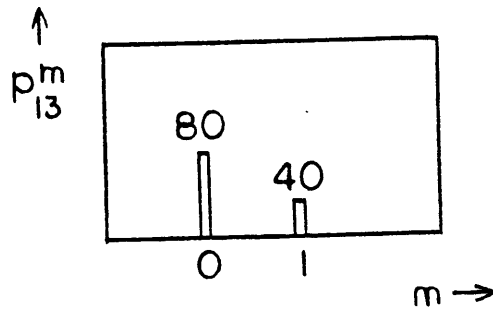
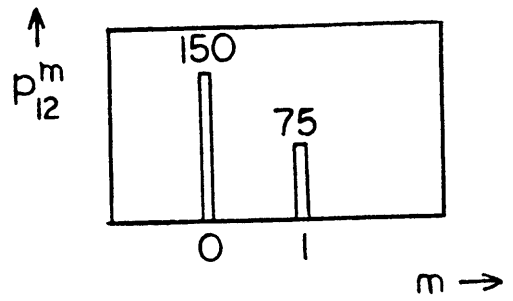


FIG. 4-5-1 PASSENGER DEMAND FUNCTION

O-D demands are symmetric--the demand p_{pq}^m is the same as p_{qp}^m .

To cut down on the length of the discussion, the given C.A.B. route authority will be pointed out as each route/connect routing is synthesized.

It is specified, from inter-carrier competition pressure, that city pair 1-2 be covered by a non-stop service, and that city pair 2-3 be covered by a one-stop or better routing. In other words, the following constraints are imposed:

$$b_{12}^0 \geq 1$$

and (4:5:1)

$$\sum_{m=0}^1 b_{23}^m \geq 1$$

There are two types of aircraft: $a = 1, 2$. The first aircraft type has a 100-seat capacity while the second has a capacity of 160 seats. An average segment load factor of 50% is assumed, which cuts down on the number of actually available seats, resulting in effective capacities of $s_1 = 50$ and $s_2 = 80$, respectively. The block times (in minutes) required to fly the direct distances between the city pairs are given below. We have neglected wind velocities and assumed a symmetric city pair block time, i.e., $t_{ij}^a = t_{ji}^a$.

$$T_1 = [t_{ij}^1] = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 0 & 60 & 80 \\ 60 & 0 & 120 \\ 80 & 120 & 0 \end{bmatrix} \end{matrix}$$

(4:5:2)

$$T_2 = [t_{ij}^2] = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 0 & 50 & 70 \\ 50 & 0 & 100 \\ 70 & 100 & 0 \end{bmatrix} \end{matrix}$$

It can be seen that aircraft type a = 2 is the faster of the two.

Only direct operating cost (with aircraft financial cost included) will be considered in the cost function. Cost per departure for a route is simply the cost per block hour times the total block hour times the total block time (in hours) of the route. Since block times are given in minutes, the cost per block minute will be specified:

$$c_1 = 7\,833 \text{ dollars/min}$$

(4:5:3)

$$c_2 = 9\,167 \text{ dollars/min}$$

It is seen that aircraft type 2 is more expensive to operate.

On the revenue side, a simplified figure of yield per passenger is given in dollars:

$$Y = [y_{pq}] = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 0 & 20 & 25 \\ 20 & 0 & 30 \\ 25 & 30 & 0 \end{bmatrix} \end{matrix} \quad (4:5:4)$$

At this point, the example problem is fully defined by the above set of given parameters. The RISE algorithm will be able to take off from here and determine the routes/connect routings to be included in the final route network.

4.5.2 The Solution Algorithm

We will now go through the solution algorithm in all its stages and substages, for both Phases I and II.

Phase I

Phase I is the first loop through the stages $m=0,1$. It will provide an initial feasible solution, subject to improvement in the subsequent loops of Phase II.

STAGE $m=0$

This stage will handle all the non-stop routes. Three substages are contained in each stage corresponding to the three states: 1-2, 1-3, and 2-3.

>substage 0.1

The algorithm starts out with an unlabelled state-stage diagram and two empty tables (one for the basic unit costs/revenues and

the other for route frequency and aircraft type bookkeeping). Noting that city pair 1-2 is required to be covered by a non-stop routing, a $-\infty$ is labelled on the grid point b_{12}^0 . Non-stop route 0.1:1-2, an authorized route, is generated/synthesized. Its inclusion into the route network is considered. This non-stop route is actually forced by the $-\infty$ penalty cost to enter the state-stage diagram. It immediately satisfies the coverage constraint on 1-2. Figure 4-5-2 documents the current state-stage diagram and the two tables. Notice, that the entire quantity of p_{12}^0 is carried by the non-stop route 0.1 (to record this, the p_{12}^0 arrow is shaded). Route 0.1 is flown twice a day by aircraft type 2, at a cost of $o_{n2}^1 \quad o_{c2}^1 = \916 and yielding a revenue of \$3000. Grid point b_{12}^0 is labelled with the route profit of \$2084, and $b_{12}^0 = 1$, as denoted by the circle around the grid point.

>substage 0.2

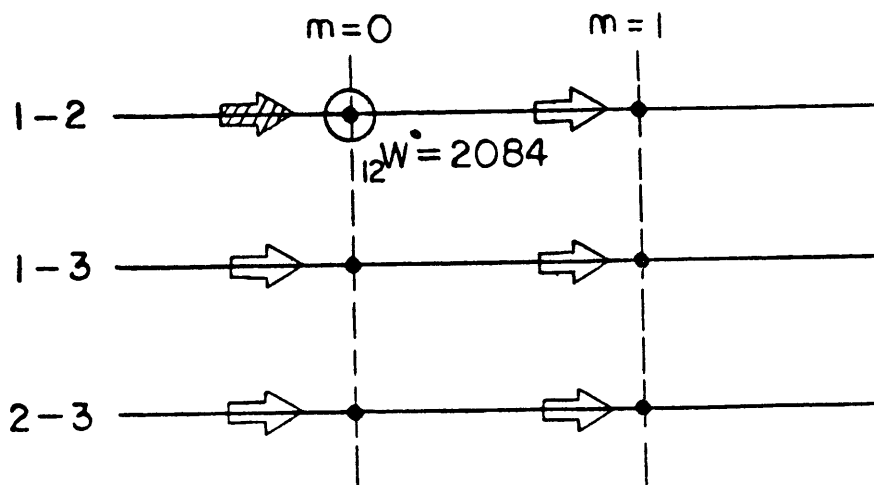
No authorized non-stops can be synthesized to fly between 1-3. The grid point b_{13}^0 remains unlabelled. But the basic unit cost/revenue associated with this grid point is entered into the unit cost/revenue table, as shown in the corresponding cell in Figure 4-5-3.

>substage 0.3

Similarly, no non-stop route is authorized to serve city pair 2-3. The basic unit costs/revenues for this grid point are recorded in the table of Figure 4-5-3. After this substage, the whole stage $m=0$ has been completed. We proceed to the next stage - - - $m=1$.

STAGE $m=1$

In this stage, all the one-stop routes will be synthesized and evaluated. Again we will go through the three substages corresponding



1-2	$\overset{\circ}{C}_1 = 470$ $\overset{\circ}{C}_2 = 458$ ${}_{12}u^{\circ} = 3000$	
1-3		
2-3		

1-2	$\overset{\circ}{n}_1 = 2$ ${}_{12}u^{\circ}_1 = 3000$	
1-3		
2-3		

FIG. 4-5-2 SUBSTAGE 0.1

	m=0	m=1
1-2	$\overset{\circ}{C}_1^1 = 470$ $\overset{\circ}{C}_2^1 = 458$ ${}_{12}U^0 = 3000$	
1-3	$\overset{\circ}{C}_1^2 = 627$ $\overset{\circ}{C}_2^2 = 940$ ${}_{13}U^0 = 2000$	
2-3	$\overset{\circ}{C}_1^3 = 940$ $\overset{\circ}{C}_2^3 = 916$ ${}_{23}U^0 = 600$	

FIG. 4-5-3 SUBSTAGES 0.2 AND 0.3

to the three states.

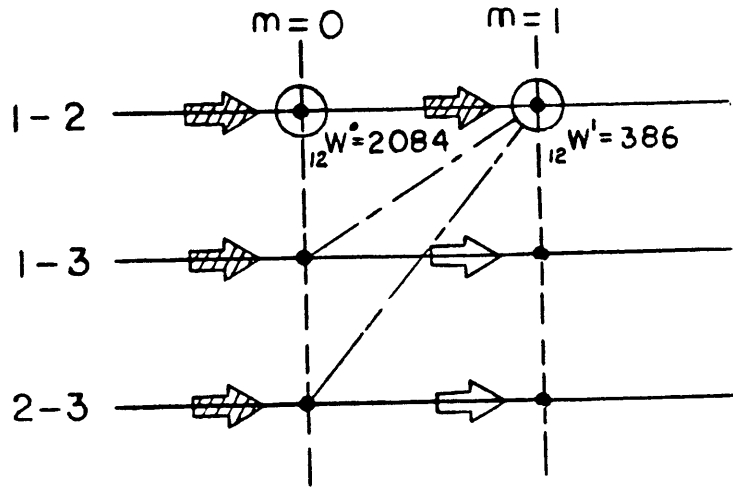
>substage 1·1

Route 1·1:1-3-2, being an authorized (as well as the shortest*) one-stop route between 1 and 2, is synthesized from segments 1-3 and 2-3. The basic unit cost for this grid point b_1^1 is computed by simple addition as explained in section 4.3.2 (see the upper table in Fig. 4-5-4). By assigning traffic, computing route frequency and substituting the appropriate cost/revenue figures in the basic improvement Equation 3:3:20 (as explained in section 4.4.1), it is decided that route 1·1:1-3-2 should be flown. We label the grid point b_{12}^1 . The associated bookkeeping procedures are documented in Fig. 4-5-4. The route network, up to this point, looks like Fig. 4-5-4a.

>substage 1·2

No authorized one-stop routes exist to serve city pair 1-3. Only a connect routing 1·2·1:1-2-3 (read: the first of the one-stop connect routing making a transfer at 2, with the actual routing being 1-2-3) can be synthesized. The connect routing uses the non-stop 0·1:1-2 and segment 2-3 of the one-stop 1·1:1-3-2. It would require only increasing the frequency of 0.1 by 1, incurring a cost of only \$458. The improvement operation equation (3;3:20) indicates adopting the connect routing project. The necessary bookkeeping is shown in Fig. 4-5-5. It is seen that in the lower table of

*Notice that in a three-node network, there is only one one-stop route/routing possible between a pair of cities, which will automatically qualify it as the shortest route/routing. For this reason, we will save using the adjective "shortest" for all subsequent one-stop routes/routings.



1-2	${}^0C_1 = 470$ ${}^0C_2 = 458$ ${}_{12}U^0 = 3000$	${}^1C_1 = 1567$ ${}^1C_2 = 1857$ ${}_{12}U^1 = 1500$
1-3	${}^0C_1^2 = 627$ ${}^0C_2^2 = 940$ ${}_{13}U^0 = 2000$	
2-3	${}^0C_1^3 = 945$ ${}^0C_2^3 = 917$ ${}_{23}U^0 = 600$	

1-2	${}^0n_1 = 2$ ${}_{12}U_1^0 = 3000$	${}^1n_1 = 2$ ${}_{12}U_1^1 = 4100$
1-3		
2-3		

FIG. 4-5-4 SUBSTAGE 1.1

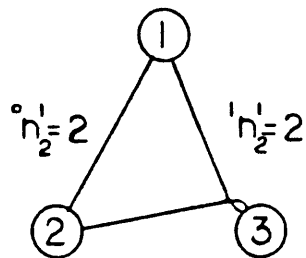


FIG. 4-5-4a ROUTE NETWORK AT THE
END OF SUBSTAGE I.I

the figure that we have increased the route frequency for 0.1 from 2 to 3.

>substage 1.3

The authorized route 1.3:2-1-3 is synthesized. However, the improvement operation equation reveals that the do-nothing alternative with zero profit is better than instituting route 1-3 (which incurs a loss). The decision is then to leave p_{23}^1 unserved. Notice no penalty cost of $-\infty$ is assigned to grid point b_{23}^1 since the requirement that city pair 2-3 be covered by a one-stop or better routing has been satisfied by the presence of the non-stop segment 2-3 of the one-stop route 1.1:1-2-3.

At this point, we have come to the conclusion of Phase I of the algorithm. It is seen that an initial feasible solution has been obtained. Furthermore, all the shortest aircraft routes have been synthesized in the state-stage diagram. The bookkeeping at the end of Phase I is shown in full in Fig. 4-5-6.

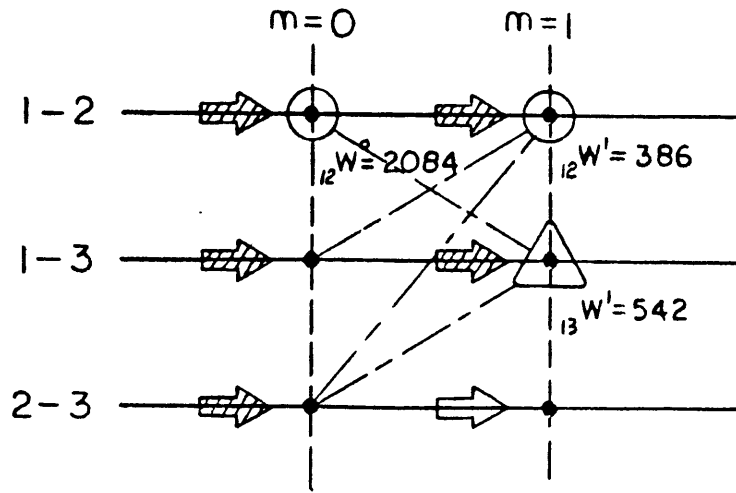
Phase II

Phase II begins with a second loop around the various stages and sub-stages. At each grid point, a perturbation method is used to examine whether there exists a comparable project to replace the accepted project which will result in a higher system profit. The perturbation examination is nothing more than the basic improvement operation expressed in Equation 3:3:20.

STAGE $m = 0$

We start out again with the non-stop routes. For each stage, the three substages are gone through step by step.

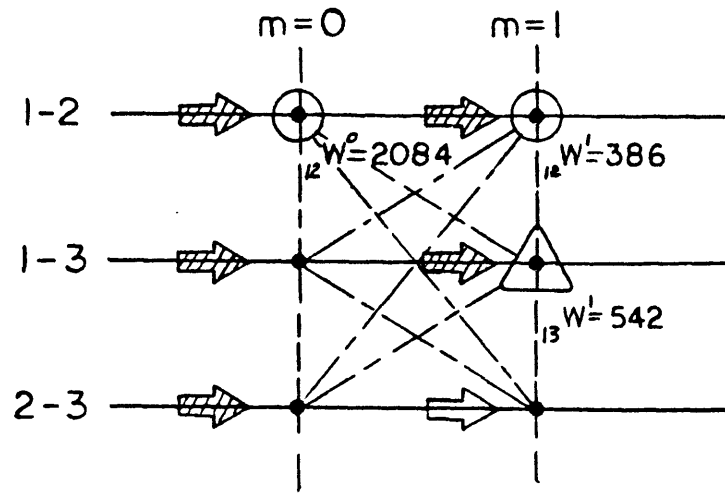
>substage 0.1



1-2	${}^0C_1 = 470$ ${}^0C_2 = 458$ ${}_{12}u^0 = 3000$	${}^1C_1 = 1567$ ${}^1C_2 = 1857$ ${}_{12}u^1 = 1500$
1-3	${}^0C_1^2 = 627$ ${}^0C_2^2 = 940$ ${}_{13}u^0 = 2000$	${}^1C_1^2 = 1415$ ${}^1C_2^2 = 1375$ ${}_{13}u^1 = 1000$
2-3	${}^0C_1^3 = 945$ ${}^0C_2^3 = 917$ ${}_{23}u^0 = 600$	

1-2	${}^0n_2^1 = 2^3$ ${}_{12}u_1^0 = 3000$	${}^1n_2^1 = 2$ ${}_{12}u_1^1 = 4100$
1-3		$\begin{cases} \Delta^0 n_2^1 = 1 \\ \Delta^1 n_2^1 = 0 \end{cases}$ ${}_{13}u_1^2 = 1000$
2-3		

FIG. 4-5-5 SUBSTAGE 1.2



	${}^0C_1 = 470$	${}^1C_1 = 1567$
1-2	${}^0C_2 = 458$	${}^1C_2 = 1857$
	${}_{12}u^0 = 3000$	${}_{12}u^1 = 1500$
	${}^0C_1 = 627$	${}^1C_1 = 1415$
1-3	${}^0C_2 = 940$	${}^1C_2 = 1375$
	${}_{13}u^0 = 2000$	${}_{13}u^1 = 1000$
	${}^0C_1 = 945$	${}^1C_1 = 1097$
2-3	${}^0C_2 = 917$	${}^1C_2 = 1398$
	${}_{23}u^0 = 600$	${}_{23}u^1 = 300$

1-2	${}^0n_2 = 3$	${}^1n_2 = 2$
	${}_{12}u^0 = 3000$	${}_{12}u^1 = 4100$
1-3		$\begin{cases} \Delta^0 n_2 = 1 \\ \Delta^1 n_2 = 0 \end{cases}$ ${}_{13}u^1 = 1000$
2-3		

FIG. 4-5-6 SUBSTAGE 1.3

Once again, we question the merit of including route 0·1:1-2 in the route network: "Can a comparable project serve as a replacement for 0·1, giving rise to a higher system at no expense of rerouting passengers via longer routings?"

The comparable route which can carry p_{12}^0 can be readily picked out from the state-stage diagram in Fig. 4-5-6. Route 1·3:2-1-3 clearly can cover city pair 1-2 on one of its non-stop segments. The alternative of using route 1·3 as a replacement for 0·1 would incur a cost of $1 \cdot 3 \cdot 1 \cdot 3 = 2 \times \frac{1 \cdot 3}{n_2} \cdot \frac{1 \cdot 3}{c_2} = 2 \times 1398 = \2796 , and would yield a revenue of ${}_{23}u^1 + {}_{12}u^0 = 300 + 3000 = \3300 . The marginal profit is therefore ${}_{12}w_3^1 = 3300 - 2796 = \504 . The basic improvement operation Equation 3:3:20 shows: $\max({}_{12}w_1^0, {}_{23}w_3^1) = \max(2084, 504) = \2084 , which says that the existing project of 0·1 is superior. The decision is therefore to retain the present label ${}_{13}w_1^0$ and the associated bookkeeping for grid point b_1^0 .

Notice that in the improvement operation computation, we have made a short-cut in the arithmetic. We have left out considering whether segment 1-2 of the connect routing 1-2-3 should be provided by 0·1:1-2 or 1·3:2-1-3. Due to the author's familiarity with the algorithm, he knows off-hand that using 1·3:2-1-3 to provide the segment would cost more, and as a result the arithmetic can be bypassed. He will share his insight on this regard in the next section, entitled "Some Computational Aspects."

>substage 0·2

No re-examination is necessary at this grid point since there exists no route 0·2:1-3 (and hence ${}_{13}w_2^0$ is zero by definition).

> substage 0.3

No reexamination is necessary for the same reason as above.

This is the end of stage m=0. We proceed to the next stage.

STAGE m=1

The perturbation method for route network improvement is carried out in exactly the same way in this stage, as will be seen.

>substage 1.1

At grid point b_1^1 , let us reexamine the merit of route 1.1:1-3-2. A comparable set of routes which will serve the demands p_{13}^0 , p_{23}^0 and p_{12}^1 carried by 1.1 would consist of only route 1.3:2-1-3, which can merely carry p_{13}^0 . The basic improvement operation Equation 3:3:20 then compares between ${}_{12}w_1^1$ and ${}_{23}w_3^1 = \max ({}_{12}w_1^1, {}_{23}w_3^1) = \max (386, 216) = \386 . The decision is therefore to keep 1 1 1-3-2.

>substage 1.2

Under the given route authority restrictions, there is no alternative way of carrying p_{13}^1 other than via the connect routing 1-2-3. Without any improvement computation, we decide to keep the present connect routing.

>substage 1.3

The question arises, just as for substage 1.3 of Phase I, whether the route 1.3:2-1-3 should be included in the route network. Since the state-stage diagram is labelled exactly the same way it appeared when we faced the same decision before, the cost/revenue computations are identical to what they were. The decision is: reject route 1-3 and leave p_{23}^1 unserved.

CONCLUDING COMMENTS

It is seen that the labelling in two consecutive loops around the stages remain exactly the same. According to the method of successive approximation, an optimal solution has been obtained. We graph the resulting route network in Figure 4-5-4a (with $\alpha_{n_2}^1$ increased to 3). The system profit is the sum of the labels, which equals \$3101. The other statistics such as costs, revenue, passenger flow, the passenger demand actually carried, route frequencies, aircraft type assignments and fleet size requirements can be deciphered from the final version of the bookkeeping table and the state-stage diagram.

4.5.3 Some Computational Aspects

In 3.3, R.I.S.E. was "manipulated" into a decomposable form in the spirit of Dantzig-Wolfe/Tomlin. In this chapter an enumerative algorithm of the dynamic programming type has been given to solve the problem. In this section, we will examine (i) computer storage requirement and (ii) computation speed of the algorithm.

We have previously counted the number of rows and columns in the decomposable formulation. Given the assumptions such as symmetry, degree of connectivity, etc., the master program takes roughly $3/4 N^2$ rows, the revenue equations takes $7/8 N^2$ and the cost equations, $1.575 N^2$. For an 80-city system, 4800, 5600 and 10,080 rows are required respectively, giving a total of 20,480. There are $.8125 N^2$ columns in the master program. The revenue computation adds $2.5 N^2$ more columns

while the cost computation adds another $.825 N^2$. For an 80-city system, this means 5200, 16,000 and 5280 columns respectively, giving a total of 26,480.

The tableau size of 20,480 by 26,480 is actually not meaningful. There are two reasons. First, we are mainly dealing with the 4800 by 5200 master tableau. The subprograms are solved one at a time. It is never necessary to solve the entire master program. Only several rows and columns of the master program need to be enumerated at a time. Second, we are not dealing with a simplex type algorithm, where the number of rows determines the size of the linear program basis. The dimension of the tableau does, however, give a rough idea about the huge size of the problem. It would be formidable to try to solve a problem of this size without decomposition. Also, particular attention must be paid to design a computed data structure capable of storing the 542 million entries of information in this tableau (of which a large percentage are zero entries, fortunately).

Viewed as a dynamic program, the amount of data storage required to be in core is only a small subset of the tableau. Simpson[1969] gives some guidelines on the data storage requirements of a dynamic programming algorithm. The high speed memory requirement is quoted to be at least twice the state space, while the low speed memory requirement amounts to the dimension of the state-stage space. Let us give a rough estimate in the state-stage space of the R.I.S.E. problem on this basis. If we assume that (i) for a city pair only the shortest routes are considered, (ii) a symmetric route network, and (iii) only half of the city pairs connected by through service, the state space is theoretically $N^3/4$ in

dimension. For an 80-city system, the high speed memory requirement is $2 \times N^{3/4} = 256K$ storage locations*. Assuming that two-stops are the longest route generated, there will be three stages in the D.P. algorithm. The low speed memory requirement is $3 \times N^{3/4} = 384K$ storage locations. Add them together, the total speed memory requirement of 630K fits well within the 1500K core of the IBM 370/65. By a careful data structure design, we have avoided using secondary storage, during the execution of the algorithm, thus saving the time necessary to bring the data into core from disk/tape/data cell, as well as the total secondary storage requirement. The actual storage requirement and computational speed will be documented in Chapter 5.

As a rule of thumb, the execution time of a dynamic programming computation grows exponentially as the number of decision variables and only linearly with the number of stage subproblems, i.e., the "substages" in our terminology. In our context, there are $N^2/2$ substages for each non-stop/one-stop/two-stop stage. We would speculate, therefore, that the computational speed grows roughly as $N^2/2$, although tree pruning techniques such as branch-and-bound (to be discussed immediately below) will speed up the algorithm significantly. Some computational experiences will be shown in the next chapter.

We recall that the route/connect-routing selection procedure is based on the primal revenue/cost/profit figures. To derive these

*In our problem, additional arrays are necessary to record the C.A.B. route authority, route frequency, etc.

figures involves post-optimality procedures on passenger flow and other computations. To evaluate each of these route candidates by reassigning passengers for each route network configuration would be rather time consuming, especially if we are repeating for all city pairs and again for non-stop, one-stop and two-stop routes. A short cut is obtained by way of a branch-and-bound routine. By watching out for monotone functions and upper/lower bounds, a good deal of computation can be saved through the "exclusion" and "rejection" rules. The branch-and-bound procedure allows us to bypass a large part of the computational burdensome task of reassigning traffic and other optimization steps.

Rather than go into the cumbersome details of the branch-and-bound rules, let us just outline the basis upon which it works. A close examination of the cost and revenue primal price expressions in Section 3.3.3 reveals some interesting facts, which we assert below without elaboration:

- system revenue is monotonically increasing as more routes are generated
- a conservative figure for route cost can be obtained by neglecting the second term of Equations 3:3:12 and 3:3:13
- a conservative cost figure for revenue passenger mile can be obtained by counting only the RPM for the new traffic induced by the introduction of the new route.

These assertions are the basis of the branch-and-bound refinement to the D.P. procedure.

Let us make some concluding remarks on the RISE algorithm and its storage and computational requirements. We can think of the RISE dynamic program as a tree construction routine. The branching logic is the route/routing generation procedure, in which legal routes/routings are generated by raising the contiguity matrix to its powers. A route is evaluated as soon as it is generated, by reassigning passengers on the route network. An unacceptable route/routing can be rejected "early in the game" and the corresponding branch of the tree pruned so that no further branching occurs from that node. This, in a nutshell, is how generation and selection interplay and how they are carried out simultaneously, rather than treated as two separate, disjointed procedures.

A dynamic programming algorithm can be coded inefficiently. Our claim that D.P. can be used to solve R.I.S.E. is based on several characteristics of R.I.S.E. which work in our favor. Storage requirements are not prohibitive because: (i) we are dealing with a macro-problem expressed in terms of a route network, rather than detailed models expressed in a "schedule map" in which the additional dimension of time is present; (ii) three stages corresponding to non-stop/one-stop/two-stops are quite sufficient to model the U.S. domestic trunk line network. Only very seldom do we need to define a fourth stage corresponding to three-stop routes, which are negligible for most purposes; (iii) the C.A.B. route authorities are geographically symmetric; and (iv) the sequential way in which the routes are generated readily lend themselves to a space saving LIST data structure bookkeeping scheme. Computation time is acceptable because: (i) a branch-and-bound procedure is used to

speed up to regular recursion equation of D.P. in the route selection step; (ii) when reassignment of passengers is necessary in the evaluation step, a set of efficient post-optimality rules modelled after that of Murchland [1969] and Chan [1969] is used; (iii) part of the route/routing generation scheme is essentially a minimum path algorithm for which efficient matrix solution techniques exist; and finally (iv) no execution time is wasted in accessing data from secondary storage, since all information is in core.

Let us say a word about the convergence of the algorithm. The basic improvement operation, Equation 3:3:20, is actually a clever enumerative step. The combinatorial space, though huge, is nevertheless finite. The number of enumerative improvement steps will therefore be finite also.

4.6 Theoretical Generalization

About a year ago, Geoffrian [1970a],[1970b] published two papers which shed some light on the general theory of decomposition in mathematical programming. We will view the decomposition of R.I.S.E. in this framework. This section serves as a review of the whole R.I.S.E. decomposable approach in a more generalized theoretical term.

The decomposable formulation of R.I.S.E. as summarized in Figure 3-3-3 is rather involved. There are the master program, O-D subprogram, primal revenue and cost computations. Conceptually, the core of R.I.S.E. has a much simpler form. It consists of essentially the route covering matrix of the master program, with only one type of decision variable--

the set of non-stop, one-stop, and two-stop routes/routings R^0 , R^1 and R^2 . The generalized form of R.I.S.E. has the following appearance:

$$\begin{aligned} \max \quad & (r_0(R^0) \circ r_1(R^1) \circ r_2(R^2)) \\ & (c^0(R^0) + c^1(R^1) + c^2(R^2)) \geq 1 \\ & R^0 \quad \quad \quad \varepsilon^0 C \quad \quad \quad (4:6:1) \\ & \quad \quad \quad R^1 \quad \quad \quad \varepsilon^1 C \\ & \quad \quad \quad \quad \quad R^2 \quad \quad \quad \varepsilon^2 C \end{aligned}$$

where the first constraint is a linking constraint of the remaining three constraints. It corresponds to what we referred to as the route covering constraint in Chapter 3. The right hand side of this constraint is a vector of 1's, each of its entries corresponds to a city pair which is required to be covered by m -stop routing routes or better. $c^0(R^0)$, $c^1(R^1)$, $c^2(R^2)$ simply denote the route covering matrices corresponding to the set of non-stop, one-stop and two-stop route/routings. The remaining three constraints simply state that route/routings are generated within the combinatorial spaces of non-stop, one-stop and two-stops, or more specifically that the set of non-stop routes R^0 are selected from the non-stop chains 0C , R^1 from 1C , and R^2 from 2C . The objective function is to maximize profit which comes from the returns of the non-stop routes $r_0(R^0)$, from the one-stop routes $r_1(R^1)$ and two-stop routes $r_2(R^2)$.

Viewing RISE as a sequential optimization procedure, the system profit cannot be obtained by simply adding up individual returns r_0 , r_1 and r_2 . The returns r_0 , r_1 and r_2 are interdependent since the introduction of one-stop routes may affect the traffic flow on the non-stop routes and affect the income from the non-stop routes, r_0 . Take a simple example, suppose a city pair A-B is formerly covered by a non-stop route, a one-stop A-B-C is now introduced that redundantly covers the city pair A-B by the first segment of the route. Traffic will be diverted from the non-stop to the A-B segment of the one-stop. Setting all other things aside, the system income (or profit) from the non-stops, $r_0(R^0)$, is decreased due to a decision made in the set of one-stop routes R^1 . The symbol "o" is introduced to take into account this interdependent effect between R^0 , R^1 and R^2 . We call "o" the "composition" symbol. The objective function then reads: "The system income is composed of the non-stop, one-stop and two-stop incomes", instead of the more familiar, additive objective function, which says "system income is non-stop income plus one-stop income plus two-stop income".

Notice that the nested min and max operators are absent from the objective function. The nested min operator has been taken care of by generating only the shortest routes/connect-routings via the contiguity matrix method (represented as $R^i c^i C$ constraints). The nested max operator has been absorbed in our procedure of computing primal cost.

Formulation Equation 4:6:1 is essentially the master program without the rows of 1's for each O-D pair. The primal revenue and primal cost,

hence marginal profit, are imbedded in the return functions r_i 's. Instead of writing the route/routing passenger variables, we explicitly use the set of routes R^0 , R^1 and R^2 as decision variables. Passenger flow is determined implicitly when we compute the primal cost of a route. O-D pair connectivity is evaluated implicitly when we evaluate the primal revenue. Equation 4:6:1, though simple in appearance, contains all the information shown in the detailed R.I.S.E. selection formulation of Figure 3-3-3. In addition, the route generation aspect of RISE is also shown as the constraints $R^i \in C$.

Geoffrian pointed out that there are two decomposition approaches: the "primal" method or resource-directive approach and the "dual" method or price directive approach. The resource-directive method determines iteratively a series of artificial resource variables, \underline{b}^i , such that we solve iteratively each subproblem:

$$\begin{aligned} \max (r_i(R^i)) \\ c_i(R^i) > \underline{b}^i \quad i = 0,1,2 \\ R^i \in C \end{aligned}$$

The price directive method determines iteratively a series of dual prices $\underline{\lambda}_i$'s such that we solve iteratively each subproblem:

$$\begin{aligned} \max \quad r_i(R^i) + \underline{\lambda}_i^T c^i(R^i) \\ i = 0,1,2 \\ R^i \in C \end{aligned}$$

RISE uses the resource directive (primal) approach while Tomlin/Dantzig and Wolfe used the price directive (dual) approach. We prefer a primal method for two reasons. First, it enables a good initial feasible solution to be utilized. This allows RISE to be used as an improvement algorithm where we start with the existing route map and go straight to Phase II without stepping through Phase I. Second, a primal method, by definition, maintains primal feasibility throughout the execution of the algorithm. If the iterations in the algorithm are stopped prior to optimality due to computer time restrictions, a good feasible solution better than the one we start out with is still available.

The artificial variables \underline{b}^2 's used in our resource-directive method have a physical interpretation. They are vectors comprised of either 0's or 1's. $\underline{b}^0 = \begin{Bmatrix} 0 \\ 1 \\ 1 \end{Bmatrix}$ indicates that while the first city pair is not covered by a non-stop route, the second and third city pairs are. $\underline{b}^1 = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}$ indicates that the first city pair is covered by a one-stop while the second and third city pairs are not. Jointly $\underline{b}^0 + \underline{b}^1 = \begin{Bmatrix} 0 \\ 1 \\ 1 \end{Bmatrix} + \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix}$ says all city pairs are covered. The physical interpretation of \underline{b}^2 's are borne out more explicitly when we look at the complete decomposable formulation after the introduction of these artificial "resource" variables:

$$\max \{ r_0(R^0) \circ r_1(R^1) \circ r_2(R^2) \}$$

$$c^0(R^0) \geq \underline{b}^0$$

$$c^1(R^1) \geq \underline{b}^1 \quad (4:6:2)$$

$$c^2(R^2) \geq \underline{b}^2$$

$$\underline{b}^0 + \underline{b}^1 + \underline{b}^2 \geq \underline{1}$$

with the R^i 's to be selected from the chains iC 's

If we compare the above formulation with Equation 4:6:1, we would notice that in the linking constraint the $c^i(R^i)$'s are replaced by the \underline{b}^i 's. The linking constraint states that the three sets of routes R^0 , R^1 and R^2 taken jointly must cover the specified city pairs. The formulation Equation 4:6:2 explicitly brings out the fact that to cover a city pair, we can allocate the task to the non-stops, one-stop or two-stops.. In other words, a job of covering city pairs by routes/routings is to be split up among the non-stops, one-stops and two-stops in the amount \underline{b}^0 , \underline{b}^1 and \underline{b}^2 . The term "resource directive" is derived from this context of splitting up a central task (resource) among different subsystems. The word "resource" has been used historically because in most formulations the linking constraint is a resource constraint such as the budget constraint.

In order to handle route generation and selection simultaneously, a flexible tree search scheme like dynamic programming and branch-and-bound is used. Dynamic programming solves Equation 4:6:2 via the following set of recursion equations:

$$\text{non-stop: } f_1(\underline{b}^1) = \max_{R^0} r_0(\underline{b}^0, R^0)$$

$$\text{one-stop: } f_2(\underline{b}^2) = \max_{R^1} (r_1(\underline{b}^1, R^1) \circ f_1(\underline{b}^1)) \quad (4:6:3)$$

$$\text{two-stop: } f_3(\underline{b}^3) = \max_{R^2} (r_2(\underline{b}^2, R^2) \circ f_2(\underline{b}^2))$$

where \underline{b}^i 's are the artificial resource vectors and the chains ${}^i c$'s are generated by raising the contiguity matrix to its own power.

The composition symbol "o" remains with us. It says that the system income in the objective function is composed of returns from non-stops, one-stops, and two-stops in a non-additive manner. In this set of recursion equations, it is easy to see how generation and selection interact with each other. The state transformation, or branching scheme, from ${}^0 c$ to ${}^1 c$ to ${}^2 c$ (hence from \underline{b}^0 to \underline{b}^1 to \underline{b}^2) is the route generation aspect of RISE. The max (·) operators handle the selection aspect. In each stage of the D.P. algorithm, generation and selection are done simultaneously.

For reference purposes, a flow diagram of the D.P. algorithm is

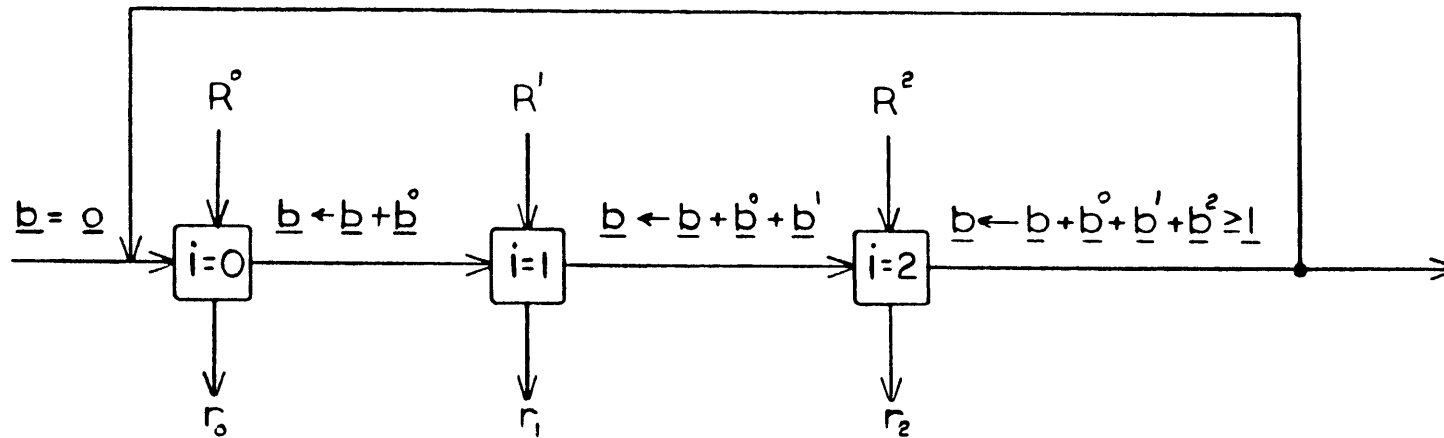


FIG. 4-6-1 FLOW DIAGRAM OF RISE

shown in Figure 4-6-1. RISE is broken up into three stages ($i = 0, 1, 2$) in the figure, corresponding to non-stops, one-stops, and two-stops. In the $i=0$ stage, non-stop route/routings R^0 yield a return (or income) of r_0 , one-stops R^1 yield r_1 etc. Coordinating the various stages is the artificial resource variable \underline{b}^i 's. The sum of \underline{b}^i 's have to ensure that certain city pairs are covered by the specified m -stop or better routing, as shown by the expression $\underline{b} + \underline{b} + \underline{b}^0 + \underline{b}^1 + \underline{b}^2 \geq \underline{1}$ in the figure. The first pass through the algorithm is Phase I, in which an initial feasible route network is configured. Subsequent passes through the stages constitute Phase II, where the route network is incrementally improved until two successive route networks remain unchanged. The optimal solution is then obtained, and the algorithm terminates in accordance with the method of successive approximation.

In this section, we hope we have reviewed from a theoretical standpoint the plan of the whole chapter. We have seen how R.I.S.E. can be conceptualized in a simple decomposable formulation like Equation 4:6:1. By introducing an artificial "resource" variable \underline{b}^i in Equation 4:6:2, we show how R.I.S.E. can be solved using a resource-directive or primal approach. Finally, we see how Equation 4:6:2 directly lends itself to the recursion equations of D.P. in Equation 4:6:3 where route generation and selection are treated as parallel rather than disjointed processes.

REFERENCES

1. Balinski, M. L. "Integer Programming: Methods, Uses, Computation," Management Science, vol. 12, no. 3 (November 1965).
2. Bellman, R. E., K. L. Cooke, and J. A. Lockett. Algorithms, Graphs and Computers. Academic Press (1970).
3. Benders, J.F. "Partitioning Procedures for Solving Mixed-Variables Programming Problems," Numerische Mathematik, vol. 4, pp. 238-252 (1963). (A more readable outline of this paper is found in Balinski's paper "Integer Programming: Methods, Uses, Computation".)
4. Chan, Y. P. "Optimal Travel Time Reduction in a Transport Network: An Application of Network Aggregation and Branch and Bound Techniques." M.I.T. Department of Civil Engineering Research Report #R69-39 (July 1969).
5. Dantzig, G. B., and P. Wolfe. "The Decomposition Algorithm for Linear Programs," Econometrica 29, pp. 767-778 (1961).
6. Geoffrion, A. M. "Elements of Large-Scale Mathematical Programming," Management Science, vol. 16, no. 11 (July 1970a).
7. Geoffrion, A. M. "Primal Resource-Directive Approaches for Optimizing Nonlinear Decomposable Systems," ORSA (May-June 1970b).
8. Jarvis, J. J. "On the Equivalence between the Node-Arc and Arc-Chain Formulations for the Multi-Commodity Maximal Flow Problem," Naval Research Logistics Quarterly, vol. 16, no. 4, pp. 525-529 (December 1969).
9. Manheim, M. L. Hierarchical Structure: A Model of Design and Planning Processes. Cambridge, Massachusetts: M.I.T. Press (1966).
10. Mesarovic, M., D. Macko and Y. Takahara. Theory of Hierarchical, Multilevel systems. Academic Press (1970).
11. Murchland, J. D. "A Fixed Matrix Method for All Shortest Distances Distances in a Directed Graph and for the Inverse Problem," London Business School Transport Network Unit Report LBS-TNT-91 (January 1969).

12. Nemhauser, G. L. "Decomposition of Linear Programs by Dynamic Programming," Naval Research Logistics Quarterly, vol. 11, nos. 2 and 3, pp. 191-196 (1964).
13. Simpson, R. W. "Scheduling and Routing Models for Airline Systems," Department of Aeronautics and Astronautics FTL Report #R68-3 (December 1969).
14. Tomlin, J. A. "Minimum-Cost Multicommodity Network Flows," Operations Research, vol. 14, no. 1 (1966).

CHAPTER 5

A CASE STUDY AND SOME COMPUTATIONAL EXPERIENCES

In the previous chapters, we have gone through the following steps in our analysis: problem identification (Chapter 2), model building (Chapter 3), and solution (Chapter 4). The present chapter will be devoted to model verification, in which the usefulness of the model will be tested on a case study from industry.

There are three parts to the chapter. First, we will point out that part of the mathematical model which has been implemented on the computer and available for running case studies. Second, we will use the software system to run a case study from American Airlines. The role that the computer model plays in analyzing "real world" problems will be discussed. Third, we will report on the computational efficiencies of the program with several test cases of different sizes.

5.1 The Status of Software Development

It was pointed out in Chapter 3 that the optimization problem addressed in this piece of research is characterized by (i) a huge combinatorial space which necessitates a repeated application of "identification" and "optimization" steps, (ii) a special shape of the objective function, and (iii) an integrality requirement. These peculiarities exclude the use of "off-the-shelf" software packages such as MPSX/370 or OPHELIE/LP for the

solution of the model. It has been a solid two man-year effort of software development by the author just to have an available solution algorithm to verify the validity and usefulness of our optimization model.

5.1.1 The Computer/Software System

Software development has been carried out in both the IBM 360/67, a time-sharing machine, and the IBM 370/155, used in a batch mode. Each individual program was debugged and tested on the time-sharing system CP/CMS with the 360/67. The debugged programs were then added one at a time to the software package and tested out in the batch processing mode with the 370/155. Currently, an operational software package -- RISE-I -- is available on the batch processing environment for production runs.

RISE-I is coded in the FORTRAN-G language. The software system consists of a total of 40 routines (not counting system programs). Programs are organized in a "modular" design so that future extension to the system can be implemented with minimal difficulty.

The object code of RISE-I takes up approximately 120 thousand bytes, including all the systems routines called and the common area. The array size required for a 25-city system is about 35 thousand bytes, and is projected to be about 353 thousand bytes for an 80-city system. The core requirement of RISE-I for most airline networks therefore falls well within the core capacity of 1500 K bytes available on the 370/155. RISE-I requires negligible secondary storage support, which is used merely to store the programs and to handle approximately $N^2/2$ (where N is the number of cities in the system) pieces of input card records. This means about 300

input records for a 25-city system and 6400 input records for an 80-city system.

5.1.2 The Implemented Part of the Model

The software package RISE-I, in its present status, has only incorporated certain parts of the optimization model as formulated in Chapter 3. The C.A.B. route regulation constraint has been implemented in full. The inter-carrier route competition constraint, however, has not been programmed. The algorithm in RISE-I only carries out Phase I of the optimization procedure, in which a good feasible route network is constructed ab initio. Phase II of the algorithm which successively improves on the initial feasible solution has not been developed, although all the program modules for the improvement step exist within the current Phase I package. It is a matter of coordinating these modules via some data links to fully implement the Phase II improvement step.

RISE-I has incorporated only a fixed (i.e., totally inelastic) set of origin-destination passenger demand functions. The author refrained from programming the full set of demand functions since there is no way he could obtain such a set of demand functions directly from American Airlines. Such a set of demand functions has to be estimated from statistical studies, which would constitute a thesis by itself.

Only the case of a symmetric passenger demand matrix has been programmed. The extension to the case of an asymmetric demand can be carried out fairly readily.

RISE-I currently can only handle a single aircraft type. The extension to multi-aircraft types would simply involve coding a couple more routines.

Presently, only direct operating cost (with aircraft financial cost) is computed. Indirect operating cost calculation has been left out thus far. This is not a drawback in any sense, since the algorithm works only on direct operating costs, as explained in section 2.2.4 of Chapter 2. Indirect operating cost is just computed ex post anyway.

There is an assumption made in the coding of RISE-I regarding the passenger flow distribution. The entire origin-destination (O-D) traffic is assigned to the first available route constructed to serve the city pair. For example, if a non-stop route is constructed first in the route map to serve city pair X-Y, all the O-D traffic between the city pair goes onto the non-stop. Even if a one-stop route/routing is constructed later on to serve the same city pair, it would not carry any traffic from X to Y, since all the O-D demand has been loaded onto the former non-stop already. Along the same line, if a one-stop route X-Y-Z is again constructed to cover city pair X-Y, the segment X-Y in X-Y-Z would still carry no traffic from X to Y, since all the O-D demand is assumed to have been loaded onto the original non-stop route X-Y. This particular traffic flow assumption was coded for computational convenience, since the author was operating under time constraints. The assumption cannot be defended rigorously. Given a reasonable amount of time, a more refined traffic assignment procedure, as developed formally in sections 3.2.3 and 4.3.1, in its mathematical rigor,

can be worked out quite straightforwardly.

5.2 A Case Study

In this section, RISE-I will be used to synthesize a route network in the American Airlines system. A comparison will be made between this synthesized route network and the existing network used by the carrier. Such a comparative study is a good way to bring to light exactly how serviceable the model is in routing analysis.

5.2.1 Definition of the Case Study

An analysis is carried out for the B707-320 fleet of American Airlines in the August peak season of 1970. The 707-320 fleet served 24 cities in the system. It was the longest range aircraft with the largest seat capacity at the time for American. The fleet mainly carries the east-west long-haul traffic.

In order to carry out an objective comparative study, we have attempted to follow the same "rules of the game" as the schedule planners in American in constructing our route network. The O-D passenger demand carried by the 707-320 fleet is supplied directly from the carrier. Similarly, they have supplied the C.A.B. route authorities, the fleet characteristics (such as seat capacity, speed and range), intercity distances and block time, revenue function, etc. However, the method of computing direct operating costs is different. The algorithm in RISE-I evaluates direct operating cost based on unit cost per block hour [Chan - 1970].* American Air-

*See also Chapter 2, section 2.2.3 for our method of costing.

lines has its own method of costing. In general, we would say the route network configuration procedure of RISE-I is based upon the same premises as that used by the carrier.

5.2.2 Analysis Results

In this section, we will discuss the similarities and differences between the route network synthesized by RISE-I and the existing version used by American.

In spite of the huge combinatorial degree of freedom that is allowed by the C.A.B. route authority, the network suggested by RISE-I shows remarkable similarity to the existing American system in general appearance. An examination of Figs. 5-2-1 and 5-2-2 will bear this out. As a whole, there are more routes in the American system (henceforth called the AA system, standing for American Airlines) than in the RISE-I network. This is due to the asymmetric route pattern of AA, while RISE-I assumes symmetry in its route network. For example, AA may serve city pair X-Y by a non-stop, but Y-X by a one-stop. Thus there are two routes serving between the cities X and Y. RISE-I, with its symmetrical algorithm, would suggest serving X-Y in the same way as Y-X. There is only one route between cities X and Y. The asymmetry of the AA practice is rather puzzling to the author, since the C.A.B. route authorities are always symmetric and the O-D passenger demand of the AA system is symmetric. The asymmetry could have been necessitated by fleet routing to meet maintenance requirements and/or the difference in eastbound vs. westbound route frequencies due to time zone changes.

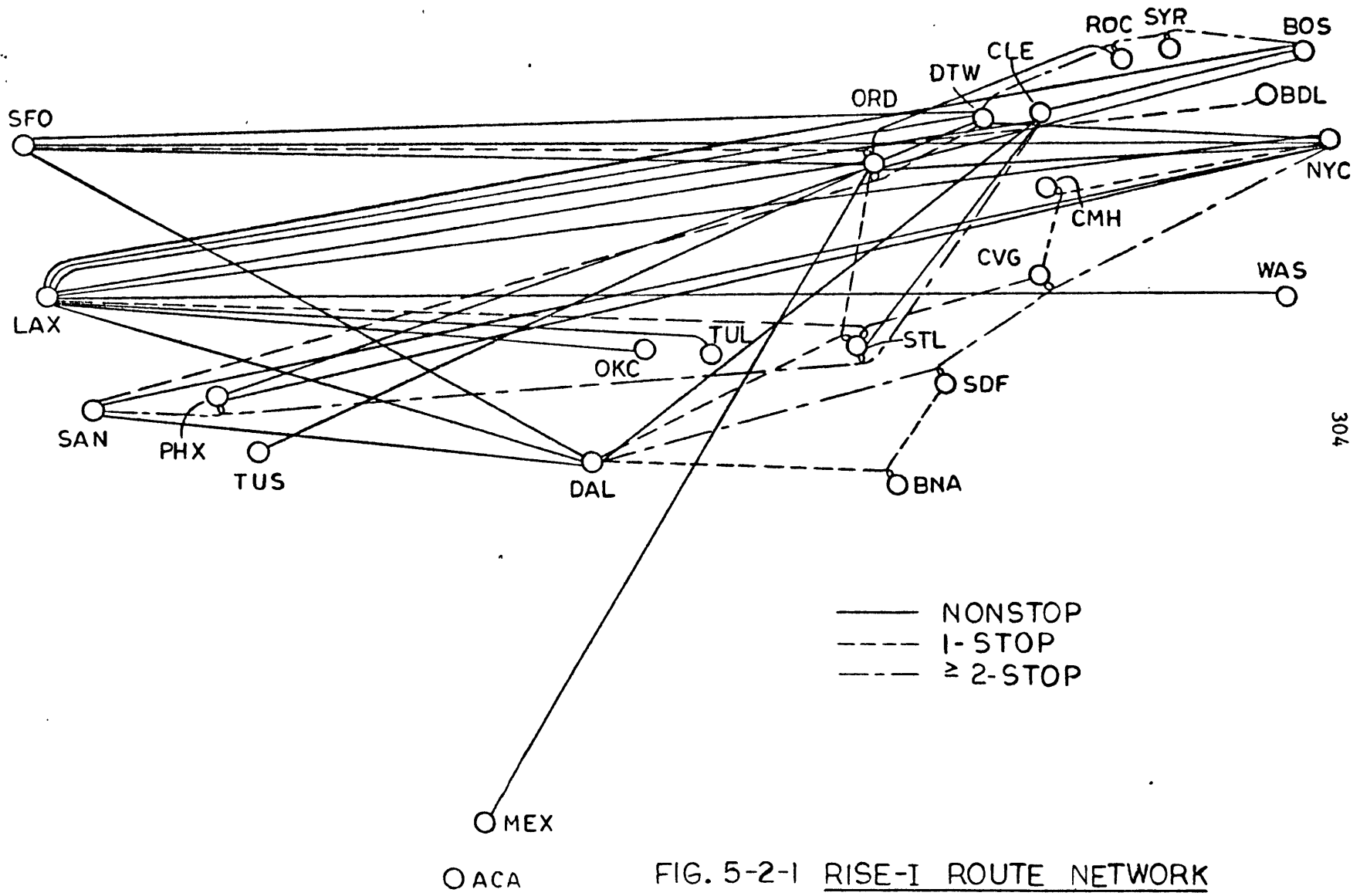


FIG. 5-2-1 RISE-I ROUTE NETWORK

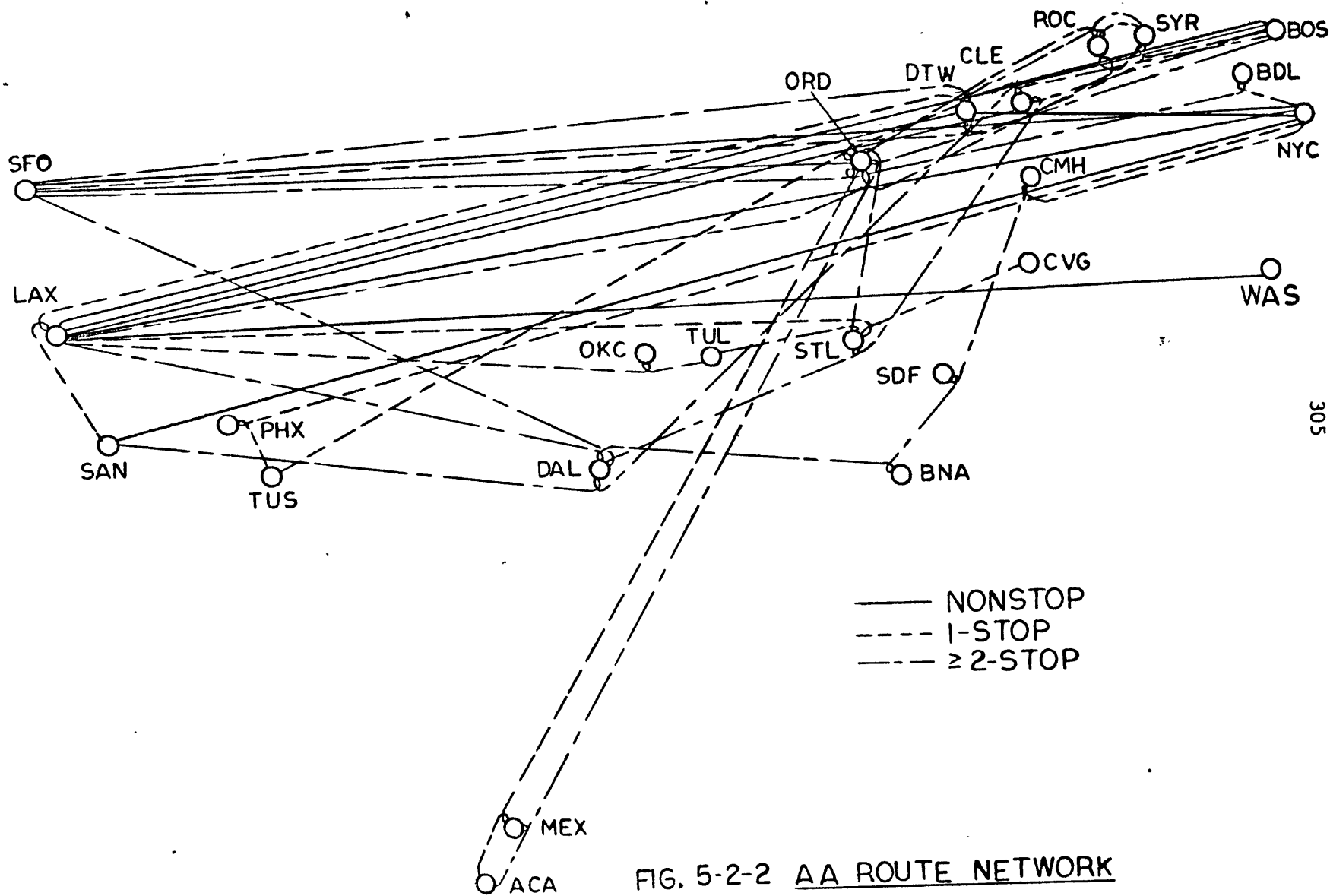


FIG. 5-2-2 AA ROUTE NETWORK

market entries and exits

Out of the 552 city pairs of the 24-city system, RISE-I provides 707-320 services between 186, or 33.7% of them. This compares with the 137 city pairs served by AA, which is 24.8% of the city pairs in the system. The comparison indicates that there are a number of profitable city pair markets which AA is allowed to serve, yet they have not gone into serving these markets by 707-320's. In a total of 46 city pairs where service is introduced by RISE-I, service neither existed in the eastbound nor the westbound direction before. We call these brand new service introductions "market entries" for the 707-320 fleet. In spite of the fact that route competition has not been incorporated into RISE-I, and the fact that we are analyzing only one of six fleet types, 12 of the city pair market entries have been substantiated by the carrier as a sound decision after considering all aircraft types and all the other factors in schedule planning.* There is an additional couple of city pair market entries, namely between CVG-SDF* and SDF-CVG, where AA has been considering instituting service since passenger demand potentials appear promising.***

While there are market entries, there are also market exits. RISE-I suggests a number of city pairs where service in both directions should be

*This refers to the analysis by the passenger flow model used in American Airlines.

**CVG is Cincinnati, Ohio, while SDF is Louisville, Kentucky.

***This service introduction was recommended by the passenger flow model used in American Airlines.

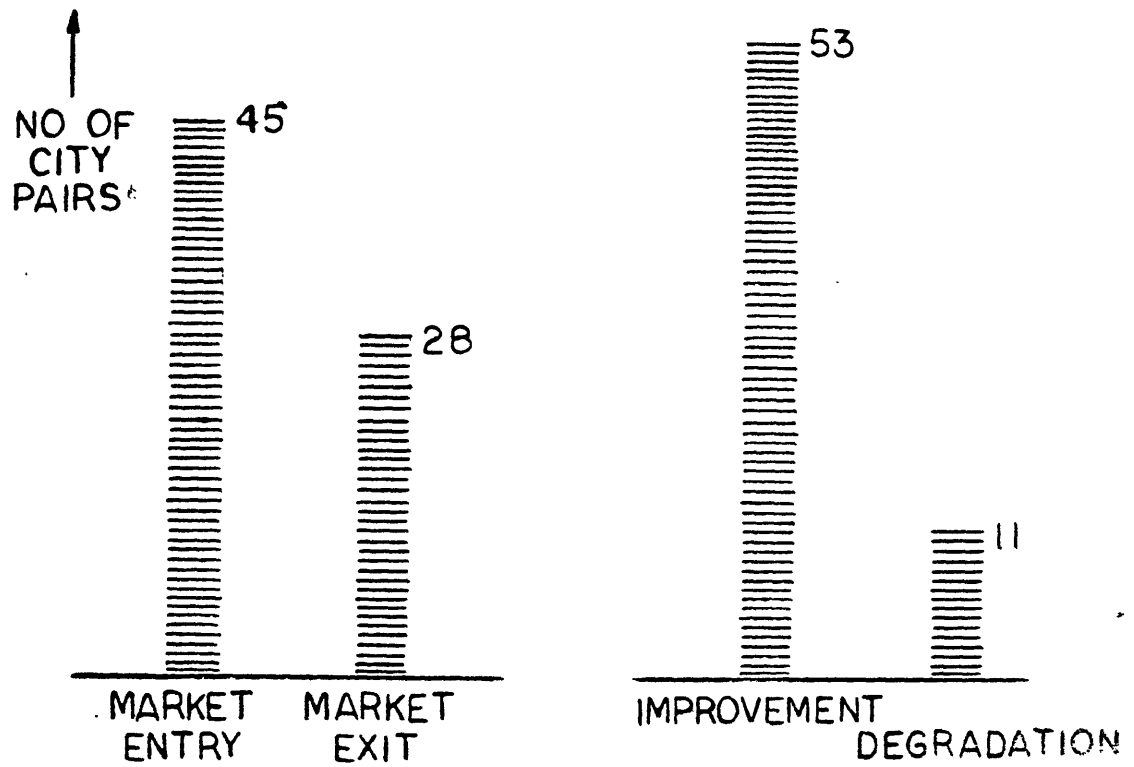


FIG. 5-2-3 DIFFERENCES BETWEEN RISE-I AND
THE A A ROUTE NETWORK

discontinued altogether. Fig. 5-2-3 indicates that it is suggested that the carrier should consider stopping service in a total of 28 city pairs by the 707-320 fleet. An example of market exit is the service to Acapulco, Mexico (see Fig. 5-2-1). Partly because of the scanty traffic in the summer and partly because of the RISE-I traffic flow logic bias as explained in section 5.1.2, the particular computer run suggested that the 707-320 service to Acapulco be scrutinized. Rigorous analysis done by the author shows that the traffic assignment logic of RISE-I has, in this particular case, biased us against serving an acceptable market.

RISE-I suggests 46 city pair market entries and 28 city pair market exits. In other words, for each market exit, RISE-I suggests 1.64 market entries. As explained above, 26.1% of the market entries are further substantiated by the carrier's own analysis.

service improvement/degradation

Fig. 5-2-4 shows a distribution plot of the number of city pairs served by non-stop, one-stop, and two-stop or longer routings. For example, there are 86 city pairs served by non-stop routings in the RISE-I route network, compared with 60 city pairs in the AA network. The distribution depicts that the RISE-I network provides more non-stop and one-stop connect routings than the AA network, while the latter provides more one-stop, more-than-two-stop routings and more-than-two-stop connect routings. Very qualitatively, we can sense that RISE-I tries to serve passengers via shorter (Improved) routings than does AA.

Referring to Fig. 5-2-3, RISE-I picks out a total of 53 city pairs for

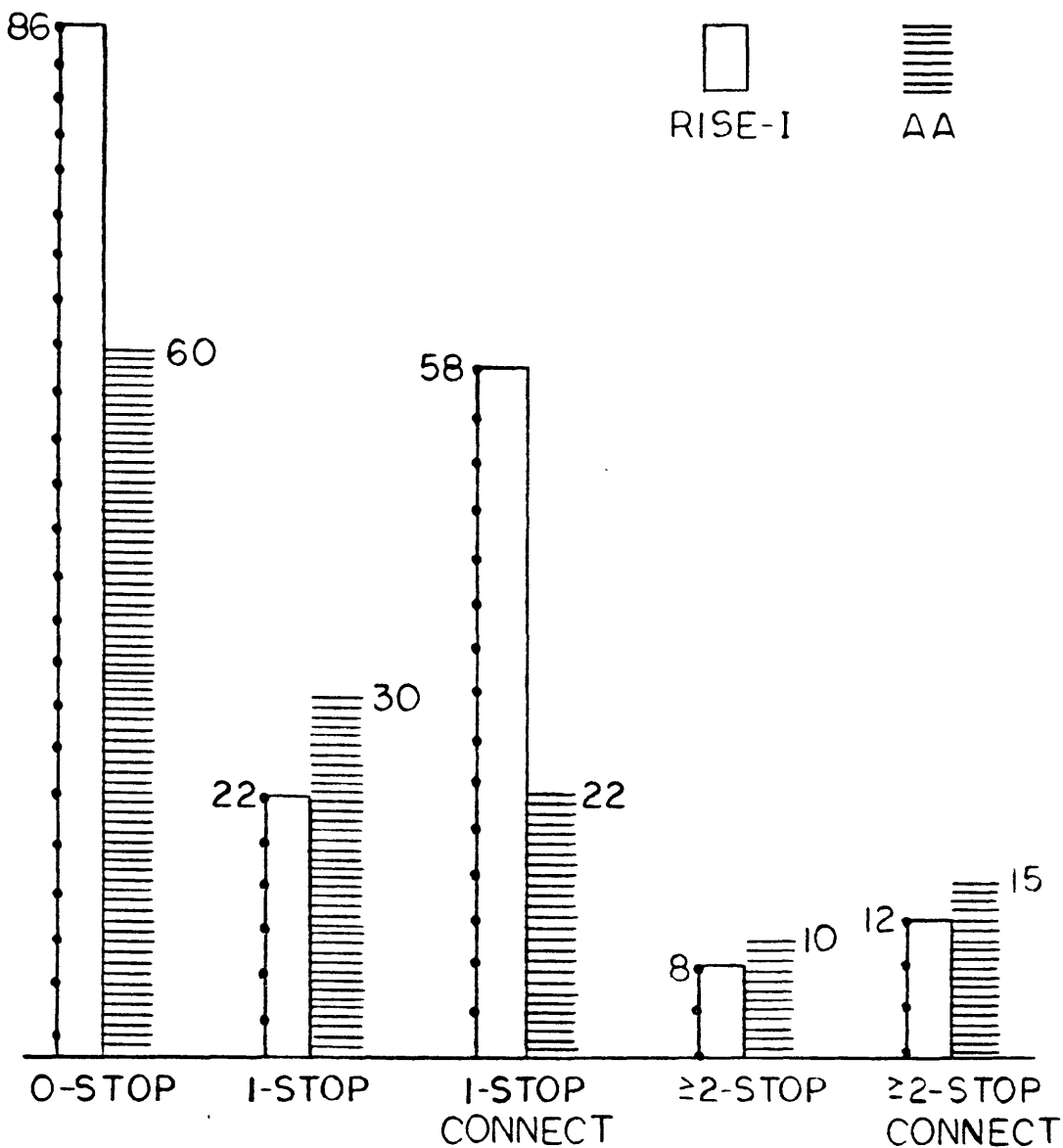


FIG. 5-2-4 A COMPARISON OF THE NO. OF CITY PAIRS SERVED BY EACH LEVEL OF SERVICE

service degradation. In other words, for every degraded city pair, there are 4.82 upgraded city pairs. Ten city pairs, or 18.9% of the city pairs with improved services, are endorsed by the carrier's own analysis.

system summary statistics

On the whole, RISE-I suggested slightly more capacity can be provided to capture a higher traffic volume in the 707-320 system. RISE-I recommends an available seat mile (ASM) figure of about 18,796,000, which accommodates about 12,166,000 revenue passenger miles (RPM). This compares with 17,579,000 ASM and 11,263,000 RPM of the existing AA system. Both RISE-I and AA achieve a comparable system load factor -- they are 64.7% and 64.1%, respectively.

With the 6.9% increase in ASM capacity, a slightly larger fleet requirement of 305 block hours per day compared with the existing AA figure of 282 block hours is needed. This represents an 8.3% increase in fleet requirement.

The expanded activities suggested by RISE-I are justified by very favorable cost and revenue figures. It is indicated that the additional traffic would increase revenue from \$555,000 to \$613,000 per day -- a 10.4% increase. Direct operating cost will be cut down from \$255,000 to \$170,000 per day without considering aircraft financing cost, and to \$243,000 a day with aircraft financing cost included.* They represent a cost reduction of 33.2% and 8.4%, respectively.

*The readers are reminded that RISE-I and AA use different cost formulae, as pointed out in section 5.2.1.

5.2.3 Conclusive Comments

The analysis so far speaks quite favorably of the serviceability of RISE-I. To be professionally objective, we would like to remind the reader that there are assumptions made in the coding of RISE-I. The assumptions have been explicitly pointed out in section 5.1.2. There is one more reservation we would like to bring out here. The route network as constructed by RISE-I does not take into account the routing of aircrafts for maintenance considerations. For example, in Fig. 5-2-2 there is an eastward route out of Tulsa, Oklahoma, in the AA route network which is absent in the RISE-I network. The eastward route out of Tulsa is actually an unprofitable route. It is used merely to ferry planes out of the maintenance center at Tulsa. RISE-I, which works on a marginal profit basis, therefore rejects such an unprofitable route out of Tulsa. The way RISE-I configures the route network implies, from the point of view of profit, that planes after maintenance work has been performed on them, should be brought back into the network system on westward flights. It is economically unsound to route them directly to the east portion of the United States.

The aircraft routing consideration can actually be considered if necessary. The routing analysis can be incorporated with RISE-I in a simultaneous manner as suggested in Chapter 1. In that light, the aircraft routing problem is by no means a shortcoming of RISE-I. And, as a matter of fact, we may want to ask: "Why choose Tulsa as a maintenance station when planes have to be ferried empty in and out of the city in order to get serviced?" Would it not be more rational, in the long run, to place

the maintenance center at a more accessible city in the network?

Let us make some conclusive remarks about the whole case study. Table 5*2*1 summarizes all the vital statistics of the comparative study between the RISE-I route network and the AA network. It was observed that although both 707-320 networks have general similarities in topology, the level of service can be improved significantly by slightly expanding the system available seat-mile. The system capacity expansion would open up a number of potentially profitable city pair markets and upgrade a large number of services. While the expanded capacity would necessitate a slightly higher fleet requirement, the decision is more than justified by the increase in revenue.

5.3 Other Computational Experiences

Besides the formal case study, a number of computer runs have been performed to assess how fast the algorithm executes. The observed computational speeds have been quite encouraging.

In Table 5*3*1 is recorded the running time for each of four test cases -- 5-city, 9-city, 16-city, and 24-city systems. They take, respectively, 4, 7, 17, and 51 system seconds on the 370/155.

For comparative purposes, we have displayed also in Table 5*3*1 a comparable software package developed by Peat, Marwick, Mitchell and Company [Jessiman, et al.- 1970]. Their reported running times for a 6-city and 25-city network are 54 and 380 system seconds, respectively, which consume at least seven times as much computation time. Considering that they are

TABLE 5*2*1
SYSTEM SUMMARY STATISTICS

	<u>AA</u>	<u>RISE-I</u>
City pairs served	137	186
RPM	11,263,000	12,166,000
ASM	17,579,000	18,796,000
System load factor	64.1%	64.7%
Fleet requirement (block hours/day)	282	305
Revenue (\$)	555,000	613,000
DOC (\$)	255,000*	234,000**

*based on a method of costing based on American Airlines data

**based on cost per block hour computation, with aircraft financial cost included

TABLE 5*3*1
APPROXIMATE PROGRAM RUNNING TIMES
(in system seconds)

RISE-I (IBM 370/155):

<u>5-city</u>	<u>9-city</u>	<u>16-city</u>	<u>24-city</u>
4	7	17	51

PMM (CDC 6600):

<u>6-city</u>	<u>25-city</u>
54	380

using a CDC-6600, which is supposedly a faster computational machine than the IBM 370/155, we have reasons to feel gratified about the algorithmic efficiency of RISE-I. Credit is attributed to two areas: (i) the primal decomposition method, in which the pruning rules of the dynamic programming/branch-and-bound algorithm are quite effective, and (ii) the computer coding has been carefully designed and cautiously programmed.

REFERENCES

1. Chan, Y. P., "Aircraft Direct Operating Cost - Related to Gross Weight," FTL memo M70-6, Flight Transportation Laboratory, Department of Aeronautics and Astronautics, M.I.T., 1970.
2. Jessiman, W. A.; Basile, J.; Ward, D. E., and D. P. Maxfield, "Inter-city Transportation Effectiveness Model," paper presented at the 37th National Meeting of O.R.S.A., April, 1970; and the related documents by Peat, Marwick, Mitchell and Company.

CHAPTER 6

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

In this dissertation, we have set out to improve the route structure of an airline network. The "real world" problem was first identified in its economic and corporate context. A mathematical optimization model was then built to represent the pertinent issues of the problem. We followed by offering a solution technique and verifying the model with a case study. What has been learned after going through this process? What bearing does our experience have on future research and development in this subject area? This chapter will attempt to answer these questions.

The plan of presentation is as follows: We will summarize our contribution in modelling, solving and verifying the route network improvement problem. The summary will provide the author with the necessary substantiations to advance his viewpoint, or his thesis, on the subject matter at hand. Based on this reasoned viewpoint, he will recommend future extensions to the work that has been performed in this dissertation.

6.1 Modelling the Route Network Improvement Problem

The airline route structure improvement problem is one out of many issues in routing and scheduling. It is a more long run problem than, for example, the real time decisions involved in schedule control -- in the sense that while an airline is unlikely to change its route structure day-in day-out, weather or mechanical malfunctions would necessitate

minute-to-minute alterations in aircraft movements. It is on the recognition of the difference between the far-reaching planning horizon vs. the more myopic decision time frame that we have proposed to model the scheduling process in a hierarchical structure, in which the problem is broken up into a number of priorly ranked issues, with route configuration being one of them.

Routing and scheduling can be classified as the "supply" side of a transportation system, since the process provides the service for people and goods movement. It is well known, however, that different levels of transportation service would induce different quantities of demand. For this reason, we have modelled our passenger demand as a function of whether the scheduled service is a nonstop, multistop, or connect routing.

U.S. domestic trunks are regulated by the C.A.B. Each carrier is restricted by the terms of Route Certificates regarding whether and how service can be provided between city pairs in the network. A graph-theoretic method using the concept of a contiguity matrix is devised to represent the route certificate quantitatively. Through straightforward manipulations of the matrix, the schedule planner automatically obtains all the different ways city pairs can be served under the C.A.B. route authority.

The eleven domestic trunks compete with each other in an oligopoly market. Route competition is one of the ways they try to differentiate their "products" (or more appropriately, service) to the travelling public. Very often, when a competitor is offering an m-stop service

between a city pair, the carrier concerned has very little choice but to match up with a comparable service. The route competition pressure exerted on a carrier has been formulated as a set covering matrix in which a city pair is specified exogenously to be "covered" by an m-stop or better routing.

In the U.S. airline industry, private corporate operators are rendering a service to the travelling public. While a private operator is concerned with profit maximization, usually an average business traveller wants to make sure he/she executes his/her trip in the minimum time path. They are two decisions makers with two different objectives--the prescriptive system optimizing operator versus the descriptive user optimizing traveller. An insightful carrier would try to align its aircraft routes with the shortest passenger routings so as to capture the highest revenue without unduely incurring extra operating costs. A dual objective function with min (·) operators nested within a max (·) operator is used to model this phenomenon.

6.2 Optimization Techniques for Problem Solution

After a mathematical optimization model, R.I.S.E., had been formulated for the route configuration problem, the research went into its problem solution phase. The progress of research went through a number of evolutions before a satisfactory technique was found. The nature of the physical problem in many cases motivated the technique.

In our constrained optimization problem, the feasible region is a huge combinatorial space of routes generated by raising the contiguity

matrix to its own power. Search is carried out over this feasible region, resulting in a selection of the best routes to be included in the route network. The selection is carried out by an integer program characterized by (i) nonconvexity, (ii) a dual objective function, and (iii) a max $\langle \cdot \rangle$ operator nested within another max (\cdot) operator in the objective function.

If route generation and selection are treated as two disjointed processes, in which the former is solved first to provide the feasible region for the latter, the dimensionality problem is formidable. The only practical approach would be to simultaneously generate a handful of routes/routings "as needed" and to simultaneously select among them. To facilitate this, the integer program is recast into a decomposable format where the most promising routes/routings are generated and immediately subject to selection. The selection criterion is based on the marginal profit of a route/connect-routing. No dual prices are used. Hence it is classified as a primal decomposition method.

The decomposable scheme has an additional computational advantage. It enables us to solve the dual objective function in a straightforward fashion. If only the minimum time routes/routings are generated to be appended to the decomposition tableau, the dual objective function reduces to a single maximum system profit objective. And we recall that the minimum time route/routing computation can be readily handled within the contiguity matrix framework. The transformation of the original optimization problem into a decomposable format therefore not only cuts down on dimensionality, but also facilitates the solution of the

dual objective function.

Still, the decomposable format is plagued with the remaining ill-behaved properties, (i) integrality, (ii) nonconvexity, and (iii) the nested max $\langle \cdot \rangle$ operator. The author feels that from a practical viewpoint, it is not cost-effective to search for an elegant, algebraic solution method. Instead, an enumerative approach is followed.

The adoption of an enumerative solution technique such as dynamic programming (D.P.) has been motivated by a number of factors. First, the master program subprogram, and column/row generation scheme of the decomposition formulation suggests to the author a D.P. state-stage space. Second, the marginal profit selection criterion of primal decomposition can readily be turned into a recursive operation of D.P. Third, the nature of the route structure problem is such that we think of non-stops first, then one-stops, two-stops, etc.--which is exactly a multistage formulation.

Dynamic programming is such a technique that nonconvexity and integrality presents no computational problem. The only remaining ill-behaved property at this point seems to be the nested max $\langle \cdot \rangle$ operator in the objective function. As will be seen, it can be accommodated in a straightforward manner in the evaluation step below.

The algorithm consists of three basic steps: synthesis, evaluation improvement. In the synthesis step, a route/connect-routing is synthesized in the state-stage diagram via the contiguity matrix method. In the evaluation step, traffic is assigned on the synthesized route/routing and its cost/revenue is computed explicitly (hence

marginal profit is available which is revenue minus cost). Because cost is calculated explicitly, the max $\langle \cdot \rangle$ presents no mathematical difficulty (as it would, if a formal algebraic type "elegant" method is used). In the improvement step, a recursive, basic improvement operation is used to incrementally improve the solution. By ensuring that the specified city pairs be covered by an m-stop or better routing in this step, primal feasibility is always guaranteed during the algorithm. In fact, the competition requirement that a city pair be served by m-stop or better routing helps in pruning the combinatorial tree.

A repeated application of the above three steps constitute the RISE algorithm (which stands for Route Improvement, Synthesis and Evaluation). The first pass through the state-stage diagram provides an initial feasible solution. Subsequent passes successively improve on the solution. This technique is called the method of successive approximation in dynamic programming. The method can be visualized as a numerical way to equilibrate supply and demand incrementally. If the algorithm is stopped before its formal completion due to limited computational resources, a feasible solution better than the one we start out with is always available.

A 40-routine software package RISE-I for the algorithm has been implemented. During program development, familiarity with the nature of the route network configuration problem leads to computational refinements using branch-and-bound, which speeds up the enumerative scheme. Because the software package is tailor-made, it

offers an inexpensive and practical problem-solving tool.

6.3 Results and Findings

After three years of research, it is rather heartening to obtain results and findings which substantiate our viewpoint, or thesis, on the subject matter at hand. The presentation in this section will consist of (i) the results of a case study from American Airlines, (ii) the computational experience of the algorithm, and (iii) a statement of the author's reasoned viewpoint (or his thesis) based on the findings cited.

RISE-I has been used to analyze the 707-320 fleet of American Airlines for the peak season of 1970. This is a private sector application of the model, since we are dealing with a profit-oriented corporation (instead of the C.A.B., which has concerns over "public convenience and necessity"). The case study indicates a slight expansion of the system capacity in terms of total available seat miles (ASM) would be desirable. The expansion of ASM would appreciably (i) upgrade the overall system level of service, (ii) open up a number of authorized city pair markets, all at a favorable system load factor, revenue and direct operating cost. It shows that within the existing route authority held by American, there are a number of superior combinatorial route structure configurations apparently not considered by the carrier at the time. An analysis of the total 53-city system of American Airlines for the summer of 1970 substantiates this point. The

analysis reveals that on the average only two out of five authorized non-stops were actually served by the carrier.

The computational efficiency of the algorithm is quite encouraging. From the computer storage requirement standpoint, our research progress has reduced the original integer program size from the order of $N^3 \times N^4$ through the decomposable formulation ($N^2 \times N^2$), to the D.P. state-stage space which measures in the order of $N^2 \times M$, where N is the number of cities in the system and M -stop being the longest route generated. It is estimated that almost all of the trunk line networks can be accommodated by RISE-I in a machine such as the IBM 370/155 (which has a 1500K core). From the execution speed point of view, a number of test cases have been run ranging from a five-city to a 24-city system. Computation time is at least seven times shorter than a comparable software package produced by Peat, Marwick, Mitchell and Company.

It is the viewpoint of the author that analyzing routing and scheduling problems on a network system framework is an insight-providing tool to transportation planning, because the huge number of combinatorial alternatives possible and the profound system/network effects are often too feeble for the unaided human mind to comprehend. Mathematical optimization techniques and computer technology have made available to analysts/planners network models which would suggest to him/her a number of worthwhile system alternatives to be considered. Some of these alternatives may prove to be more superior than the existing practice. This viewpoint has been substantiated to a certain extent by the case study cited above.

The process the author has gone through in problem identification, model building, solution and verification is extremely educational. The experience has led him to believe that problems that arise from the transportation network context without undue simplifying assumptions, are of such complexity that existing solution techniques are usually not available. This can be witnessed from the original integer program formulated for route selection. Flexible techniques, which often may not be highly mathematically elegant, promise to be the most expedient methods to solve such problems in a practical context. In many cases efficient solution methods can be motivated by the intrinsic nature of the problem under consideration. The research experience from the initial integer programming formulation, through decomposition, to the final enumerative solution techniques speak for these viewpoints. From an applications standpoint, the solution method must be well suited to the capabilities of the contemporary computer. Usable solutions to the problem at hand should be available for use even if the optimization technique fails to reach at the "optimum". These last two points are again the observations from our computational experience.

The thesis advanced above is found to be endorsed by two authorities in the field. Bellman, et. al. [Chapter 2, 1970] speaks for the use of a flexible technique suited for the computer, and motivated by the physical problem. Balinsky [1965] thinks highly of the practicality of enumerative approaches to integer programming.

6.4 Research Status and Recommended Extensions

To answer the questions of "where do we stand, and where do we go from here?", we will proceed in two parts. The first part pertains to the applications side of the model, while the second part deals with the optimization techniques.

The application of R.I.S.E. has been solely with the private sector up to this point. The case study was drawn from a corporate organization—American Airlines. The only place where public sector is modelled is in the dual objective function where the travelling public is identified as consumers maximizing their own convenience in travel. The extension to the problem of C.A.B. route application by a carrier involves only carrying sensitivity analysis with the model inputs. R.I.S.E. needs little extension before it can be useful in public sector applications by a government agency such as the C.A.B. The model already has a dual objective function (one for the carrier, the other for the travelling public). The formulation of demand as a function of the level of service lends itself handily in measuring social welfare in terms of consumer surplus or the willingness to pay. The C.A.B. can use the model to analyze granting route awards, the network routing issues in merger studies, and can even address the issue of regulation/de-regulation in the present oligopoly market using R.I.S.E. as a simulation tool. These applications have been discussed in Section 2.4 of Chapter 2.

On the optimization techniques side, a clear distinction has to be made between the mathematical model versus the part of the model that

has been coded on the computer. In spite of the two solid man-years of programming, the capability of the software package is tiny in comparison to that of the general mathematical solution algorithm presented. The existing 40-routine coding can only handle (i) a fixed, or perfectly inelastic demand, (ii) a single fleet type, and (iii) a monopoly market without intercarrier route competition. The obvious extension, as far as computer programming is concerned, is to expand the coding to catch up with the general capabilities of the mathematical model and solution method. This means additional routines would be coded to accommodate a demand function, multi-aircraft types and inter-carrier route competition. It is projected that these extensions are straightforward since the core of the mathematical algorithm, including the complete C.A.B. route authority has already been implemented.

As far as the mathematical optimization formulation and the method of successive approximation algorithm are concerned, they have been fully developed in this piece of research. Refinements in the pruning rules of the branch-and-bound type may further accelerate the already efficient algorithm. We recommend the main thrust of the optimization research be devoted to integrate R.I.S.E. with other existing routing and scheduling models in a comprehensive, multilevel optimization procedure. It would be a significant contribution if a rigorous method can be devised to break up a large scale schedule planning problem into its logical hierarchical levels and stages, in the spirit of the simul-sequential approach outlined in Chapter 1.

REFERENCES

1. Balinski, M. L. "Integer Programming: Method, Uses, Computation," Management Science, vol. 12, no. 3 (November 1965).
2. Bellman, R. E., K. L. Cooke, and J. A. Lockett. Algorithms, Graphs and Computers. Academic Press (1970).

BIOGRAPHICAL NOTE

Dr. Chan was born in Canton, China on 12 August, 1945. He received his Bachelor of Science in Civil Engineering degree from M.I.T. in 1967. He joined the Transportation Systems Planning/Operations Research curriculum in the graduate school of M.I.T. where he received the Master of Science in 1969. His doctoral degree research was performed in the Flight Transportation Laboratory, Department of Aeronautics and Astronautics.

Dr. Chan's industrial experiences include urban transportation planning with Deleuw, Cather and Company. He was part of a consulting team to the Salt Lake City Transportation Center Study by Ford Motor Company. His interest in air transportation was initiated by working with the Operations Research Group of the Sales Development Department in McDonnell Douglas, where he was involved in the Airline Schedule Planning Model. In the Corporate Planning Department of American Airlines, Mr. Chan furthered his schedule planning work. The work with McDonnell Douglas and American Airlines has evolved into his doctoral dissertation.

Dr. Chan has been performing research and teaching in the past four and one-half years at M.I.T. He has been involved in U.S. Department of Transportation research contracts and authored the following final reports:

"Optimal Travel Time Reduction in a Transport Network: An Application of Network Aggregation and Branch-and-Bound Techniques," M.I.T. Department of Civil Engineering Research Report R69-39.

"Aggregation in Transport Networks: An Application of Hierarchical Structure," co-authored with K. Follansbee, M. Manheim and J. Mumford, M.I.T. Department of Civil Engin-

earing Research Report R68-47.

His doctoral dissertation, entitled "Route Network Improvement in Air Transportation Schedule Planning," will be published as a research report from the Flight Transportation Laboratory.

Dr. Chan is a member of the Operations Research Society of America, Association for Computing Machinery, Highway Research Board, Institute of Management Science, American Society of Civil Engineers and the American Institute of Aeronautics and Astronautics. His academic honors include a fellowship from the Harvard-M.I.T. Joint Center for Urban Studies, Tau Beta Pi and Chi Epsilon memberships.

APPENDIX A1

Issued pursuant to
Order No. E-25644UNITED STATES OF AMERICA
CIVIL AERONAUTICS BOARD
WASHINGTON, D.C.-----
CERTIFICATE OF PUBLIC CONVENIENCE AND NECESSITY

(as amended)

for Route 63

WESTERN AIR LINES, INC.

is hereby authorized, subject to the provisions hereinafter set forth, the provisions of Title IV of the Federal Aviation Act of 1958, and the orders, rules, and regulations issued thereunder, to engage in air transportation with respect to persons, property, and mail, as follows:

1. Between the terminal point Los Angeles, Calif., the intermediate points San Francisco--San Jose and Oakland, Calif., and Portland, Oreg., and the terminal point Seattle, Wash.;
2. Between the terminal point San Diego, Calif., the intermediate points Palm Springs, San Bernardino, Long Beach, and Los Angeles, Calif., and Las Vegas, Nev., and (a) beyond Las Vegas Nev., the intermediate points Oakland, San Francisco--San Jose, and Sacramento, Calif., and the terminal point Reno, Nev.

The service herein authorized is subject to the following terms, conditions, and limitations.

Western 63

(1) The holder shall render service to and from each of the points named herein, except as temporary suspensions of service may be authorized by the Board; and may begin or terminate, or begin and terminate, trips at points short of terminal points.

(2) The holder may continue to serve regularly any point named herein through the airport last regularly used by the holder to serve such point prior to the effective date of this certificate; and may continue to maintain regularly scheduled nonstop service between any two points not consecutively named herein if nonstop service was regularly by the holder between such points prior to the effective date of this certificate. Upon compliance with such procedure relating thereto as may be prescribed by the Board, the holder may, in addition to the service hereinabove expressly prescribed, regularly serve a point named herein through any airport convenient thereto, and render scheduled nonstop service between any two points not consecutively named herein between which service is authorized hereby.

(3) The holder shall not deplane at Las Vegas, Nev., persons, property, or mail enplaned at Las Vegas, Nev., for the period during which Bonanza Air Lines, Inc., is authorized to provide service between Las Vegas, Nev., and Riverside-Ontario, Calif.

(4) The holder shall not engage in single-plane service between Las Vegas, Nev., and Reno, Nev.

Western 63

(5) The holder shall not deplane at San Francisco—San Jose or Oakland, Calif., persons, property, or mail enplaned at Sacramento, Calif., or deplane at Sacramento, Calif., persons, property, or mail enplaned at San Francisco--San Jose or Oakland, Calif.

(6) The holder shall not schedule single-plane service through the San Jose airport between San Francisco--San Jose, Calif., and the following points: Seattle, Wash., Portland, Oreg., Las Vegas and Reno, Nev., and Los Angeles, Long Beach, and San Diego, Calif.

The exercise of the privileges granted by this certificate shall be subject to such other reasonable terms, conditions, and limitations required by the public interest as may from time to time be prescribed by the Board.

This certificate shall be effective on September 7, 1967.

IN WITNESS WHEREOF, the Civil Aeronautics Board has caused this certificate to be executed by the Secretary of the Board, and the seal of the Board to be affixed hereto, on the 7th day of September 1967.

HAROLD R. SANDERSON

Secretary

(SEAL)