NATIONAL BUREAU OF STANDARDS REPORT

10 592

A SIMULATION MODEL FOR ESTIMATING AIRPORT TERMINAL AREA THROUGHPUTS AND DELAYS

Technical Report
to the
Federal Aviation Administration
Systems Analysis Division
Systems Research and Development Service

by
Technical Analysis Division and
Applied Mathematics Division

Report No. FAA-RD 71-9

NBS

U.S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards [1] was established by an act of Congress March 3, 1901. Today, in addition to serving as the Nation's central measurement laboratory, the Bureau is a principal focal point in the Federal Government for assuring maximum application of the physical and engineering sciences to the advancement of technology in industry and commerce. To this end the Bureau conducts research and provides central national services in four broad program areas. These are: (1) basic measurements and standards, (2) materials measurements and standards, (3) technological measurements and standards, and (4) transfer of technology.

The Bureau comprises the Institute for Basic Standards, the Institute for Materials Research, the Institute for Applied Technology, the Center for Radiation Research, the Center for Computer Sciences and Technology, and the Office for Information Programs.

THE INSTITUTE FOR BASIC STANDARDS provides the central basis within the United States of a complete and consistent system of physical measurement; coordinates that system with measurement systems of other nations; and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. The Institute consists of an Office of Measurement Services and the following technical divisions:

Applied Mathematics—Electricity—Metrology—Mechanics—Heat—Atomic and Molecular Physics—Radio Physics [2]—Radio Engineering [2]—Time and Frequency [2]—Astrophysics [2]—Cryogenics.[2]

THE INSTITUTE FOR MATERIALS RESEARCH conducts materials research leading to improved methods of measurement standards, and data on the properties of well-characterized materials needed by industry, commerce, educational institutions, and Government; develops, produces, and distributes standard reference materials; relates the physical and chemical properties of materials to their behavior and their interaction with their environments; and provides advisory and research services to other Government agencies. The Institute consists of an Office of Standard Reference Materials and the following divisions:

Analytical Chemistry—Polymers—Metallurgy—Inorganic Materials—Physical Chemistry.

THE INSTITUTE FOR APPLIED TECHNOLOGY provides technical services to promote the use of available technology and to facilitate technological innovation in industry and Government; cooperates with public and private organizations in the development of technological standards, and test methodologies; and provides advisory and research services for Federal, state, and local government agencies. The Institute consists of the following technical divisions and offices:

Engineering Standards—Weights and Measures — Invention and Innovation — Vehicle Systems Research—Product Evaluation—Building Research—Instrument Shops—Measurement Engineering—Electronic Technology—Technical Analysis.

THE CENTER FOR RADIATION RESEARCH engages in research, measurement, and application of radiation to the solution of Bureau mission problems and the problems of other agencies and institutions. The Center consists of the following divisions:

Reactor Radiation—Linac Radiation—Nuclear Radiation—Applied Radiation.

THE CENTER FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides technical services designed to aid Government agencies in the selection, acquisition, and effective use of automatic data processing equipment; and serves as the principal focus for the development of Federal standards for automatic data processing equipment, techniques, and computer languages. The Center consists of the following offices and divisions:

Information Processing Standards—Computer Information — Computer Services — Systems Development—Information Processing Technology.

THE OFFICE FOR INFORMATION PROGRAMS promotes optimum dissemination and accessibility of scientific information generated within NBS and other agencies of the Federal government; promotes the development of the National Standard Reference Data System and a system of information analysis centers dealing with the broader aspects of the National Measurement System, and provides appropriate services to ensure that the NBS staff has optimum accessibility to the scientific information of the world. The Office consists of the following organizational units:

Office of Standard Reference Data—Clearinghouse for Federal Scientific and Technical Information [3]—Office of Technical Information and Publications—Library—Office of Public Information—Office of International Relations.

[1] Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D.C. 20234.
[2] Located at Boulder, Colorado 80302.
[3] Located at 5285 Port Royal Road, Springfield, Virginia 22151.

NATIONAL BUREAU OF STANDARDS REPORT

NBS PROJECT		NBS REPORT
4314569	May 1971	10 592

A SIMULATION MODEL FOR ESTIMATING AIRPORT TERMINAL AREA THROUGHPUTS AND DELAYS

Judith Gilsinn

with

E. H. Short, W. A. Steele, D. Klavan

Technical Report
to the
Federal Aviation Administration
Systems Analysis Division
Systems Research and Development Service

by
Technical Analysis Division and
Applied Mathematics Division

Report No. FAA-RD 71-9

U.S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

TABLE OF CONTENTS

page

ABSTRACT

This report documents a simulation model (DELCAP) designed to estimate airport throughputs and aircraft delays, taking into account their dependence on (1) the traffic level and mix of user types, (2) the airport configuration, and (3) the separation rules in force. The model is implemented in two parts: a preprocessor to facilitate data entry by providing standard data input which a user may elect instead of providing his own, and an event-oriented simulation model. The report includes a discussion of the elements in the airport system which are modeled, a description of the simulation model's logic, and a set of sample outputs. Listings of the model programs, and a users' guide to their operation, appear as appendices. The report concludes with suggestions for next steps in this development effort, including validity and sensitivity analyses, model modification, and data collection.

1. INTRODUCTION

DELCAP is a fast-time computer simulation model designed to provide measures of airport throughput and user delays (under IFR operations), for a broad range of scenarios described by

(a) airport configurations and operating modes

(b) mixes of user aircraft types, and

(c) separation criteria.

It is presently conceived as mainly a planning and analysis tool, for use in answering single queries or a series of "what if?" questions concerning what effects changes in (a) and/or (b) and/or (c) would have on delays and throughputs.

The model is written in the "simulation language" SIMSCRIPT I.5, and is presently operational on the UNIVAC 1108 computer at the National Bureau of Standards (NBS). In addition to the simulation, there is a FORTRAN preprocessing program which allows the user to provide model input in an easier and more flexible manner than that required by the standard SIMSCRIPT input format.

The model has been run for a variety of airport runway configurations and is believed to be capable of handling all the configurations schematized in Table 1 of FAA's AC 150/5060-3A ("Airport Capacity Criteria Used in Long Range Planning"), under any mode of operation which has been described to us as "reasonable". (An example of an "unreasonable" mode of operation is an open V configuration operating away from the point and having landings on both runways. In this case the landing patterns would cross in the air.)

The design of a computer simulation model is generally guided (either explicitly or implicitly) by one of two opposing philosophies. The first of these, which might be termed the "include every detail" view, aims at maximum realism; its ideal is a model which closely and comprehensively mirrors all the workings of the process being simulated. It is hoped that such a model will "leave nothing out" and will thus be capable of answering almost any reasonable question which might be asked about the system under study. One consequence is that modeling and programming efforts can proceed without awaiting prior specification in operational terms of just what questions are to be posed.

On the negative side, however, such models are necessarily expensive to build, to provide with reliable data at the required level of detail, and to run. They are usually so large and complicated that only experienced computer personnel can operate them. Because of these drawbacks, a common result is that they are run only a few times (if at all). Moreover their development involves a special danger: building the "perfect model" tends to become an end in itself, the designer losing sight of why it was desired in the first place.

The *other* view, which might be called the "model as little as you can" philosophy, holds that models should be designed to answer *specific* classes of questions, and should include *only* those details of the system being modeled which directly affect the answers to these questions. Such models, therefore, usually cannot adequately answer questions other than those for which they were designed. There is also the danger, in designing such a model, of omitting some relevant detail because its relevance was not recognized in time. This risk can never be eliminated absolutely, but it can be reduced to a level acceptable in practice by careful study of the system before actual design of the simulation is begun. Indeed, proponents of the "model as little as you can" view usually spend at least as much time in studying the system as in the construction of the simulation model itself.

The advantage of this particular approach is that the model is of manageable size, and therefore easier and less expensive to operate. It can actually be *used* by planners, to answer questions of the type it was designed to answer. Because of this ease and low cost of operation, many different cases and variations can be simulated.

There is a drawback to the "model as little as you can" approach, which may well be the real reason many modelers subscribe to the "include every detail" philosophy. A model which is more "abstract", i.e. which is based on deliberate and selective abstracting from real-world complexity, is also more difficult to explain. Use of such a model lays the designer open to accusation of "why did you omit this or that detail?", and leaves him with a genuinely difficult task if the model must be explained solely in the operationally-oriented terms familiar to the potential user of the simulation. A proper description should, however, produce conviction that the model includes all those aspects of the system essential for answering the questions at which it is directed.

In this project, we have subscribed to the second, "model as little as you can", point of view. The present document is therefore designed to describe, to the prospective user of the DELCAP simulation, which elements of the air transportation system have been modeled and how they are represented. Chapter 2 will provide a description of the terminal area as seen through the eyes of the model. Chapter 3 will provide a more computer-oriented view of the model. The input data which are needed will also be discussed in this chapter. Model outputs are taken up in Chapter 4, which includes descriptions of a number of sample "runs" of the model. Chapter 5 summarizes the present form of the DELCAP effort, and lists some possible directions for future work to make DELCAP an even useful tool. This chapter also describes how the model might operate in a computer time-sharing environment as a highly accessible analysis tool A bibliography is included as Chapter 6.

There are eight appendices to this report. The first of these, Appendix A, gives input formats and operating instructions for any user who wishes to run the model. Sample input decks are also portrayed. This section is intended for readers without extensive computer backgrounds, but those with more computer experience should also find it useful. It can best be described as a "DELCAP users' manual", which includes information necessary for running the existing code.

The next four appendices are meant for the programming-oriented reader who might be interested in altering the code or in transferring it to a different computer. Appendix B gives a list of the model elements, subroutines, variables and arrays used both in the DELCAP simulation and its preprocessor. Appendices C and D give listings of the preprocessor and the simulation, respectively.

The preprocessor is written in FORTRAN and the simulation is SIMSCRIPT. Both of these languages are available on a wide range of computers. However, as anyone who has worked on several different computers is well aware, the implementation of the same language on different "machines" is apt to be quite different. Appendix E has been prepared to document those properties of the DELCAP model and its preprocessor which are known by NBS project staff to be machine dependent, and hence to require possible alteration when DELCAP is converted to another computer. Others may come to light during the conversion process. It was known during the design and development stages that the model might be converted to other machines, and this fact was taken into consideration in designing the model; hence the conversion process should be less difficult than if conversion had been an afterthought.

The next two appendices contain mathematical derivations of two formulas used in the DELCAP simulation. The final appendix contains a copy of a Mock Instruction Manual for the DELCAP model as it is envisioned to operate in its final form.

We conclude this section by listing the project staff involved in the DELCAP effort:

Operations Research Section (Applied Math Division)

J. Gilsinn (Project Leader)

D. Klavan

Mathematical Modeling Group (Technical Analysis Division)

E. Short

W. Steele

In addition to those listed above, R. Penn of the Technical Analysis Division and A. J. Goldman of the Applied Mathematics Division provided valuable criticism and guidance in the course of the model design and development.

2. FUNCTIONAL DESCRIPTION OF THE MODEL

2.1 General Description

The DELCAP simulation model is designed to provide measures of throughput and user delays for a variety of terminal-area scenarios which can be characterized by

(a) airport runway configurations and operation modes

(b) mixes of aircraft types, and

(c) separation rules.

It is intended to handle primarily IFR operations, and so the rules referred to in section (c) above may be interpreted as IFR separation rules.

DELCAP is envisioned as a planning and analysis tool for use in investigating the effects on throughput and delay of changes in any of (a), (b), and/or (c). It is not designed, for instance, to provide controller work-load information or to output gate assignments for various airlines. It is designed to aid in answering questions such as, "Can the present configuration at a given airport handle an increase of 50% in traffic?", or "Would delay be decreased by reserving one runway for large and medium size jets and large piston aircraft?". In accordance with the "model as little as you can" philosophy described in Chapter 1, DELCAP has a limited scope and limited application. It is designed to be used in planning for next year or several years hence, rather than planning day-to-day operations. It can be used to examine the effects on the level of congestion of adding extra runways, or of reducing required IFR separation. It cannot, however, answer the much more difficult question of how safe a current or proposed procedure is. It is envisioned as only one part of the planning process, a tool to be used by the planner rather than an end in itself.

Because DELCAP has the rather limited aim of providing measures of delay and throughput, it does not have to simulate in detail the path flown by each aircraft. All that need be represented in detail are those aspects of the flight of an aircraft which impede the free progress of other aircraft. If one aircraft is holding up a second, which thereby holds up a third, etc., it is necessary only to simulate the first and to accumulate correctly the total effects on the others.

Since in DELCAP we are interested only in delay and throughput, it is not necessary to know exactly where a delayed aircraft actually is during the delay period. It may be sitting in a line on a taxiway, or waiting at its gate. It may be flying a full race-track-like holding pattern, or just a U-shaped curve. Our implicit assumption in the current version of DELCAP is that aircraft are under control and can be delivered to the ILS gate for landing or to the end of the runway for takeoff, as soon as separation rules allow.

[This "perfect control" assumption is necessarily somewhat optimistic. It can be tempered by adjoining, to the model, appropriate probability distributions for time increments representing the deviations from "earliest possible delivery time" which reflect less-than-perfect control by controllers and/or pilots. Such "Monte Carlo" elements have been postponed in our initial development work in favor of concentrating on the basic model logic.]

DELCAP is a terminal-area simulation. Landings enter the model approximately at handoff to tower approach control, and leave it as they turn off the runway. Takeoffs enter the model approximately 15 minutes before departure, and leave at handoff to center control. The present version of DELCAP does not deal with on-the-ground activities except for the actual landing or takeoff on the runway. Its "accounting" includes a figure representing the time needed for a takeoff to depart its gate and taxi to the runway, but variations in this time due to taxiway and/or ramp congestion are not now modeled. In short, the priority decisions required by time and budget limits have led to heavy emphasis on airborne activities in this generation of DELCAP, with extension to ground activities seen as a natural "growth step" requiring more care and deliberation than was available as a "residual" in our present study.

The acronym "DELCAP" stands for DELay CAPacity, the two items the simulation was designed to measure. Delay is a user disutility while capacity is in some sense a system utility. The decision as to the proper balance between these two is a policy issue, and involves a variety of considerations relating to limitations on "system" budgets and personnel, value-of-time for operators and passengers, etc. The DELCAP model's role in this whole planning process is to aid by translating FAA rules, decisions, and airport configuration and operation changes into delay and throughput figures.

Since the scope of the model is much less than the full gate-to-gate span of a flight, the delay estimated in the DELCAP model is only a portion of the total delay experienced by an air passenger. Specifically, it is that portion of delay which may be affected by changes in(a), (b), or (c) of page 7. Delay is measured as the difference between (i) the time in an ideal scenario, in which the present landing or takeoff is the only aircraft using that terminal, and (ii) the time in the real situation where other planes may simultaneously be seeking to use the area.

For a landing, the time under the ideal scenario is a sum of: A minimum flying time from handoff to the outer marker of the desired landing runway; a time to fly the final approach path; and a runway occupancy time. The "real" scenario as simulated by DELCAP may include an additional time period (possibly zero) between handoff and the outer marker to account for other aircraft landing before this one. This period arises as follows. At handoff the aircraft is filed in a waiting queue. When the time comes that the aircraft could have reached the outer marker in the absence of other aircraft, the queue and the final approach path are examined. If there are other aircraft before this one in the queue, it must wait until it is finally at the head of the line. Then the "current obligations" of its final approach path and runway are examined to determine when this aircraft could first cross the outer marker and proceed down the final approach path without violating separation rules. The delay to this aircraft is then the difference between (i) when it could have left the queue were there no other aircraft interfering with it, and (ii) the actual time it was able to leave the queue. A similar process applies to a takeoff.

The "capacity" measured by the DELCAP model is actual throughput, the number of operations (landings plus takeoffs) performed in a given unit of time (e.g., an hour or a day). This kind of "capacity" measure differs from the common definition of capacity. The capacity of a jug is the amount of liquid the jug can hold. This represents the absolute limit for the jug, just this much and no more. The definition of capacity used in the FAA's Airport Capacity Handbook [4] refers to the maximum traffic rate which can be handled without average delay exceeding a prescribed level. Previous work by NBS [6] offered an alternative to this definition, based on a definition of capacity as maximum mean throughput rate.

The DELCAP model's capacity is a throughput measure more akin to this last than to that contained in the Airport Capacity Handbook. It differs from the previous NBS work in one significant respect -- it is not a maximum possible throughput rate. Rather it is that throughput rate achievable from a given distribution of traffic over the day (or other simulated time period). If traffic is light, throughput will be small. As traffic increases to a level which saturates the airport, mean throughput will approach the NBS-defined capacity. In busy periods the airport throughput will be close to capacity; as peak traffic levels pass to lower values, throughput will decrease. As traffic fluctuates, resultant delay will fluctuate, and both will be recorded by DELCAP. Thus DELCAP records the throughput resulting from a given temporal traffic distribution.

The DELCAP simulation is a critical-event model. That is, within the simulation, time is successively advanced from its current value to that of the "next significant event", rather than being stepped along in uniform increments. For periods with a low traffic level, this type of model saves much computation. Even during periods of high density traffic, such a simulation does not increase the number of calculations required. The SIMSCRIPT simulation language is designed for the critical event type of simulation. The modeler need only provide suitable descriptions of each type of critical event, including how these events depend upon one another. The SIMSCRIPT code then executes the events in chronological order. The critical events for the DELCAP system are illustrated in Figure 2.1.1. This figure is in the form of a flowchart because it illustrates the idealized progression of events. "Idealized" is used here to refer to the order of events as they occur for a single aircraft, without events for other aircraft interspersed among them. This is not necessarily the order in which the events occur. As an example, two hypothetical flights and an illustrative list of the critical events for each are given in Figure 2.1.2. The order in which the events would actually be performed by the simulation is given in Figure 2.1.3. Two events happen to occur at 9:17; the order of these two is immaterial.

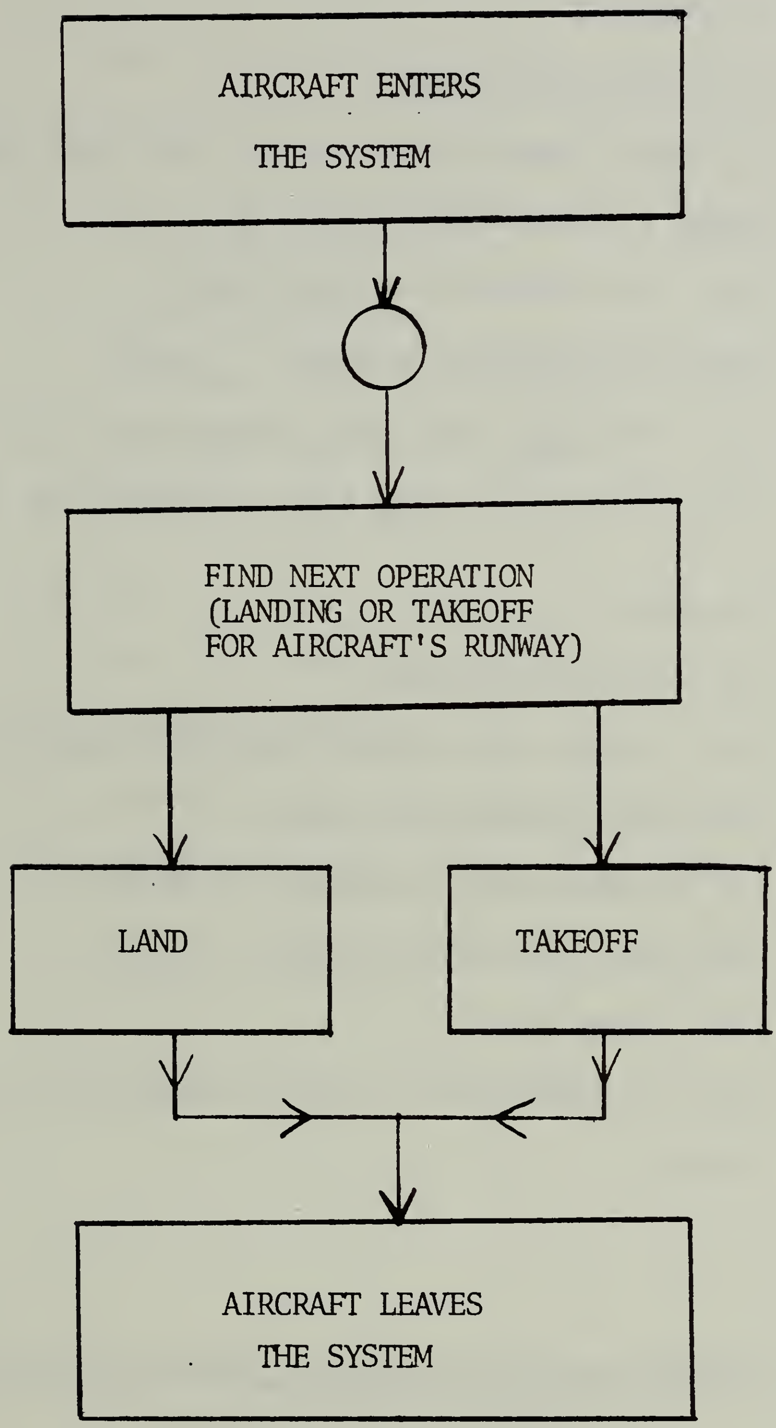

Figure 2.1.1: DELCAP Critical Events (Relative to a Single Aircraft)

FLT 1 : Landing on runway 1

(1) 9:00 - FLT 1 enters the system.

(2) 9:12 - FLT 1 has flown to outer marker; find next operation on runway 1 (assume next is FLT 1).

(3) 9:13 - FLT 1 has started down flight path; find next operation on runway 1 (none).

(4) 9:16 - FLT 1 has flown down final approach path, FLT 1 lands.

(5) 9:17 - FLT 1 leaves the system after occupying the runway.

FLT 2 : Takeoff on runway 2

(1) 9:05 - FLT 2 enters the system.

(2) 9:15 - FLT 2 leaves gate; begins taxiing to runway 2; find next operation on runway 2

(3) 9:17 - FLT has been given clearance to takeoff; find next operation on runway 2 (none).

(4) 9:18 - FLT 2 takes off.

(5) 9:23 - FLT 2 is handed off to center control; FLT 2 leaves the system.

Figure 2.1.2: Illustrative List of Critical Events for Two Flights

9:00 FLT 1 (1) enters the system

9:05 FLT 2 (1) enters the system

9:12 FLT 1 (2) find next operation on runway 1

9:13 FLT 1 (3) find next operation on runway 1

9:15 FLT 2 (2)find next operation on runway 2

9:16 FLT 1 (4) land

9:17 FLT 2 (3) find next operation on runway 2

9:17 FLT 1 (5) leaves the system

9:18 FLT 2 (4) takeoff

9:23 FLT 2 (5) leaves the system

Figure 2.1.3: Actual Order of Occurrence of Critical Events for Illustrative List

The basic concept around which the DELCAP simulation model is built is that of tieup. The presence of one aircraft limits the actions which another controlled aircraft may take. For example, present FAA separation rules require any IFR landing aircraft to be separated by 3 miles[1/] from other landing aircraft and 2 miles from preceding takeoffs on the same runway; other restrictions apply between takeoffs. Parallel runways have special rules, depending on how far apart the center lines for the parallel runways are. These various separation rules are imposed in the model by declaring certain points traversed by an aircraft to be off limits ("tying up" those points) to all subsequent aircraft for a period sufficiently long to insure that the required separation is maintained. In the present version of DELCAP, tieups can occur at one of three such points associated with each runway. Two pertain to landings, namely the outer marker and the runway threshold; the third, which refers to takeoffs, is the point at the end of the runway from which a takeoff starts its roll.[2/]

The 3-mile separation[3/] between successive landings is enforced, either by tying up the outer marker from the time the first aircraft crosses it until the second has flown three miles, or by tying up the runway threshold from the time the first aircraft has landed until the second has flown 3 miles. (Which of these alternatives applies depends on the relative landing speeds of the two planes.)

1/ All distances referred to in this report are in nautical miles Speed will therefore be in knots.

2/ Throughout the remainder of this report, the term "end of runway" will be used to designate this point. It and the runway threshold are assumed to represent approximately the same physical point on the runway.

3/ The 3-mile value and the values for the other separation distances are adjustable parameters in the simulation model.

The other separation rules are imposed in a similar manner. The prohibition against more than one aircraft being on the same runway at the same time is enforced by tying up the runway threshold and the end of the runway for the entire time a landing or takeoff occupies the runway. Interference between operations on intersecting runways can also be represented by the tieup concept: the end and threshold of a runway are tied up from the time an aircraft on an intersecting runway starts its roll or touches down until it clears the intersection. Interference between successive landings on parallel runways 3500 to 5000 feet apart can be portrayed by appropriately tying up the outer marker or the runway threshold of both runways.

The separation required between a pair of departing aircraft depends on the particular paths followed by the two aircraft after departure. If they diverge immediately only a 1-minute separation is required, if within five miles then a 2-minute separation is called for, etc.[4/] All but one of these separation rules are in terms of time; the exception, a distance separation (3 mi.), is converted to the weighted average (over aircraft types) of the time to fly that distance. Each takeoff is associated with a departure path when it first enters the model. The simulation assumes that aircraft diverge as soon as their departure paths diverge. Thus the separation required between two

4/ See Terminal Air Traffic Control (FAA 7110.8A) for a complete description of these rules.

takeoffs from the same runway depends solely on the pair of departure paths involved. An analogous second set of values for separation times applies to takeoffs on different runways. (This procedure for simulating separation between succeeding takeoffs is an approximation which may merit refinement in later work.)

Two final considerations will be emphasized in this overview. First, in keeping with the design of DELCAP as a planning tool, it is necessary that the simulation be fast to operate, while representing enough of the system to provide usable delay and throughput figures. The current DELCAP has been run to simulate a day's activity (about 1000 operations) at an airport with the runway configuration of New York's JFK; this took about 12 seconds on the UNIVAC 1108 computer. (The time-figure includes the time required for reading input data, but not that for program compilation.) It seems clear that with this speed, computer-cost and running-time considerations should not deter a planner from analyzing the full "reasonable" range of cases of interest to him.

Besides being fast, DELCAP must also be easy to operate if it is to be useful as a planning and analysis tool. The most time-consuming and laborious task in running any simulation is preparing the input. Both the job of collecting the information (if it is not readily available) and that of formatting it in computer-readable form can be difficult, indeed forbidding. These problems were recognized in the design of the DELCAP model. The simulation itself is written in SIMSCRIPT, which requires a rather awkward and rigidly prescribed input format. To overcome this, a FORTRAN preprocessing program was written. This program accepts input in a more easily prepared format and was specifically designed to make a user's task less laborious. Data are grouped into six categories. The user may specify any or all of these categories, or may elect not to specify a category, thereby accepting a standard set of values for that category. At present the configurations for two airports (ATL and JFK) are available as "standard" sets. Other airports may be included by assembling the necessary data and adding them to the input tape.

The six data categories are:

1. parameters describing each type of aircraft, such as landing and liftoff speeds and runway occupancy times,
2. mix of aircraft types using the terminal area,
3. number of aircraft, desiring landings and takeoffs respectively, to enter the model for each hour of the day,
4. separation rules,
5. description of the airport configuration and operation, and
6. distribution of runway use by aircraft type, separately for landings and takeoffs.

Standard values for 1 and 4 are independent of the airport, while standard values for each of the other four categories are provided for each airport in the airport data list. A user need specify only those data which he wishes to differ from standard values, plus the airport identifier for the standard values which depend on airport. This policy, of data input "by exception" to a standard data set, facilitates ease of input while still allowing full flexibility.

This concludes our overview of the DELCAP model and of the design criteria under which it was built. The next section will describe in greater detail how DELCAP simulates terminal-area activities. Its preprocessor will be described in the following section.

2.2 Specific Description of Simulation Model

To explain the operation of the DELCAP model in further detail, we will in this section "fly" a hypothetical aircraft through the simulation, describing how its progress is represented by the various elements of the model. The next chapter describes in detail the computer-program implementation of this procedure. Here, our aim is a functional description of the simulation, rather than a computer-oriented view.

Figure 2.2.1 describes the critical events which occur to a single aircraft:

1. generation of flight,
2. sequencing of operations on a runway,
3. maintain separation, and
4. land or takeoff.

A landing aircraft enters the system at handoff to tower approach control and is filed in a landing queue(for its runway)in which it remains until the minimum time to fly from handoff to the outer marker has elapsed. The next event which affects this aircraft is the choice, depending on the operational procedure for its particular runway, of the next operation (a takeoff or a landing) which is to occur on this runway. Once the aircraft has come to the head of the queue and the next operation is a landing, the model must ensure that the aircraft does not violate any separation rules pertaining to it. Finally it is removed from the queue, is scheduled to land, and the relevant points are tied up to ensure that other aircraft will remain separated from it.

A takeoff enters the system about 15 minutes before departure as scheduled in a filed flight plan and at this time is filed in a queue in which it remains until a few minutes before scheduled takeoff. When the flight has reached the head of the queue and the next operation to be scheduled is a takeoff, the flight is removed from the queue, and is scheduled to takeoff as soon as possible while maintaining required separation. Once it has been scheduled relevant points are tied up to ensure other aircraft are separated from it.

The rest of this section will describe each of these four critical events in greater detail.

2.2.1 Flight Generation

Aircraft enter the simulation in two ways, one stochastic and the other deterministic. In any one simulation run, either or both methods may be used. In the stochastic form, aircraft are generated in a Poisson manner. (Employment of this particular distribution is somewhat arbitrary, and it can be replaced by a different one if found more appropriate.) The use of a Poisson distribution in effect assumes that the probability of the arrival of an aircraft in the system at any time is independent of any previous arrivals.

Such randomness may be an appropriate representation for low or moderate traffic densities. In a highly saturated air traffic system, however, congestion in terminal areas is felt in other sectors as well. Since IFR traffic is under control from takeoff to landing, traffic patterns are in fact not random. When a terminal area becomes saturated, flights destined for this terminal may be held up at distant points and handed off to the terminal area only as a slot in a stack becomes available. This extreme case, almost diametrically opposite to Poisson randomness, represents a nearly uniform distribution of arrivals into the terminal-area landing pattern.

A similar situation may exist with regard to takeoffs: the arrival of a takeoff in the "system" is about 15 minutes before departure as scheduled by filed flight plan, but in actual practice when the flight plan is filed the pilot can be advised of expected delay and may revise his flight plan. The simulation deals only with revised plans, which in heavy traffic would tend to exhibit a less random behavior.

The particular form of Poisson generation used in DELCAP does allow representation of the diurnal peaking of traffic density. The user can specify both the mean number of landing aircraft to be generated for each hour of the day, and the mean number of takeoffs. The expected time (as a fraction of an hour) between generation of successive takeoffs or landings is computed as the reciprocal of the number of takeoffs or landings entering the system per hour. If the next scheduled landing or takeoff to be generated enters the system in the next hour, it is rescheduled according to the generation rate for the proper hour. This avoids a situation such as the following:

(a) Hour 1 has 1 landing per hour.

(b) Hour 2 has 12 landings per hour.

(c) Flight 1 is generated (say) at 1:30; Flight 2 is then scheduled (say) for 2:30 on the basis of the "1-hour spacing" rule still in force at 1:30.

If the simulation were really to wait until 2:30 to start generating flights an average of five minutes apart, then the expected number of flights for Hour 2 would be only half as large as it should be. The rescheduling procedure described above can be shown to provide the correct expected number of landings. For the preceding example, flight 2 would initially be scheduled for 2:30. At 2:00, however, it would be rescheduled using the expected time until the next landing as 5 minutes. Subsequent flights would continue to use this 5-minute figure until some flight was scheduled for Hour 3. At 3:00, this flight would be rescheduled, and so on.

A second, deterministic method of generating flights is available. The simulation may be provided (on a magnetic tape or on cards) with a list of landings and takeoffs in the order of their entries to the system. This form of traffic generation may be used either instead of or in conjunction with the Poisson generation described above. For example, scheduled air carrier flights might be input in a deterministic fashion while unscheduled flights could be generated by the Poisson process. (It should be noted that in this case the input values of the expected numbers of landings and takeoffs for the Poisson process should reflect unscheduled traffic levels only, not the combination of both.)

We should emphasize here that in a computer, the stochastic form is "pseudo-random" rather than truly random. If the same starting number ("seed) for the computer's pseudo-random-number generating routine is used in two runs, exactly the same sequence of "random" numbers will occur. Therefore the stochastic mode of traffic generation provides a repeatable traffic sample, which may properly be used in comparing various alternatives. If however the expected number of landings or takeoffs per hour is changed between two runs, then different traffic samples will occur.

Each flight which enters the system is (1) designated as either a landing or a takeoff, (2) associated with a specific runway, (3) identified as of a particular aircraft type category, and if a departure, (4) assigned to a particular departure path. In the stochastic generation process these selections are made through a random process, while for deterministically generated flights they are read in as part of the flight description. For the stochastic process, the simulation must be provided with (1) one distribution of aircraft types for landings and a second one for takeoffs, (2) the distribution of runway use by aircraft type for landings and also for takeoffs, and finally

(3) the distribution of departure paths for aircraft using each runway. As a flight is generated, the appropriate distributions are sampled to assign to it a type, a runway and (if it is a takeoff) a departure path.

[It should be clear from the description of this process, that the runway choice is not influenced by the traffic levels of the various runways at the time of generation. That is, the runway-use distribution presently can reflect only general traffic levels for each runway; this distribution is not altered within the simulation, remaining constant throughout. Permitting such "adaptive" alterations would require only a fairly minimal change in the DELCAP model, but the user would then have to specify those "threshold" traffic levels at which a change in runway assignment would be encouraged.]

After a flight is generated and has received its proper runway, type, and departure path assignments, it is placed into a queue of aircraft waiting to land or takeoff from its assigned runway. There are two separate queues for each runway, one for takeoffs, and another for landings. These queues are organized in a first-in first-out (FIFO) manner. This means that aircraft which are to land on a particular runway land in the order in which they enter the system, regardless of speed or category of aircraft. It does not mean, however, that aircraft landings on different runways (or landings and takeoffs on the same runway) occur in the order in which they entered the system.

This first-come first-served process models the presently stated control policy. To model a more sophisticated sequencing policy, the queues could instead be ordered based on some attribute such as type classification or flight priority level. (SIMSCRIPT list structure allows such changes with a minimal amount of recoding.)

Landings remain in the queue for at least the minimum time for that type of aircraft to fly from handoff to the outer marker of the proper runway. Takeoffs are generated about 15 minutes before planned departure and are filed into the queue at this time.

2.2.2 Operation Sequencing

The next section of the simulation which affects our hypothetical flight is the routine which sequences operations on runways. If for example our aircraft is to land on "runway R", then it can move up in its queue only when this routine, applied to runway R, selects the lead operation of that runway's landing queue rather than its takeoff queue.

There are four different operational procedures which govern whether the next aircraft to use the runway in question will be a landing or a takeoff. Runways may be restricted to landings only or takeoffs only. In either of these two cases the simulation will schedule the next operation of the permitted type if one is available.

The last two procedures apply to dual-use runways (used for both landings and takeoffs), and are slightly more complicated. The first of them seeks to alternate landings and takeoffs during heavy traffic periods (and to take the first available operation otherwise). If the last operation was a takeoff, for instance, then the landing queue is examined. If the first flight in it will be available when the runway and final approach path are next free, then that landing is designated as the next operation on this runway. However, if the landing queue is empty or the first aircraft in it will not be immediately available, the takeoff queue is examined. If the first flight in the takeoff queue will be available immediately, it is designated as the next operation. But if neither landing nor takeoff will be available immediately, the one which will be available first is chosen. (Of course if there are no aircraft at all in either queue to use this runway at the time, then no "next operation" is scheduled.)

The fourth procedure attempts to model the "landings take precedence" rule. This procedure is very similar to the one described above for alternating operations, except that the landing queue is always examined first. If the first landing is available immediately then it is assigned as the next operation. If not, then that operation (landing or takeoff) which would first be available becomes the next operation. At present the model provides no method to change the operational procedure for a runway during the simulation, although it would require only a minimal change to do so. The user would then be required to specify the conditions or times for which this should occur.

2.2.3 Maintain Separation

The next block of the model which affects our hypothetical flight is the section which insures that required separations are maintained. Relevant when the flight reaches the head of its designated runway's landing or takeoff queue, this routine finds the first time the flight can land (or takeoff) on that runway. For takeoffs the end of the runway is examined, and the flight is allowed to start its roll as soon as this point is free (no longer tied up). For landings both the outer marker and the runway threshold are examined, since a landing must pass both of these points: Let t_1 be the time at which the outer marker will be free(no further tieups for it are currently known), and t_2 the corresponding time for the runway threshold. Let T be the time it takes this flight to fly the final approach path from the outer marker to the runway threshold. Then the flight may start to fly down the final approach path at whichever of t_1 and t_2 -T is later.

Several tieups may be in force at any of the three points: outer marker, runway threshold and the end of the runway. The model always forces an aircraft to wait until there are no currently known tieups applicable along its path, even though theoretically there might be a gap between tieups sufficiently large to allow fitting an operation between two previously scheduled ones. That this is an infrequent event in practice depends on the fact that in the model takeoffs and landings are scheduled approximately the same time before touchdown and start-of-roll respectively. For any runway the takeoffs are scheduled in the order in which they are generated, and similarly for landings. The sequencing routine takes care of the order of operations on any one runway and takes into account the times at which those operations can occur. Therefore the only case in which a flight might be squeezed in between two tieups would involve inserting that flight before a tieup caused by a flight which has been previously scheduled on another runway. This circumstance is unlikely to occur except in the situation of a slow landing which has just started down the flight path on one runway and a fast takeoff which could take off on a second runway and remain safely separated from the landing.

Allowing in the simulation for such "insertions" would require major complication of the model logic. Granting that the situation will seldom occur (indeed we have only seen it once or twice in our debugging runs), its overall effect on aggregate measures of capacity and delay is minimal. On this basis, it was decided that the extra complication would be unwarranted.

2.2.4 Land or Take Off

Finally, our hypothetical flight is ready to land or take off. In the simulation this process essentially consists of tying up the appropriate points to ensure that "following" aircraft maintain required separation from the given one. "Following" is used here to refer to those flights whose landing or takeoff is scheduled after the current one. In order to maintain the required separation between landings either the outer marker (if the following landing is slower) or the runway threshold (if the following landing is faster) is tied up from the time the current flight passes that point until the second flight has flown the separation distance. Both the runway threshold and the end of the runway are tied up for the period the current aircraft occupies the runway. A landing ties up the end of the runway, and a takeoff the runway threshold, for a time period sufficient to maintain the separation between a takeoff and a following landing. (The formula for this separation, based on the assumptions of constant final approach speed and constant takeoff acceleration, is derived in Appendix F.) A takeoff ties up the end of the runway for a separation time required between takeoffs. Points on other runways with which the present runway interferes are tied up in a similar manner. For intersecting runways, the runway threshold and the end of the runway are tied up until a landing or a takeoff passes the intersection point or turns off the runway. (The formula for the time until a landing or a takeoff reaches any point down the runway is derived in Appendix G.)

This concludes the progress of our hypothetical flight, except for a final routine which records the associated delay. This delay is recorded at touchdown for a landing, and at the start of roll for a takeoff. Therefore the delay attributed to any hour only refers to delay suffered by an aircraft which touched down or started its roll in that hour. The delay itself may have occurred in the previous hour, or possibly partially even before that.

To summarize, our hypothetical flight has gone through six steps:

1. generate flight;
2. file flight in the appropriate queue;
3. find the next flight for any runway;
4. find the first time this flight can use its runway and (for landings) its final approach path, without violating separation rules;
5. tie up the appropriate points to maintain separation following this flight; and
6. record flight delay.

2.2.5 Outputs

The final portion of this section will be devoted to describing the outputs available from the present version of DELCAP. We will describe later, in Chapter 5, the kinds of output which could be readily provided with minor changes in the program; it is sufficient to note here that the current set of outputs is only representative of the potential outputs available from the present version of DELCAP. Toward the end of this current work, a proposed set of "mock instructions" describing the DELCAP model was circulated within the FAA. This document was designed to elicit comments on the model's usefulness and descriptions of specific problem scenarios in which it could be applied. It was hoped that answers to the questionnaire accompanying the "mock instructions" would provide a better description of desired model output. However this particular goal of the circulation was not achieved, and so the present model output remains only illustrative of potential outputs. (The "mock instructions" are repeated as Appendix H.)

Figure 2.2.1 shows a typical example of model output. Chapter 4 will include a more complete description of model runs; we are here mainly concerned with the form of the output. The section which follows this will describe the preprocessor and the output from it, which provides labeling for a run by describing that run's input specification. The output in Figure 2.2.1 however, is from the simulation model portion only.

The airport configuration for this run is that of the Atlanta airport (ATL). Delay and throughput figures were accumulated for 24 hours, starting and ending at 8 a.m.[5/] The simulation was actually run for 25 hours, starting at 7 a.m. rather than 8 a.m. to allow preloading the system so that the delay and throughput calculations could start off with some traffic already present. The starting hour of 7 a.m. was chosen because there is very little traffic at this time and one hour's preload seems to be sufficient.

5/ The column labeled HOUR refers to the hour of the day, so that hour 1 is from midnight to 1 a.m., hour 8 is from 7 to 8 a.m., and hour 17 is from 4 to 5 p.m. The first hour for which delay was recorded is therefore actually hour 9.

SUMMARY REPORT FOR THIS RUN

HOUR	OPERATIONS GENERATED			OPERATIONS PERFORMED			TOTAL DELAY (MINUTES)			DELAY PER A/C (MINUTES)		
	LANDINGS	TAKEOFFS	TOTAL	LANDINGS	TAKEOFFS	TOTAL	LANDINGS	TAKEOFFS	ALL	LANDINGS	TAKEOFFS	ALL
1	2	4	6	4	17	21	9.5	728.5	737.9	2.4	42.9	35.1
2	0	4	4	2	4	6	0.	0.	0.	0.	0.	0.
3	19	1	20	16	1	17	35.3	0.	35.3	2.2	0.	2.1
4	1	3	4	4	4	8	11.3	2.8	14.2	2.8	0.7	1.8
5	0	5	5	0	5	5	0.	0.	0.	0.	0.	0.
6	5	1	6	4	0	4	0.	0.	0.	0.	0.	0.
7	4	1	5	4	2	6	0.	0.	0.	0.	0.	0.
8	3	9	12	4	8	12	0.6	2.7	3.4	0.2	0.3	0.3
9	1	20	21	0	16	16	0.	11.6	11.6	0.	0.7	0.7
10	35	14	49	21	17	38	48.9	12.2	61.0	2.3	0.7	1.6
11	33	26	59	33	20	53	452.9	61.2	514.1	13.7	3.1	9.7
12	41	24	65	35	26	61	721.0	68.5	789.5	20.6	2.6	12.9
13	44	44	88	33	27	60	762.2	267.0	1029.3	23.1	9.9	17.2
14	46	47	93	38	33	71	1467.0	986.2	2453.2	38.6	29.9	34.6
15	33	51	84	30	28	58	1563.2	1159.5	2722.7	52.1	41.4	46.9
16	33	53	86	33	29	62	2191.7	2085.1	4276.8	66.4	71.9	69.0
17	62	41	103	34	30	64	2312.2	3061.6	5373.8	68.0	102.1	84.0
18	60	54	114	27	28	55	2003.6	3400.4	5403.9	74.2	121.4	98.3
19	31	25	56	38	34	72	3904.7	5167.0	9071.7	102.8	152.0	126.0
20	13	15	28	37	27	64	4467.4	4608.9	9076.3	120.7	170.7	141.8
21	11	12	23	32	31	63	4779.9	5718.1	10498.0	149.4	184.5	166.6
22	6	11	17	39	38	77	5848.4	8272.8	14121.1	150.0	217.7	183.4
23	10	17	27	28	32	60	1515.4	6208.2	7723.7	54.1	194.0	128.7
24	20	9	29	17	35	52	42.9	4226.0	4268.9	2.5	120.7	82.1
TOTAL	513	491	1004	513	492	1005	32138.1	46048.3	78186.4	62.6	93.6	77.8

Figure 2.2.1: Sample Model Output

This extra hour of simulation also accounts for the fact that one more takeoff was performed than was generated. Even without delay, the time between generation and takeoff or landing is 15 - 20 minutes. Therefore, those aircraft landing or taking off during the first part of hour 9 (8 am to 9 am) are actually generated during the previous hour, but are not recorded. Similarly, toward the end of the simulation aircraft are generated which land or take off only after the simulation has finished. Therefore, there is no a priori reason for the recorded number of operations generated to coincide with the number performed.

The distributions of landings and takeoffs were taken from data pertaining to San Antonio International Airport. However about 60 percent of San Antonio traffic is VFR, whereas in our present DELCAP model all this traffic is required to obey IFR rules. This explains the high level of simulated delay, which would certainly be intolerable at a real airport. This run and the others described in this report should be interpreted as examples of possible model runs, rather than runs based on actual data. (Within the time and resource limits of the current effort, no search for or assimilation of data sources was possible.) The run does, however, illustrate the capability of the DELCAP model to portray peaking of landings and takeoffs and the resultant delays.

It should be remembered when examining these program outputs that the delay recorded here includes only that encountered in the terminal area of this particular airport, between handoff to tower approach control and landing , or between desired departure time and handoff to center control for a takeoff. "Delay" refers only to the extra time above the minimum required if no other traffic interfered. Airlines expect delays at certain airports, and adjust their scheduled arrival times to reflect this. The model does not include the expected time of arrival for any flight. Rather it computes the minimum time for the flight to fly from handoff and land, or to taxi to the runway and take off. Delay is then calculated as the difference between actual time and the time in this ideal scenario.

This concludes our sketch of model output format. The next section will describe the preprocessor. During the actual running of the model, the preprocessor precedes the simulation portion of DELCAP. We have here described the simulation first since its input-data requirements are what the preprocessor is designed to satisfy.

2.3 Specific Description of Preprocessor

Since the DELCAP model was designed primarily for use by planners and analysts, ease of input was a major consideration in its formulation. The SIMSCRIPT computer simulation language, chosen because it greatly facilitates simulation-model programming, unfortunately requires input in a rigid and awkward format. Therefore a preprocessor program was written to aid the user in inputting data to the simulation.

It is hoped that DELCAP will ultimately operate in a time-sharing computer environment in which an analyst would need only to (1) call up the separation rules, and the configuration and traffic levels, for the terminal area of interest in the year under investigation, (2) make any desired alterations to these called-up "standard" data, and (3) receive his delay and throughput measures, all in a relatively short time. The user should have to learn only a bare minimum about working with the computer (perhaps only how to arrange accounting procedures and associated job numbers and how to call up the DELCAP program). The DELCAP model would have, on file, airport configurations and traffic levels (present and projected) for most of the terminals in the country for the current and desired projection years. The model would, if the user requested, print out the information contained in any of its files. The user could modify (temporarily) any or all of the contents of any file for any particular run. Those items he did not wish to modify would automatically be retrieved from the proper file stored in the computer, and used without change in his run. This philosophy of input by exception to entries in a standard file, we believe to be an important feature which would make a model such as DELCAP much easier to use and therefore much more apt to be used.[6/]

The present form of the DELCAP model has not yet achieved this level of development. The current effort was aimed primarily at formulating and implementing the simulation model logic. To indicate how the proposed final form of DELCAP would operate, and also to aid in data input to the current stage of development, a FORTRAN preprocessing program was written. The DELCAP modeling system thus presently consists of two parts, the preprocessor and the simulation model, which are run in succession. On the UNIVAC 1108 these two programs can be executed one after another in a single run. For computer systems which do not allow multiple executes, two runs would be necessary. The preprocessing program accepts card and/or tape input as desired, and produces both a printed run description and a tape containing a SIMSCRIPT initialization deck[7/], which in turn provides input to the simulation model. The simulation is then run to produce the delay and throughput statistics for that particular set of inputs.

The present file of airport descriptions contains "data" for only two airports, JFK and ATL. Even these are only approximate. The runway configurations were obtained from [20] which does not include the distances (outer marker to end of runway, and end of runway to intersection) needed by DELCAP. The operational procedures (landings only, takeoffs

---

[6/] This was suggested to us by the FAA, and after considering it carefully, we enthusiastically concur.

[7/] For a further description of the SIMSCRIPT initialization deck see SIMSCRIPT, A Simulation Programming Language, by Markowitz, Hausner and Karr. A sample deck for the DELCAP simulation appears in Appendix A.2.

only, dual-use alternate operations or dual-use landings take precedence), and the distribution of runway use, could only partially be inferred from this text. No information was given on how landings and takeoffs were distributed over the hours of the day. Therefore much of the "data" for these two airports have been invented either wholly or in part. The distribution of landings and takeoffs over the day for both ATL and JFK was taken from another source [22] and applies to a different airport entirely. The computer runs described later and the existing data file, therefore, remain only illustrative of model capability. They do provide, however,a demonstration of how the model would operate as an aid in planning and analysis.

As mentioned earlier in section 2.1, the preprocessor divides model input into six data groups. The user may specify any or all of these groups. The particular division employed is somewhat arbitrary and may be altered if experience running the program suggests a better arrangement. The number of data groups could also be changed, but there is a tradeoff between the flexibility of a larger number of data groups and the minimum amount a user must specify. The six groups are:

1. description of aircraft types,
2. mix of aircraft types,
3. landing and takeoff rates by hour of day,
4. separation rules,
5. airport configuration and
6. fraction of each type of aircraft using each runway and departure path.

Input for data groups 1 and 4 does not depend on the airport, and so there is only one set of standard values for these groups. The "standard" separation rules are taken as:

a. 3 mile interlanding separation,

b. 2 miles separating a landing from a preceding takeoff for aircraft with Distance Measuring Equipment,

c. 4 mile departure/arrival fix for aircraft without DME,

d. 1 minute separation between diverging aircraft, 3 minutes between aircraft using the same departure path.

The standard type classification is that given in the Airport Capacity Handbook, pages 2.10 through 2.13. Figure 2.3.1 records the standard values of variables for each aircraft type.

The remaining four data groups depend on the airport involved. The user may, if he so desires, choose different groups from the file entries for different airports. (The identifier appearing on the preprocessor output will be the one associated with airport configuration.) Care should be taken when mixing inputs from different airports that they all have the same number of runways and departure fixes. In describing the airport configuration the user must include for each runway a name consisting of a two-character heading (00 is north, 13 is 130° from north or roughly southwest, etc.), and where desired a one- or two-character modifier (e.g., 13L and 13R for parallels, 13FL for a third parallel to the left of 13L). These are used only for identifying runways in the preprocessor output and are not needed by the simulation.

TYPE	SPEEDS (KNOTS)		RUNWAY OCCUPANCY (SECONDS)		HAS DME?
	LANDING	LIFTOFF	LANDING	TAKEOFF	
A	145	155	50	33	YES
B	140	148	44	30	YES
C	130	140	39	28	YES
D	115	123	33	23	NO
E	98	110	29	21	NO

Figure 2.3.1: Standard Characteristics of Aircraft Types

In describing the runway configuration the user must specify two kinds of data: (1) for each pair of intersecting runways, the distance from the end of each of the runways to the intersection with the other, and (2) for each pair of runways, the type of interference between them. There are two types of interference. In the first type the two runways are completely dependent, i.e. landings and takeoffs on one runway must be separated by the same distances from those on the other as from those on the same runway. The second type of interference requires successive landings on each one of the pair of runways to be separated by the same distance as landings on a single runway, but landings on one runway do not affect takeoffs on the other, and simultaneous takeoffs are allowed from the two runways if the takeoffs diverge. If the user does not specify one of these types of interference, the model assumes that operations on the two runways do not affect one another.

Figure 2.3.2 and 2.3.3 present sample output from the preprocessor. They describe the input for the simulation run whose output was given in Figure 2.2.1. Figure 2.3.2 gives (a) the hours for which the simulation was run (in this case, the 24 hours from 8 a.m. one day to 8 a.m. the next), (b) the characteristics of each aircraft type, and (c) the mix of aircraft types. Figure 2.3.3 gives a verbal description of the airport (in this case our hypothetical ATL). The airport description includes the operational procedure for each runway and the interference patterns among the runways. Future DELCAP refinement might include graphical (schematic, map-like) output of the airport runway configuration. Note, however, that the information now required of the user is not always sufficient to locate all runways in relation to each other. If two runways are independent, the only information available is the heading specification. Since the user is not required to specify the runway endpoint and length, there is no information available in the current data set indicating that the two parallels 9L and 9R for ATL are offset parallels. In order to have graphical output, then the user would have to enter extra information about the airport (e.g., describing the runway endpoint in some x-y grid) which is not needed by the simulation program and whose provision would require extra work. This runs counter to the motivation for the preprocessor, namely to allow the user to provide input in an easily prepared form requiring a minimum of data collection and preparation. It was therefore decided that the present verbal airport description is adequate for the current effort and may even be preferable to a pictorial description.

Many variations in the preprocessor output are possible. Since the DELCAP model has not yet been applied to a real-world problem, it is impossible at this stage to anticipate which of the possible outputs would be most useful and in what form they should appear. The output of the preprocessor is designed primarily to label a run, describing those elements which differentiate that run from others. Therefore, which of the inputs should be printed out, and how best to display them to make the differences between two runs most obvious, depends on the particular application. The present form, to be regarded primarily as an example, has proved useful for the debugging and testing runs of the current effort.

```
AIRPORT CONFIGURATION
FOR ATL AIRPORT


NUMBER OF RUNWAYS = 3

RUNWAY 1 ( 9L ) - TAKEOFFS ONLY

RUNWAY 2 ( 9R ) - LANDINGS ONLY

RUNWAY 3 (15  ) - DUAL USE, ALTERNATING OPERATIONS


RUNWAYS  9L  AND 15   INTERSECT AT A POINT  2400. FEET FROM THE
END OF RUNWAY  9L  AND  1677. FEET FROM THE END OF RUNWAY 15  .


RUNWAYS  9R  AND 15   INTERSECT AT A POINT  5997. FEET FROM THE
END OF RUNWAY  9R  AND  6520. FEET FROM THE END OF RUNWAY 15  .


RUNWAYS  9L  AND  9R  ARE SEMI-DEPENDENT PARALLELS -
SIMULTANEOUS ARRIVALS ARE PROHIBITED
```

Figure 2.3.3: Sample Preprocessor Output
Airport Configuration

THIS SIMULATION RUNS FROM 6.00 TO 8.00

AIRCRAFT DESCRIPTION

TYPE	SPEEDS (KNOTS) LANDING	SPEEDS (KNOTS) LIFTOFF	RUNWAY OCCUPANCY (SECONDS) LANDING	RUNWAY OCCUPANCY (SECONDS) TAKEOFF
1	145.	155.	50.	33.
2	140.	148.	44.	30.
3	130.	140.	39.	28.
4	115.	123.	33.	23.
5	96.	110.	29.	21.

TRAFFIC DESCRIPTION

TYPE	LANDING MIX	TAKEOFF MIX
1	19.	19.
2	58.	58.
3	2.	2.
4	14.	14.
5	7.	7.

Figure 2.3.2 Sample Preprocessor Output Aircraft and Traffic Descriptions

This concludes our functional description of the preprocessor and simulation. The next chapter will provide a more technically oriented description, tied more closely to the implementation of the model as a computer program. The reader less concerned with such matters may wish to skip to Chapter 4, which describes the outputs from illustrative runs of the model.

3. DESCRIPTION OF COMPUTER IMPLEMENTATION

3.1 General Remarks

The previous chapter described the model from the viewpoint of a reader familiar with terminal-area operations, but not necessarily with computer modeling. In this chapter, we present the model from the latter point of view.

Since the computer simulation model was written in SIMSCRIPT, we will often find it convenient to describe the model using the terminology of that language. The reader may thus desire to refer to SIMSCRIPT, A Simulation Programming Language, by Markowitz, Hausner, and Karr for a more complete account of these terms. A partial glossary of terms which will be used frequently follows:

event - The simulation program consists of a series of subprograms (called event routines) describing the change in the status of the system [1/] at each critical event. There are two types of events: exogenous events, which happen at specific times input to the simulation, and endogenous events, which occur as a consequence of some preceding event in the simulation.

1/ System here refers to that which is being modeled, the terminal area.

function - In addition to event subprograms SIMSCRIPT includes "function" routines which are similar to FORTRAN FUNCTION programs in that they calculate a value of a function for a given set of values of certain parameters.

entities - Entities are the objects which the simulation models. There are two types: temporary entities, which during the course of a simulation come into the system and later leave (e.g., flights), and permanent entities whose number remains fixed during any simulation run (e.g., runways and aircraft types).

attribute - Attributes are properties associated with temporary entities and events. They provide additional information about the specific event or entity, such as the type of a given flight or the identity of the runway upon which a given landing is to occur.

set - Sets are lists of particular entities. When an entity is added to the list it is said to be filed in the set. Sets may be ordered in any of several ways. A FIFO set is ordered in a first-in-first-out manner (e.g., the landing and takeoff queues). The opposite is a LIFO set, last-in first-out. Finally, sets may be ranked on some attribute. An example of of this in the simulation is the set of tieups associated with any of a runway's three interference points. These sets are ranked on the time at which the tieup is no longer in force.

schedule - When using the SIMSCRIPT language, the programmer need only write code for each of the critical events. Endogenous events (which occur as the result of other events) must be scheduled within the causing-event routine. These scheduled events are then put in a list of future events; after finishing each event, the SIMSCRIPT system checks to find the next event in this list.

create and destroy - When temporary entities enter the system they are said to be created, and when they leave they are said to be destroyed. In the computer, when an entity is created storage is reserved for it, and when it is destroyed this storage is returned to the SIMSCRIPT program and can be reassigned.

The DELCAP model is currently operable under the EXEC II operating system on the UNIVAC 1108 at the National Bureau of Standards. The preprocessor is written in FORTRAN V (UNIVAC's augmented FORTRAN IV), and the simulation in SIMSCRIPT 1.5. A minimum of difficulty is anticipated in converting the program to other computers, since care has been taken to design the program to be compatible with other computer systems. However, to aid in the conversion process we have included in Appendix E a list of possible areas of computer incompatibility which have arisen in other efforts in which members of this project staff have participated. SIMSCRIPT 1.5 is available on a wide range of computers, and most of the compilers were written by a single company, increasing the likelihood of easy transference.

The present preprocessor consists of about 500 FORTRAN statements, including comment cards as necessary. A listing of the program appears in Appendix C. The example preprocessor run described in the previous chapter took about 3 seconds to execute. This running time will be increased, however, when the file of standard airport data becomes large, since much of the execution time would then be spent in searching for the correct airport information. The preprocessor uses about 6600 words of core storage. The current limits on the number of runways, aircraft types and departure paths depend more on some of the formats in the preprocessor than on actual core limits. These limits are: 9 runways, 10 types, and 5 departure paths. The maximum total number of interferences allowed is 10. These values are parametrized in the preprocessor, and may be changed along with necessary formats if these limits need to be increased. It should be noted here that the term "runway" refers to a runway operating in a particular direction. (13 and 31 are two separate runways.) Since the runways are fixed for a particular run, and runways are not usually operated in both directions at once, this does not present too great a problem. However, there could be a problem in the future when the tape of standard airport data is created, since there would then be several different runway configurations for each airport which must be distinguished, (e.g., for Washington National, DCA north operations vs. DCA south operations).

The simulation model consists of about 1300 SIMSCRIPT statements including comment cards. A listing appears in Appendix D. The example runs took about 9 to 10 seconds to simulate 25 hours. (The simulation was run for 1 hour to preload the system and then for 24 hours to accumulate output statistics.) This time is exclusive of compilation, which takes about 1.5 minutes. Of course, once the DELCAP model system is in the production phase, compilation would no longer be necessary.

On the 1108, SIMSCRIPT compilation is a two-stage process. First the SIMSCRIPT object code is compiled into SLEUTH, the 1108 assembly language; then the SLEUTH is assembled. Storage has never become a problem with the simulation program. Even with 20 runways, 200 types of aircraft and 40 departure paths, the data storage requirements are less than 20,000 words. For an airport with 20 runways there could be a maximum of about 5,000 additional storage locations used to store information concerning temporary entities and events. These estimates are all very conservative. (No existing airport has 20 runways, for instance.) Thus it is not anticipated that computer storage requirements will limit the applicability of the present version of the DELCAP model.

The minimum number of data cards required for a run of DELCAP is three, two for the preprocessor and one for the simulation. The user must supply at least the first and last hour for the simulation, and also the preprocessor parameter card indicating which of the data groups should be standard data and for which airports. The simulation requires a SIMSCRIPT system specification card[2/] which will be described in greater detail in Section 3 of this chapter.

This concludes our general remarks on the computer implementation of the DELCAP model. The following two sections will describe the preprocessor and simulation programs respectively.

[2/] See pages 101 and 102 of SIMSCRIPT, A Simulation Programming Language for a further description of this card.

3.2 The Preprocessing Program

The preprocessor was designed to aid the user in providing input to the DELCAP model. It has two major functions. First, it accepts user - supplied input in a relatively easily-prepared format, and outputs these data in the more cumbersome format required by the simulation. Second, it provides "standard" values for those input items which the user does not wish to specify . In addition, it checks the input for consistency and prints warning messages as necessary. Figure 3.2.1 is a flowchart of the preprocessor program. Although the middle section appears as a loop, the six data groups differ enough in structure as to require six separate portions in the program. As is seen from the figure, the program can be broken into four sections:

1. Read the hours to be simulated and the options card.
2. Compile the data for the six data groups, from user-supplied input or standard files as directed.
3. Write a tape providing simulation input.
4. Print the run identifications.

The program always requires at least two input cards. The first of these specifies the beginning and ending times for the simulation. Time is expressed in military time, e.g., 0.00 stands for midnight, 8.30 for 8:30 a.m. and 17.20 for 5:20 p.m. The program will not at present accept a run of over 24 hours. (Note: a "24 hour run" is one which simulates 25 hours but records delay only for the final 24.) A run from 8.00 to 8.00 is interpreted as a 24 hour run from 8 a.m. one

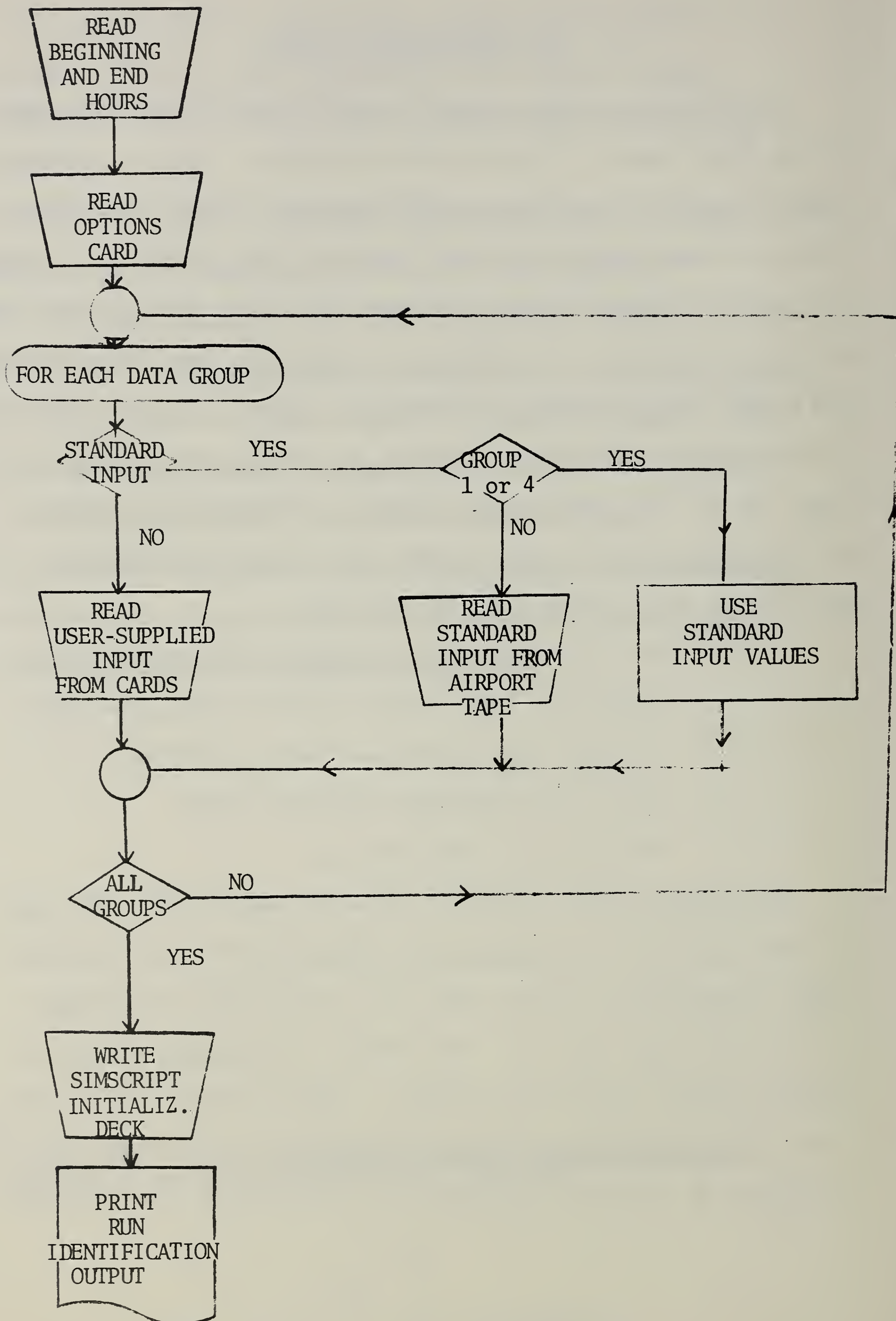

Figure 3.2.1: Preprocessor Flowchart

day to 8 a.m. the next. Thus the input to the simulation reads from hour 8 to hour 32 in this case. The simulation is always run for at least one full hour (and up to two), to preload the system before beginning to record delay and throughput. If the simulation is to start on the hour, it will be started exactly one hour early. If the simulation is to start at 5:45, it will be run for an initial hour and 45 minutes. The maximum preload time, then, is one hour and 59 minutes. The simulation always begins its preloading at the hour, but it will end at whatever minute the user has specified.

The <u>second</u> data card required by all runs of the preprocessor specifies which of the six data groups will use standard input. Even if the user wants to specify values for all six groups, this data card (in this case, blank) must appear in the deck. For data groups 1 and 4 which are not airport-specific, any non-blank character will indicate standard input. For the remaining data groups, the user who desires to use standard input must specify the airport in the airport file for which he wishes to extract the data. Airports are referred to by their 3-character code names (JFK for New York's John F. Kennedy International Airport, for instance). If the particular airport desired is not in the file, the preprocessor will print a warning message and discontinue processing. The user may request data for different data groups to be drawn from different airports. However, he should ensure that the different airports have compatible sets of data: the numbers of runways and departure fixes must be the same. Also, if the user provides a non-standard aircraft type description, he should realize that standard airport data which depend on aircraft type (such as the time to fly from handoff to the outer marker) have values referring to the standard 5 types of aircraft. The general point to be borne in mind is that the six data groups are not fully independent, so that care must be taken to avoid inconsistencies between groups.

The next subsections of this section will describe in greater detail the data contained in each of the six data groups. The formats for the input, and sample input decks, are given in Appendix A. The listing of the preprocessor code appears in Appendix C.

3.2.1 Data Group 1 - Aircraft Type Description

This data group provides the values of variables which characterize the aircraft types involved in a particular simulation. The standard input values were given in the previous chapter (Figure 2.3.1). The 5 standard types of aircraft are described in detail in the Airport Capacity Handbook (pp. 2.10 to 2.13), and depend mainly upon weight and means of propulsion. These types may be categorized roughly as follows:

(1) type A - large jets

(2) type B - medium jets and large propeller

(3) type C - small jets and medium propeller

(4) type D - light twin engine

(5) type E - single engine

If the user elects to provide his own aircraft type data, he must stipulate landing and liftoff speeds (in knots), runway occupancy times on landing and also on takeoff (in seconds), and possession or non-possession of DME for each type of aircraft. The number of types is flexible, but the current preprocessor program limit is 10. This limit is parametrized and may easily be increased with a single program change. Note that data groups 2, 5, and 6 all refer to information which pertains to aircraft type. The number of types of aircraft and the basic type information are provided by data group 1, with which the information contained in the other groups must conform. This is the responsibility of the user; the present program does not perform such consistency checks.

The unit of time used in the simulation is the hour. Therefore the runway occupancy times (given in seconds) are immediately changed to hours by dividing by 3600.

An additional variable must also be provided by the user electing to use non-standard aircraft-type input: the speed (in knots) at which a landing turns off the runway. This is used in the simulation when calculating the time at which an aircraft reaches an intersection of runways. From touchdown until turnoff, an aircraft is assumed to be constantly decelerating from landing speed to turnoff speed. The calculation of the time to reach any point down the runway, under this assumption, is given in Appendix G.

3.2.2 Data Group 2 - Mix of Aircraft Types

The second data group which the user may specify, if he so desires, is the mix of aircraft types. For each type of aircraft, he must give the fraction of the landings which are of that type and also the fraction of takeoffs. The fractions for landings must add to one, and similarly for takeoffs. If one or both does not, an error message will be printed, but processing will continue. Since the simulation model will need the cumulative distributions, the fractions are immediately converted to those distrubutions in the preprocessing program. If the user elects to accept standard data, he must specify the airport whose mix he wishes to use. The standard mixes all refer to the standard five types.

3.2.3 Data Group 3 - Landing and Takeoff Rates, by Hour of Day

This data group provides the distribution over time of aircraft entering the system. There are two distributions, one for landings and one for takeoffs, since the time-of-day patterns may be different for the two. The user who wishes to provide non-standard data must input the desired number of landings and a number of takeoffs for each simulated hour, up to a maximum of 24 hours. The preprocessor is presently set up to handle up to a day's worth of data. The simulation can be run for more than 24 hours, but the landing and takeoff rates will then repeat in a 24 hour cycle. If the simulation is to be run for less than 24 hours, the user must remember that the system is preloaded for between one and two hours before it actually starts recording delay and throughput (recall section 3.2.1). The user must provide landing and takeoff rates for the preload time period as well as the period for which delay and throughput are to be recorded. The simulation needs the average time between landings or takeoffs rather than the number of them per hour and so the preprocessor estimates the mean time between landings or takeoffs as the reciprocal of the number per hour. If there are no landings (or no takeoffs) in an hour, the time between landings is set at 9.999 hours, so there is a finite but very small probability of generating a landing. The user must specify the airport from which the landing and takeoff rates are to come. The input data tape will be searched for this information if this airport is not the same as the one specified for data group 2, or if data group 2 was user-specified.

3.2.4 Data Group 4 - Separation Requirements

This data group provides numerical values for the separation rules which are to apply to a simulation run. There are three types of separation rules: (1) interlanding, (2) landing following takeoff, and (3) intertakeoff. The fourth possibility, a takeoff following a landing, is governed by the "no two aircraft on the same runway at the same time" rule. The takeoff may taxi onto the runway and start its roll as soon as (but not before) the preceding landing has turned off the runway. The presently required interlanding separation is 3 miles.[3/] In general, all landings must maintain this separation, unless the two aircraft are to land on parallel runways separated by at least 5,000 feet.

The rules defining the separation between departing aircraft are much more complicated, and depend on the paths the departure will take. These rules for IFR aircraft are given on pages 96 and 97 of Terminal Air Traffic Control.

[3/] All separation rules are expressed in nautical miles.

The rules are approximated within the simulation by the following scheme: Each takeoff is assigned a departure path when it is generated. The simulation uses two arrays which determine the required separation between aircraft, one for departures on the same runway and one for those on different runways. These arrays contain for each pair of departure paths, a time separation between start of roll for aircraft outbound on those two paths. Our present procedure (just described) is only an approximation to the rules cited above in that (a) the separation is not made to depend on aircraft type, and (b) the separation requirement is not extended into the airspace beyond the airport.[In addition, one of the present separation rules is in terms of distance rather than time. The user may approximate this by calculating a weighted (by the mix of aircraft type) average of the time for an aircraft to fly the required separation distance (3 miles).]

The standard values for these arrays are: 1 minute for aircraft flying different paths and 3 minutes for those flying the same path, for takeoffs from the same runway; and 2 minutes for aircraft taking off from different runways but then following the same departure path. No separation rule is imposed on aircraft departing on different runways for different departure paths. (The representation scheme for airport configurations to be described in section 3.2.5 below will include a method of indicating to which pairs of runways the inter-takeoff separation must apply.)

The final type of separation is that required between a takeoff and a following landing. The separation depends on whether or not the landing possesses DME. If a landing does possess DME then it must remain a required separation distance (presently 2 miles) behind the takeoff. If the aircraft does not have DME, however, it must not pass a certain fix (called the departure/arrival fix) before the takeoff starts its roll. In our program, the standard value for the departure/arrival fix is 4 miles from the end of the runway.

3.2.5 Data Group 5 - Airport Description

This data group describes the airport's runway configuration and operating policies. The user who wishes to use non-standard data must specify for each runway a two-character heading (00 is north, 13 is 130° from north or roughly southwest, etc.), and a two-character modification if desired. The latter can be used to designate the left (L) and right (R) members of a pair of parallel runways. The four characters (heading and modification) are needed only by the preprocessor, and are used to label the runways on the preprocessor output.

The user must also provide both the distance (in nautical miles) from the runway threshold to the outer marker and the operation code for each runway. The code specifies which of four different operating policies governs the runway: (1) takeoffs only, (2) landings only, (3) dual use, alternate operations, and (4) dual use, landings take precedence. The user may pick any one of these policies for any runway, but he should make sure that the policy agrees with the distribution of runway use in data group 6.

For each aircraft type, the user must provide the minimum time (in minutes) to fly from handoff to the outer marker. (The preprocessor immediately converts the time units to hours, the time unit needed by the simulation.)

In describing the runway configuration, the user must provide for each pair of intersecting runways the distance (in feet) from the end of the first runway to the far side of its intersection with the second, and the distance from the end of the second to the far side of its intersection with the first. These distances are immediately changed to their equivalents in nautical miles.

The model does not need to have the actual layout of the airport, but it _does_ need to know how the runways can interfere with one another. The user may specify "interference code 1" which means that landings on the two runways must be separated by the same distance as are landings on one runway, but that takeoffs are independent of landings and of other takeoffs. If the user specifies code 2, then landings on the two runways must be separated by the same distance as landings on one runway, a landing on one must be separated from a previous departure on either runway, and takeoffs on the two runways must be separated by the special separations required of takeoffs on dependent runways. If the user does not specify a code for any pair of runways, it is assumed that operations on the one do not affect operations on the other. The preprocessing program uses this information to create the arrays which are used by the simulation to tie up appropriate points for the appropriate lengths of time.

3.2.6 Data Group 6 - Runway Use and Departure Fix Distributions

The user who wishes to provide his own input for this data group must supply, for each type of aircraft, the fraction which uses each runway for takeoffs and the fraction which uses each runway for landing. He must also specify for each runway, the fraction of takeoffs using each departure path. The program checks that the fractions sum to one, and prints an error message if not. In addition, it checks that a runway which is to be operated "takeoffs-only" does not have any landings using it, and that a runway being operated as "landings-only" does not have any takeoffs using it. Here too inconsistencies lead to the printing of error messages. Processing continues in spite of error messages.

The preprocessor next computes some other arrays used in the simulation model. The "latest operation on each runway" is initialized as a landing, unless the runway handles takeoffs only. The minimum time a takeoff must remain in its queue is computed as a weighted average (weighted by aircraft type) of the times to fly from handoff to the outer marker The minimum time between when a takeoff may be scheduled and when it actually starts its roll is computed as a weighted average (weighted by aircraft types) of the time for a landing to fly down the final approach path from outer marker to runway threshold. These two times can be interpreted respectively as (a) the minimum time between filing a flight plan and leaving the gate, and (b) the minimum taxiing time from the gate to the end of the runway. However, as can be seen from the way they are computed, their primary purpose in the model is to ensure that takeoffs are scheduled about the same time ahead of start of roll as landings are ahead of touchdown. A further discussion of these arrays is given in the next section.

3.2.7 Simulation Input

The main function of the preprocessor program is to provide an input deck for the simulation model. Every SIMSCRIPT program requires an "initialization deck" which gives the number of each of the permanent entities and initial values for the system variables and arrays for the current run. The DELCAP model has five permanent entities; two of these- the number of types of operations (2) and the number of hours in a day (24)- remain constant for all runs, while the other three - the number of runways, the number of types of aircraft, and the number of departure paths - can vary from run to run. The values of the preprocessor input groups, whether standard or user-specified, are written out in the format required by SIMSCRIPT.[4/] The arrays which store the simulation output delays and throughputs are initialized at zero. The preprocessor also provides one "exogenous event" card image on the simulation input tape. This card schedules the BEGIN event which starts the simulation. If the user is providing deterministic flight input, then he must also provide this card image as the first exogenous event.

[4/] See pages 115 to 128 of <u>SIMSCRIPT,A Simulation Programming Language</u> and Appendix D.

3.2.8 Preprocessor Output

The preprocessor prints out information to label a particular run. The current print-out is not complete, but has provided adequate labeling for the debugging runs of the current effort. Future running of the model in an airport-planning environment should yield a better idea of which data items would constitute the best labeling.

Sample preprocessor output appears in Figures 2.3.2 and 2.3.3, to which we now refer again. The first line of output gives the hours the simulation will be run. (Any error warning messages appear immediately following this line.) Next, the landing and liftoff speeds and runway occupancy times are given for each aircraft type. The mix of aircraft types for landing and for takeoff are listed next. At the end, the preprocessor prints a verbal description of the airport configuration, including the appropriate three-character airport identifier. Each runway is listed by number and by name (e.g., 9L), and the type of operation (takeoffs only, landings only, dual use with alternating operations, and dual use with landings taking precedence) is printed. Next, intersections are listed. Finally, interferences are listed. Two runways with the same heading are identified as "parallel". Parallels may be independent, semi-dependent (landings on one interfere with landings but not takeoffs on the other), or dependent (landings and takeoffs on one must be separated from landings and takeoffs on the other). Only semi-dependent and dependent non-parallel runways are listed.

This concludes our description of the preprocessor program. We wish to reemphasize that the particular current forms of the input to this program, and of the printed outputs, were dictated largely by our needs during program debugging. Because the program has not yet been used in any real planning situations, some of its features <u>may</u> prove awkward for users less familiar with the computer. We have conscientiously tried to foresee such difficulties and to eliminate them in advance, but as yet the program is still in the prototype stage. Only its exercise in more of a "production" environment can be relied on to reveal fully such production-use difficulties as may remain, and only after these problems are revealed can they be remedied.

3.3 The Simulation Program

The simulation program is written in the SIMSCRIPT 1.5 programming language, which is designed primarily to aid in programming critical-event simulations. The user only needs to write code for each of the types of events, describing how each alters the status of the system being modeled and how other events depend on this one. The SIMSCRIPT compiler contains several programs which execute the user-designed event routines in chronological order. Storage for temporary entities, and for events which have been scheduled but not yet executed, is dynamically allocated by other SIMSCRIPT routines. The size of arrays and their allocation of storage are computed at execution time, when sets of initial values are read for them. The simulation is started off by an exogenous event, in our simulation the BEGIN event which schedules the routine to generate the first landing flight and first takeoff flight.

Figure 3.3.1 gives a general "flowchart" of the DELCAP simulation model routines. The word "flowchart" is somewhat of a misnomer in the context of a SIMSCRIPT model. The diagram indicates which event routines occur as a result of which other routines, but it does not give the order in which they are actually executed, since this is chronological. (Recall the discussion in section 2.1.) The earlier Figure 2.1.1 gave a flowchart of the DELCAP critical events for a single aircraft. Figure 3.3.1 describes the computer implementation

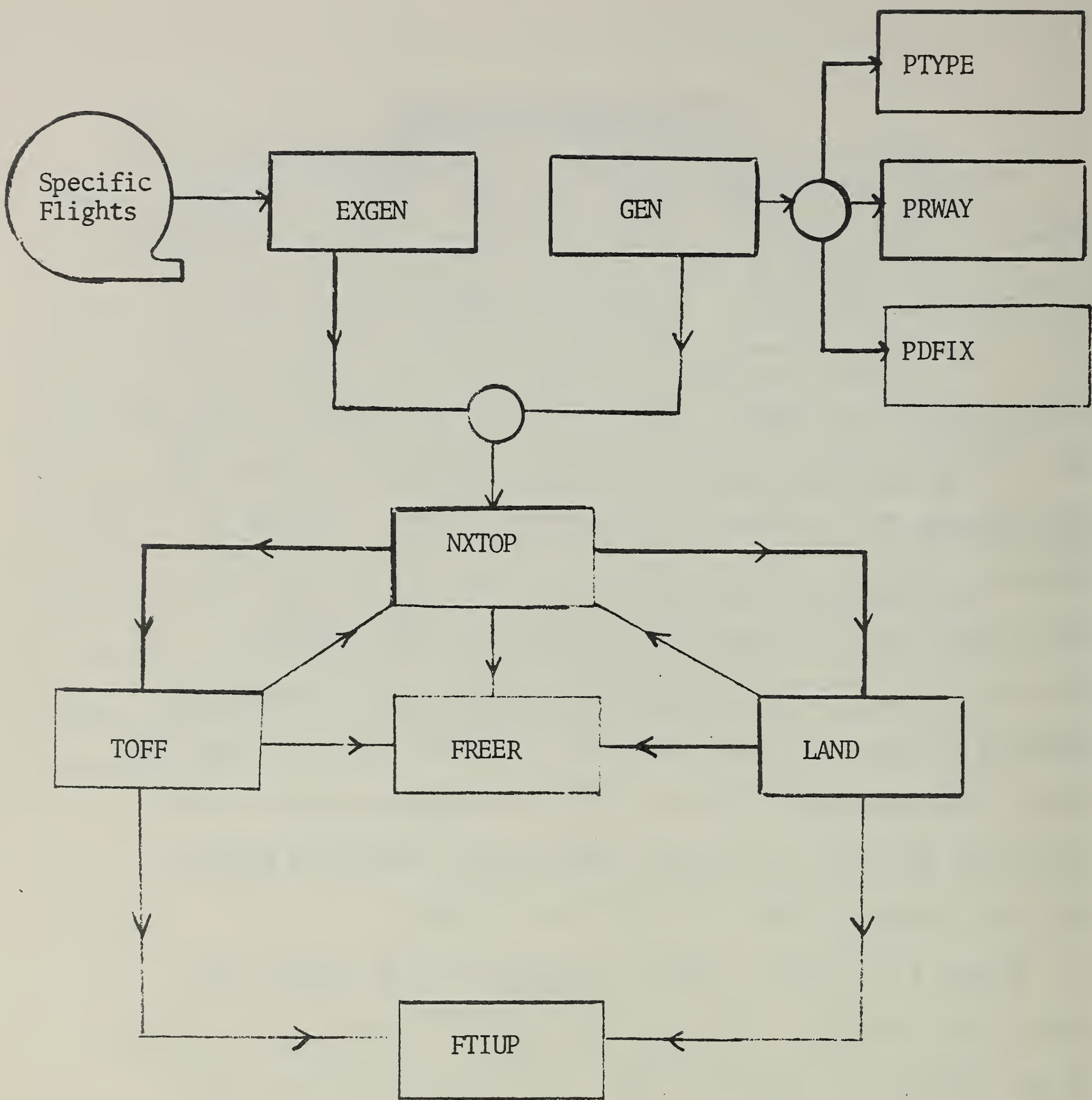

Figure 3.3.1: Flowchart of the DELCAP Simulation Routines

of this process.

Events GEN and EXGEN create flights , which are the units which move through the various events in the model. EXGEN is an exogenous event which occurs at times designated for the arrival into the system of specific flights. GEN creates flights in a stochastic manner. Stochastically generated flights are assigned a type, a runway, and a departure path by the three functions PTYPE, PRWAY, and PDFIX. Flights are constantly entering the system while other events are happening. GEN schedules the next occurrence of itself according to the Poisson process, while the next specific flight (if any are left) for EXGEN is always available. The event NXTOP finds the next operation (landing or takeoff) which is to occur on a particular runway. It is scheduled in one of two ways: (1) if the queue is empty when the current flight is filed in it, or (2) when the current flight has either begun to fly the final approach path to land or has left its gate to take off. Condition (1) is detected in GEN or EXGEN, and condition (2) in LAND or TOFF. NXTOP then schedules the next LAND or TOFF at the time the runway and/or final-approach path is free, as determined by the function FREER. Since there is a time gap between NXTOP and TOFF or LAND during which landings or takeoffs on other runways may have created new tieups for "this" runway, LAND and TOFF again determine the first time the runway is free (from FREER). Then the flight may land or depart, which in the DELCAP model implies tieing up the appropriate points for a period of time sufficient to maintain the required separations. LAND or TOFF then reschedules NXTOP, and the cycle continues. When a tieup is no longer in force, the routine FTIEUP destroys it.

Four routines do not appear in this list, since they are mainly for accounting purposes. The BEGIN event (see Figure 3.3.2) starts the simulation, and schedules the event ENDS which prints the simulation output and stops execution. The routine CHOUR (see Figure 3.3.3) updates the current hour for output of delay and throughput, and reschedules GEN for the Poisson parameter for the new hour. The routine PRINT records the delay and throughput information at touchdown for landings, and at start-of-roll for takeoffs. The sections which follow describe each of the event routines shown in Figure 3.3.1.

3.3.1 Event EXGEN. This event creates exogenously-determined flights provided by the user. This, or the stochastic generation processes or both, may be used for a particular run. When inputting the information for the routine EXGEN, the user must supply for each flight: the hour, minute, and second of entrance into the system, whether this flight is a takeoff or a landing, the runway, the aircraft type, and (for a takeoff) the departure path, all in the format described in Appendix A. The SIMSCRIPT programs read these flights one at a time at the proper simulated time. Therefore, there is no limit on the total number of flights as long as the number simultaneously active (including both those generated by EXGEN and those produced by GEN) is sufficiently small to fit in core. (For a simulation run with 20 runways, 100 aircraft types, and 10 departure paths, there could be about 6,000 flights active at any given time. This, which is permitted in the present model, is far beyond the capacity

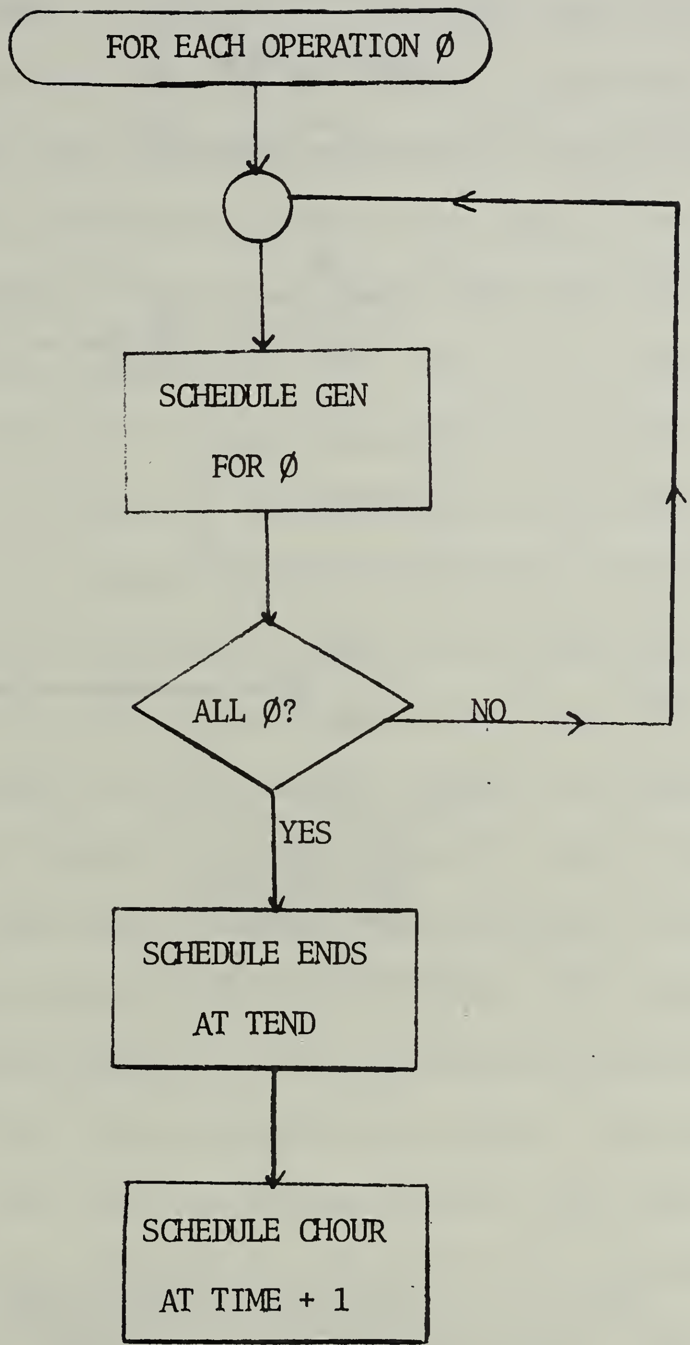

Figure 3.3.2 Flowchart of Event BEGIN

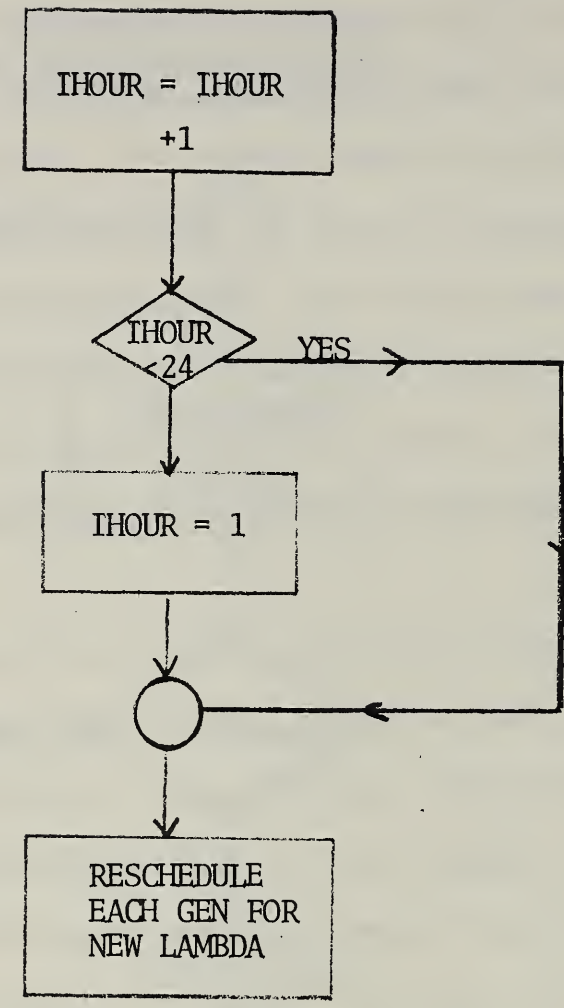

Figure 3.3.3: Flowchart of Event CHOUR

of any existing airport to handle.)

Figure 3.3.4 provides a flowchart for the EXGEN event routine. For a landing, the array TIN stores the time the current flight could (in the absence of other traffic) first cross the outer marker after flying from its hand-off point. A takeoff's flight plan becomes active about 13 to 15 minutes before its scheduled departure. In the model this time period is divided into two segments, so that takeoffs are scheduled about the same time before start of roll as landings are before touchdown. The first of these time segments (about 10 minutes), which may be thought of as representing the time between when the flight plan becomes active and when the aircraft is cleared to leave its gate, is added to the current time and stored in TIN. (The second segment ,about 5 minutes, may be thought of as representing a time interval between when the aircraft is ready to leave its gate and when it could start its roll down the runway; it will be described in greater detail in the section on the TOFF routine.)

After calculating the appropriate TIN, the EXGEN routine files the newly generated flight into the appropriate queue. There are two queues for each runway, one for landing aircraft, the other for takeoffs. The queues are organized in a first-in-first-out manner. This means there is no sequencing by aircraft type; when a slow aircraft precedes a faster one, the latter is not permitted to overtake the former, even if it could reach the outer marker first without thereby delaying the slower plane.

Each flight must remain in the queue until its TIN. Filing

flights into the queue about 10 minutes before they could actually cross the outer marker or leave a gate provides a means for identifying the aircraft type of the flight that follows the current flight. This allows calculation of the proper tieup time to ensure that two aircraft remain separated by the required distance. This distance depends on the speeds of both aircraft involved, and so cannot be calculated until the type of the second plane has been determined.

If the queue was empty before the present flight was added to it, the NXTOP routine is scheduled to occur at TIN, which is the first instant when this flight could be removed from its queue. The NXTOP routine, which schedules the next operation (landing or takeoff) for a particular runway, thus occurs in one of two circumstances: either (1) a landing or takeoff has just occurred, or (2) the runway has been idle but there is now a new flight available for it. Case (1) will be described later in conjunction with the NXTOP, LAND and TOFF routines. In case (2), which is detected in the EXGEN routine, the appropriate queue will have been empty before the flight was filed in it. Therefore the NXTOP routine is scheduled for when the flight is first available to land or take of. However, an earlier NXTOP may have been scheduled in LAND or TOFF, since the other queue may not be empty. In this situation, NXTOP is scheduled, but when it occurs the next operation will already be defined (NEXT ≠ 0) and the NXTOP routine will be terminated. This means that NXTOP may be scheduled more often than necessary. The programming alternative was the coding of a much more complicated set of tests to ensure that NXTOP is scheduled only when necessary. This did not seem warranted, in view of the lack of computer-storage problems and the logical simplicity of the current test.

3.3.2 Event GEN. This event generates flights in a Poisson manner. Landings and takeoffs are generated separately, from two different sets of Poisson parameters. This routine is first scheduled by the BEGIN routine. BEGIN schedules two GEN's, one to create a landing flight and one to create a takeoff. From then on, the GEN routine schedules the next occurrence of itself. Therefore, within GEN we wish to sample from the Poisson distribution to reschedule GEN for the next entry ("arrival") of another aircraft into the simulated system.

The procedure used in the computer for sampling from a distribution is based on the fact that the range of <u>any</u> cumulative distribution is uniformly distributed over the interval [0,1]. In the case here, we have assumed Poisson generation, so the probability of an arrival in a time period of length dt is μdt (plus comparatively infinitesimal terms), where μ is the expected number of arrivals per unit of time. Then the probability $q(T)$ that the next arrival will occur in at most T units of time is

$$q(T) = \text{prob } (t \leq T) = 1 - e^{-\mu T}.$$

Since q is a cumulative distribution, its range is uniformly distributed over the interval [0,1]. We therefore employ a standard computer subroutine to choose a random number R from this uniform distribution, and then find the T for which q(T) = R, namely

$$T = -\lambda \ln (1-R)$$

where $\lambda = 1/\mu$. The next instance of GEN is scheduled to occur in T time units. (Note that our time unit for the simulation is the hour, so λ is the reciprocal of the number of arrivals per hour.) Input to the simulation contains two sets of values for λ for each hour of the day, one for landings and one for takeoffs. As noted earlier, on the hour, each hour, the next GENs, one for a landing and one for a takeoff, are rescheduled according to the λ for the appropriate hour.

In the event EXGEN, the type, runway, and departure path are provided as part of the input. In the stochastic version GEN, however, these three items are obtained by sampling from the appropriate distributions. The simulation is provided (by the preprocessor) with the cumulative distributions of (1) type of aircraft, one for landings and one for departures, (2) runway use by each type of aircraft for landings and also for departures, and (3) departure path for each runway. The three functions PTYPE, PRWAY, and PDFIX perform the sampling processes.

Figure 3.3.5 provides a flowchart of the GEN routine. After rescheduling the GEN routine for the next landing or next departure (depending on the current operation), and sampling to obtain a type, runway, and if necessary a departure path for the current flight,

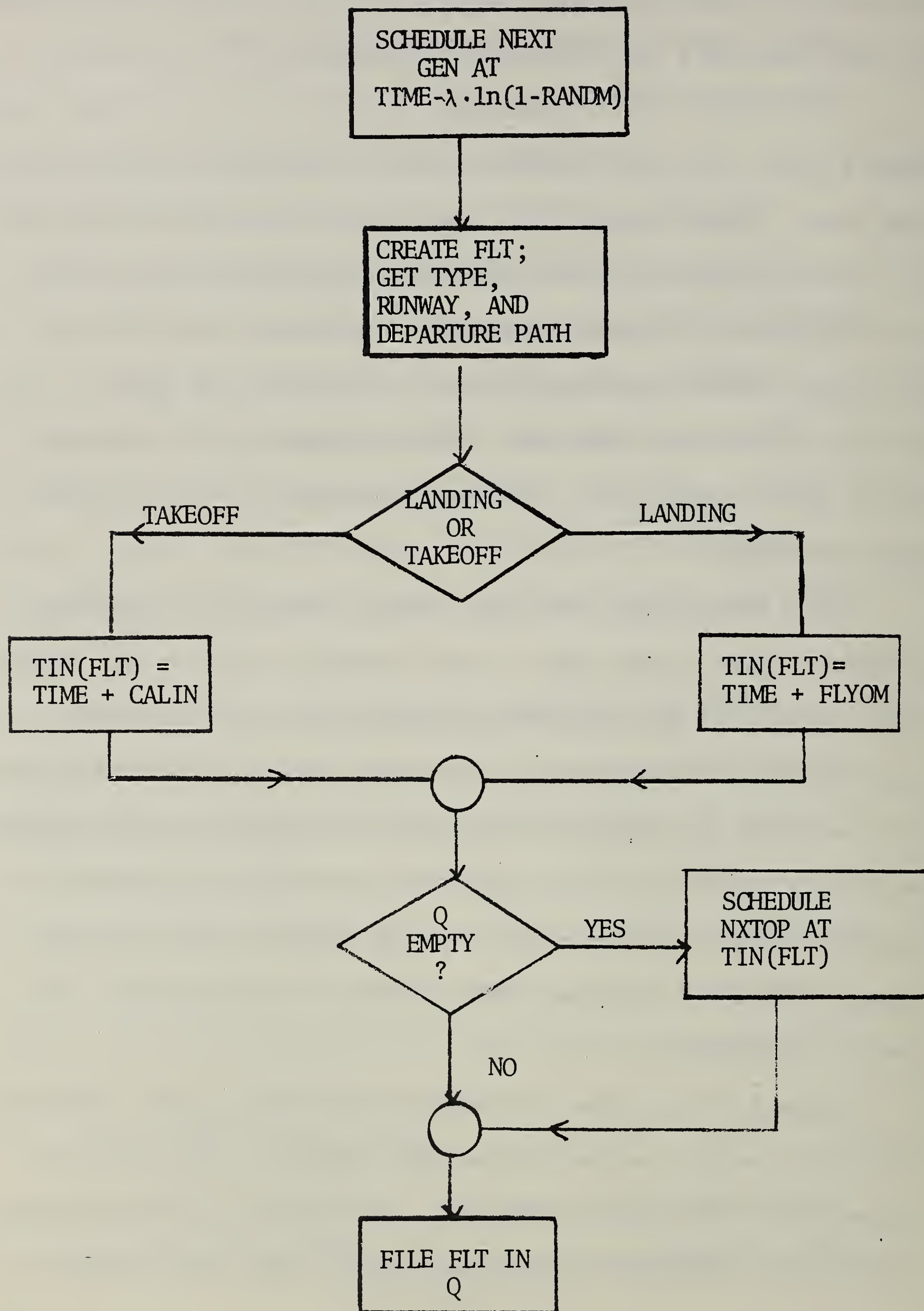

Figure 3.3.5: Flowchart of the Event GEN

the remainder of the routine is the same as for the EXGEN routine. The appropriate value of TIN is calculated, the flight is filed in its proper queue, and if the queue was empty before this flight was filed in it then the NXTOP routine is scheduled at TIN.

3.3.3 Event NXTOP. This routine finds the next operation, landing or takeoff, which will occur on a runway. Figure 3.3.6 provides a flow chart for this routine. There are four possible operational procedures (stored in the variable OPER) available for any runway: (1) takeoffs only, (2) landings only, (3) dual use, alternating operations, and (4) dual use, landings take precedence. For OPER = 1 (takeoffs only) or OPER = 2 (landings only), the sequencing of operations is trivial; since only one type of operation is allowed on that runway, NXTOP only needs to examine the appropriate queue. If it is empty, no landing or takeoff is scheduled and NEXT is set equal to zero. If the queue is non-empty, the appropriate operation is scheduled.

If OPER = 3, then NXTOP tries to alternate operations. The queue for the operation type opposite to the last operation is examined first. Let t_1 be the time the first flight in the queue for this operation could be scheduled. If TIN for this flight is less than t_1, then this operation is the one scheduled. However if t_1 equals TIN, the first flight in the queue for this operation may not be immediately available. Maybe a flight in the other queue would be available earlier. Therefore, the other queue is examined in a similar manner. Let t_2 be the time the first flight in it could be scheduled. If TIN for that flight is less than t_2, it is immediately available, and this operation is scheduled. However if the first flight in neither queue

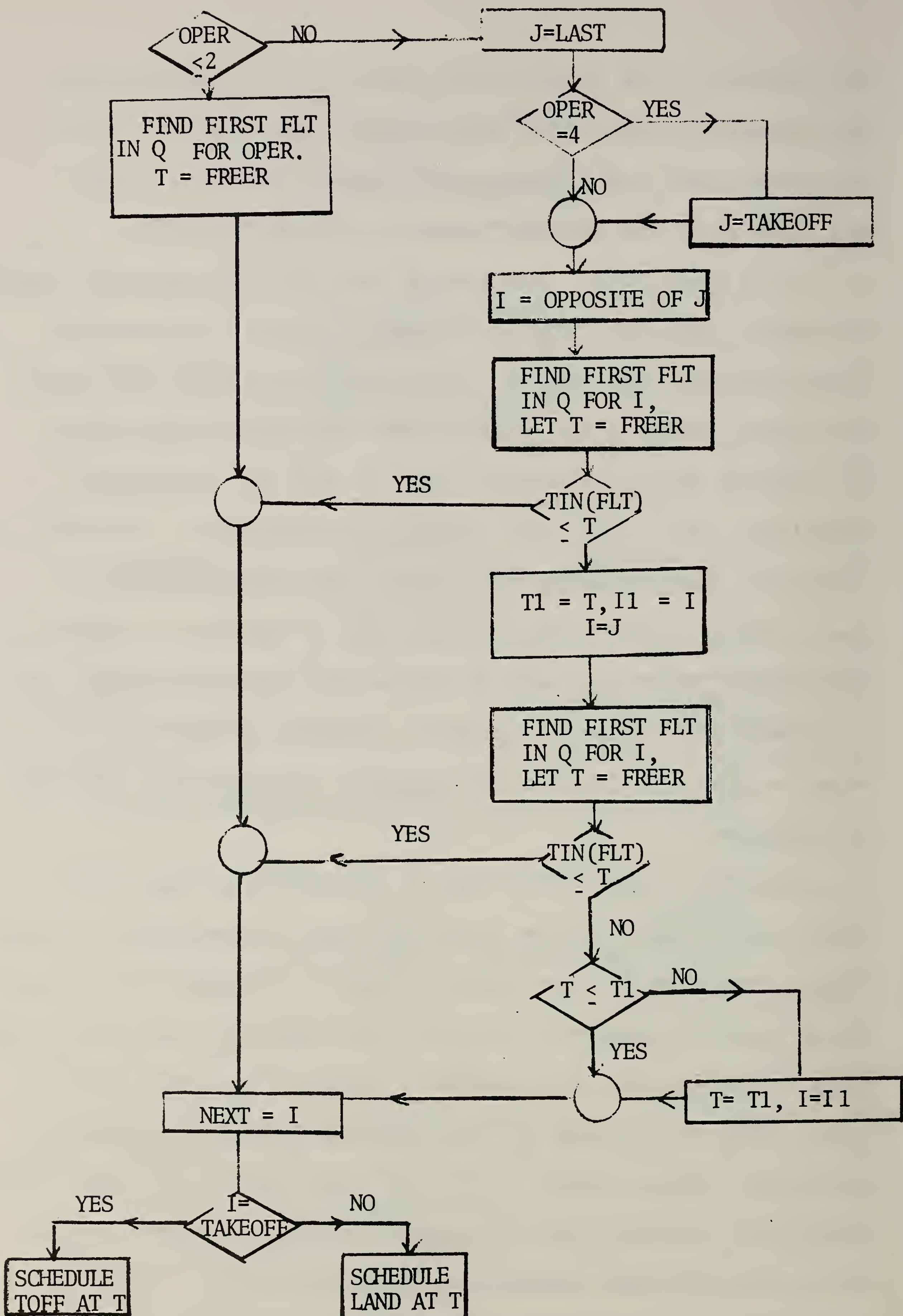

Figure 3.3.6: Flowchart of the Event NXTOP

is immediately available, that operation for which the flight is available <u>first</u> is scheduled. Therefore NXTOP will alternate operations, except when the first flight in the queue for the alternate operation is not immediately available; in that case the operation whose first flight is available soonest is scheduled. If no flights are available, NEXT is set equal to zero and no operation is scheduled.

For OPER = 4 the NXTOP routine will schedule a landing, if the first flight in the landing queue is immediately available. If not, the routine schedules that operation whose next flight is available first. As can be seen in the flowchart Figure 3.3.6, the logic for OPER = 4 is the same as for OPER = 3 except that the landing queue is always examined first.

The NXTOP routine is scheduled in one of two instances: (1) the LAND or TOFF routine has occurred, or (2) a queue was empty and a new flight has just been filed. In the second instance, NXTOP is scheduled for the time TIN at which the flight could first be scheduled. However, since the other queue for the runway need not be empty or a LAND or TOFF routine could just have occurred, another NXTOP may already be scheduled for this runway. To avoid error because of having several NXTOPs scheduled at once, an array NEXT with an entry for each runway has been introduced. Originally it is zeroed. When a next operation for a runway has been found by NXTOP, NEXT is set equal to 1 (for a takeoff) or 2 (for a landing). Then NEXT is zeroed in the LAND or TOFF routine.

Therefore NEXT is non-zero precisely when a LAND or TOFF is scheduled but has not yet occurred. NXTOP proceeds to find a next operation for a runway only if NEXT for that runway is zero. This condition is tested at the beginning of NXTOP, and if NEXT is non-zero NXTOP is immediately terminated.

3.3.4 Function FREER. This function finds the earliest time a particular flight can land or take off without violating the separation rules. FREER is first called in NXTOP, to find the time at which the LAND or TOFF routine should be scheduled.

There may be a time gap between the time the NXTOP routine occurs and the time LAND or TOFF occurs,during which other flights might add new tieups which require postponement of the operation in question. Therefore FREER is called again from LAND or TOFF, to determine when the landing or takeoff may actually occur. Figure 3.3.7 contains a flowchart of the function FREER. The left-hand side refers to landings, the upper right-hand side to takeoffs, and the lower right-hand side to both. T is the maximum of TIN and the current time, used to single out for examination only those tieups affecting the current flight. The array TR is created to contain the time (TMAX) each tieup affecting the flight will no longer be in force, and J is a count of the number of entries in TR.

For landings, both the set of tieups (OMTI) associated with the outer marker and the set (THTI) associated with the runway threshold are examined. The time of tieups in THTI is translated to the outer marker by subtracting off the amount of time it takes the current flight to fly from the outer marker to the runway

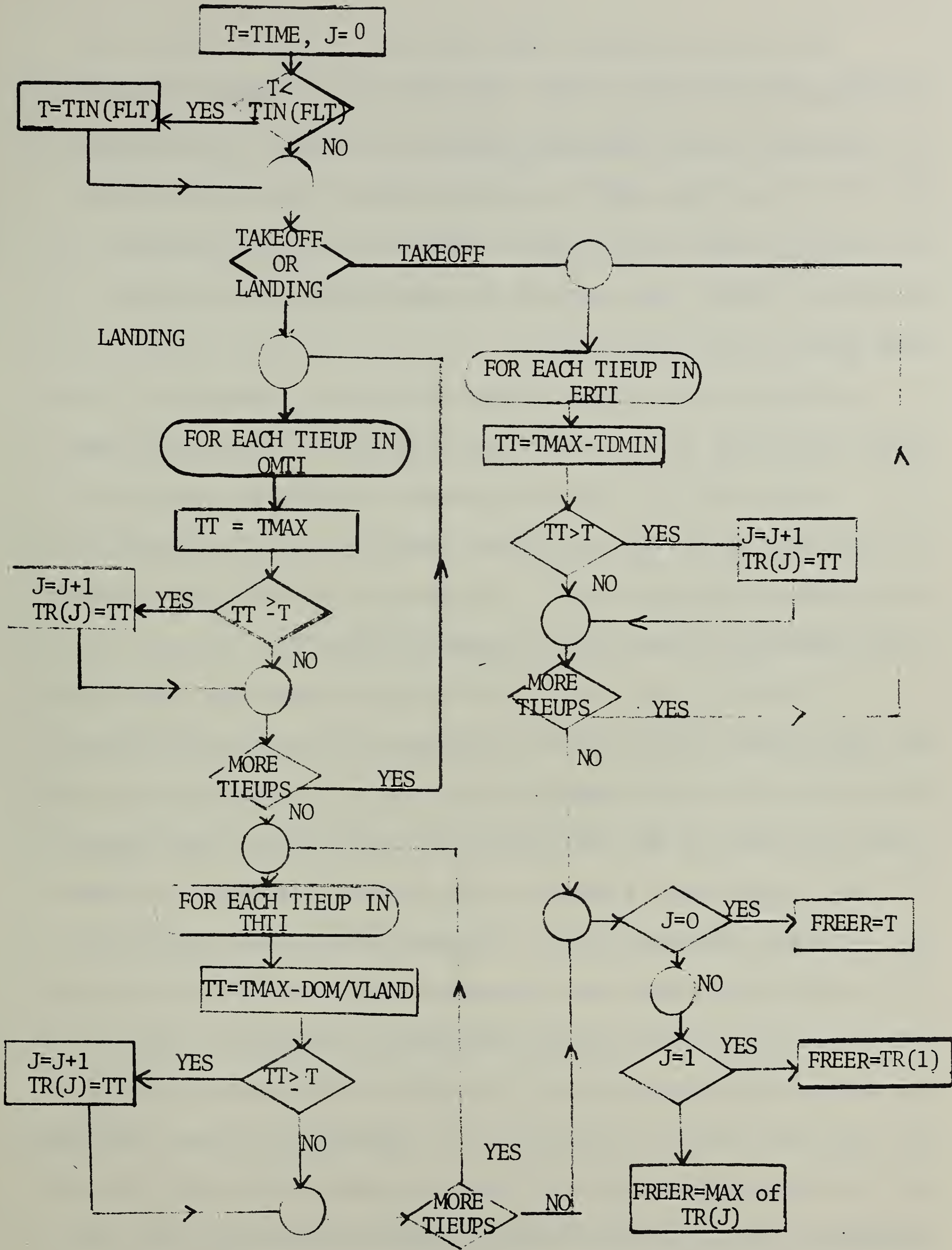

Figure 3.3.7: Flowchart of the Function FREER

threshold; this reflects the fact that the runway threshold need only be free as the current flight gets there, not before. For takeoffs, only the set of tieups (ERTI) associated with the end of the runway are examined. The time of these is translated to the gate by subtracting off TDMIN, since takeoffs are scheduled before they leave their gates to taxi to the runway.

If there are no tieups affecting the current flight (i.e. J = 0), FREER is set equal to T. If only one tieup affects the current flight (i.e., J = 1), then TR (1) will contain the time at which that tieup will no longer impede the start of the landing or takeoff procedure, and so FREER is set equal to it. If several tieups affect the current flight, FREER is set equal to the maximum of the TR's.

It should be clear from the previous description that this routine does not attempt to fit a flight in between two others, even if the gap between the two is wide enough. To do so would require a great deal more coding. The crux of the difficulty is how wide a gap is "wide enough". The tieups occurring as a result of the inserted flight must not affect any previously scheduled flight. This means that all the tieups which LAND or TOFF would create must be examined to see if they would interfere with a landing or takeoff already scheduled or in progress. This is similar to performing the whole of the LAND or TOFF routine, and involves the additional burden of identifying the flight which is being interfered with. (It is no longer just the first in a queue.) Therefore the simpler procedure, of waiting until the last tieup is no longer in force , was used in the DELCAP simulation. Future work should investigate the feasibility of elaborating this procedure.

One further difficulty can arise when a slow landing follows (i.e. lands later than) a fast takeoff. NXTOP is called as soon as the takeoff leaves its gate. The landing therefore is not permitted to cross the outer marker before then, since FREER is at least T which in turn is at least the current time of NXTOP. However, if the landing is slow enough it could in principle be scheduled earlier, and the takeoff would still be able to precede it while maintaining the required separation. Therefore, although the sequence of operations on the runway must be a takeoff followed by a landing, the sequence of routines should really be LAND followed by TOFF. This difficulty has not been resolved, but in sample debugging runs it occurred only about 2 to 3% of the time, and added only about 30 seconds extra delay at each occurrence. Therefore it does not seem to affect the DELCAP results by a significant amount.

3.3.5 Event LAND. The primary purpose of both the LAND and TOFF routines is to tie up the appropriate points in order to ensure that following flights remain properly separated from the current landing or takeoff. Figure 3.3.8 is a flowchart of LAND. LAND removes the first flight from the landing queue for the appropriate runway. Then it calls FREER to find when the runway and final approach path are first free so that this flight may cross the outer marker.

The separation rules which apply to a landing, and their implementation, are discussed below:

(1) No two aircraft may occupy the same runway at the same time. This rule is implemented by tying up the runway threshold (for landings)

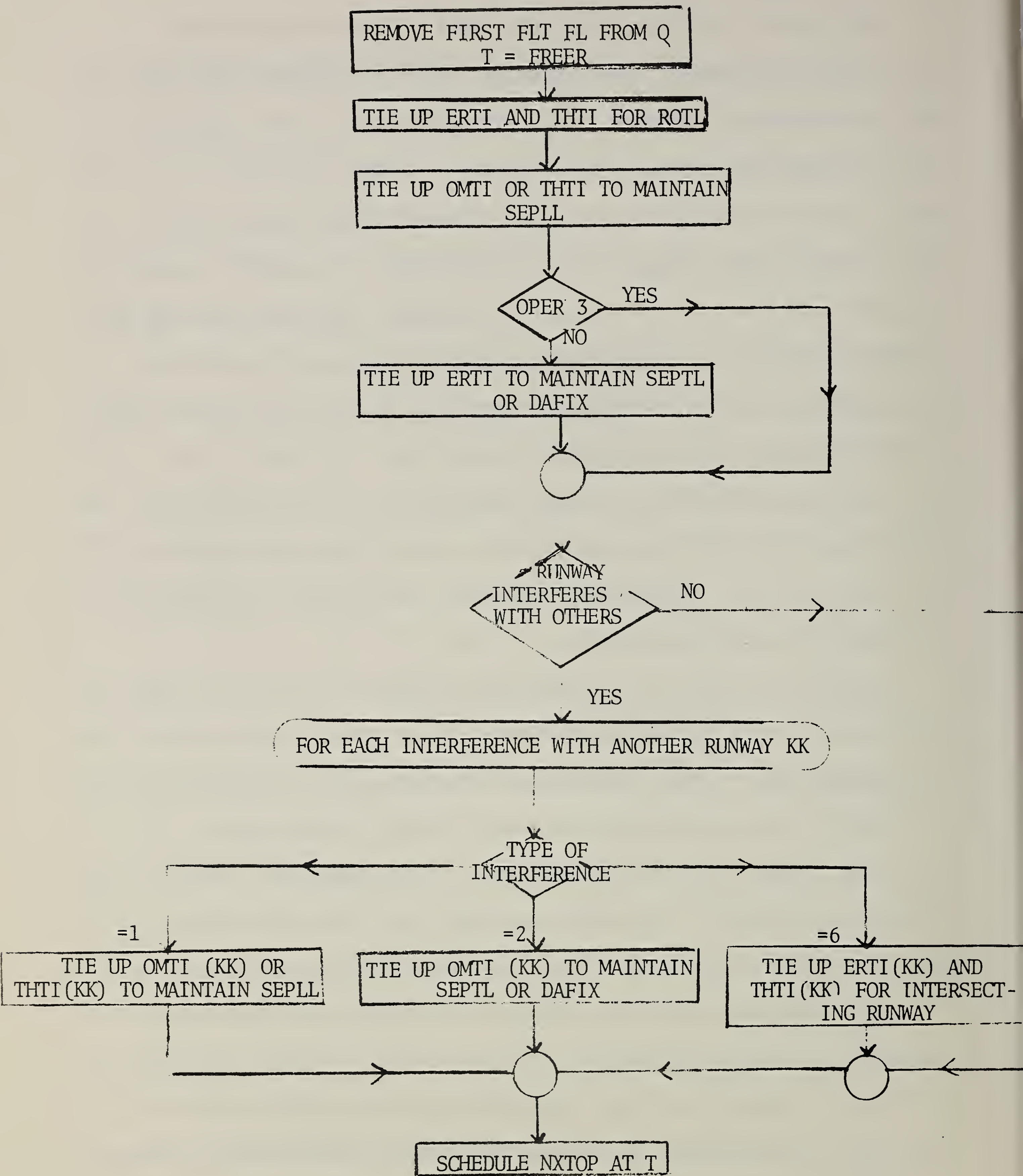

Figure 3.3.8: Flowchart of the Event LAND

or the end of the runway (for takeoffs) for the time the current landing will occupy the runway.

(2) Two landings must be separated by a minimum distance (called SEPLL, in DELCAP). We assume a constant nominal final-approach speed, depending on aircraft type. Therefore, the point at which two landings are closest while always maintaining the required separation depends on the relative speeds of the two planes. If the first is faster, they will be closest when the first crosses the outer marker. In this case the outer marker is tied up for the time it will take the second to fly SEPLL. If the second is faster, the two planes will be closest when the first just touches down. In this case the runway threshold is tied up from touchdown of the first until that time plus the time for the second to fly SEPLL.

(3) A landing must be separated from a preceding takeoff. The separation required depends on whether the landing aircraft has Distance Measuring Equipment (DME). If not, the landing may not pass a certain fix (called the departure/arrival fix) before the takeoff starts. Therefore, if a landing has passed this fix, no takeoff may occur until the landing has cleared the runway, and so the end of the runway is tied up from the time the landing passes the departure/arrival fix until touchdown. (The end of the runway is already tied up, for the period the landing will occupy the runway,by virtue of separation rule (1) above.) If the landing does possess DME, however, then the two aircraft (landing and preceding takeoff) need only be separated by

a required distance (called SEPTL in DELCAP). The standard present value of the departure/arrival fix is 4 miles from the end of the runway, and that for for SEPTL is 2 miles. As noted above, the final-approach speed is treated as constant. Under the assumption of a single constant acceleration for a takeoff on the ground and in the vicinity of the airport, the distance the landing must be from the takeoff when the latter starts its roll is

$$\text{SEPTL} + 0.5\ V^2 \cdot \text{ROTT/S},$$

where V is the speed of the landing, S is the liftoff speed of the takeoff, and ROTT is the runway occupancy time for the takeoff. (This formula is derived in Appendix F.) The end of the runway is therefore tied up from the time the landing passes this point until touchdown time.

Tying up a point is accomplished in the simulation by creating a temporary entity called a TIEUP, with attributes TMIN, the time the tieup goes into force, and TMAX, the time the tieup is no longer in force. The TIEUPs are filed in one of the sets OMTI, THTI, or ERTI, which are scanned in FREER to decide when the runway and final approach path airspace are free. Once the TIEUP is no longer in force, it is removed by the FTIUP routine which is scheduled in LAND as the TIEUP is created.

In addition to tying up points on the same runway, points on interfering runways must be tied up. Two arrays RPT and TPT control these interferences in the DELCAP simulation. For each runway and interference, RPT contains the runway being interfered with, and TPT contains the type of interference. There are six types of interference:

1. Landings on one runway must be separated from landings on the other runway by SEPLL.
2. Landings on the one runway must be separated from preceding takeoffs on the other runway in the same manner as that described in (3) above.
3. Takeoffs on the one runway must be separated from following landings on the other as described in (3) above.
4. Takeoffs on the one runway must be separated from takeoffs on the other runway by the same separation as takeoffs on the same runway.
5. Takeoffs on the two runways are independent if they diverge, but must be separated if they do not.
6. The two runways intersect.

Types 1, 2, and 6 apply to landings. Tieups for interference types 1 and 2 are computed in a manner similar to (2) and (3) above. For intersecting runways, a takeoff or landing on another runway may not be on the runway between the time the current landing touches down and the time it passes the intersection or turns off, whichever occurs first.

The time for an aircraft to travel from touchdown to an intersection a distance D from the end of the runway is

$$(1/A)(-v + \sqrt{v^2 + 2AD})$$

where v is the landing speed and A is the acceleration of the landing. We assume A is constant, so

$$A = (v_1 - v)/ROTL < 0,$$

where v_1 is the turnoff speed of the landing, v is the final approach speed, and ROTL is the runway occupancy time. This formula is derived in Appendix G.

The RPT and TPT lists are scanned, and appropriate tieups are initiated to maintain required separation between the current landing and operations on other runways. As each tieup is created, it is filed into the set for the point being tied up. At this same time, an FTIUP is scheduled to destroy the tieup once it is no longer in force.

Once all the necessary tieups have been created. the LAND routine sets NEXT = 0 and schedules NXTOP for the time the current landing crosses the outer marker. Then the delay to this flight is calculated as the difference between the time it crosses the outer marker, and TIN (which is the first time it could cross the outer marker were there no other aircraft present). The PRINT routine is scheduled at the touchdown time for this landing. PRINT adds the delay to this flight to the total delay, and increments the number of landings for the correct hour. Since all tieups to maintain separation from this landing have been created and since the delay for this flight has been calculated, the flight is no longer needed, so it is destroyed. This completes our description of the landing routine. The takeoff routine performs similar tasks related to takeoffs.

3.3.6 Event TOFF - Figure 3.3.9 is a flowchart of the TOFF routine. Much of it is similar to the LAND subroutine. The first flight is removed from the landing queue and FREER is called to ascertain the first time the flight can taxi to takeoff. Tieups are created to maintain separation, both on the same runway and on others where there is interference. NXTOP is scheduled for the time specified by FREER, the delay is calculated, and the flight is destroyed. Thus the overall structure of TOFF is similar to that of LAND.

Takeoffs, however, are special in one way. They enter the system about 15 minutes before scheduled takeoff. The TOFF routine occurs about 4 minutes before takeoff. The reason for this early scheduling of takeoffs can best be described here, in the context of the TOFF routine. Takeoffs are scheduled about 4 minutes early so that scheduling of takeoffs is compatible with scheduling of landings. Landings need to be scheduled before touchdown, since they must be properly separated from other operations along the whole of the final approach path. If takeoffs were scheduled only at start-of-roll, a following landing could be scheduled no earlier than that start-of-roll. In other words, the following landing could not cross the outer marker until the preceding takeoff had started its roll. It would greatly complicate the model if landings and takeoffs for one runway were scheduled in an order different from that in which they occur in LAND and TOFF.

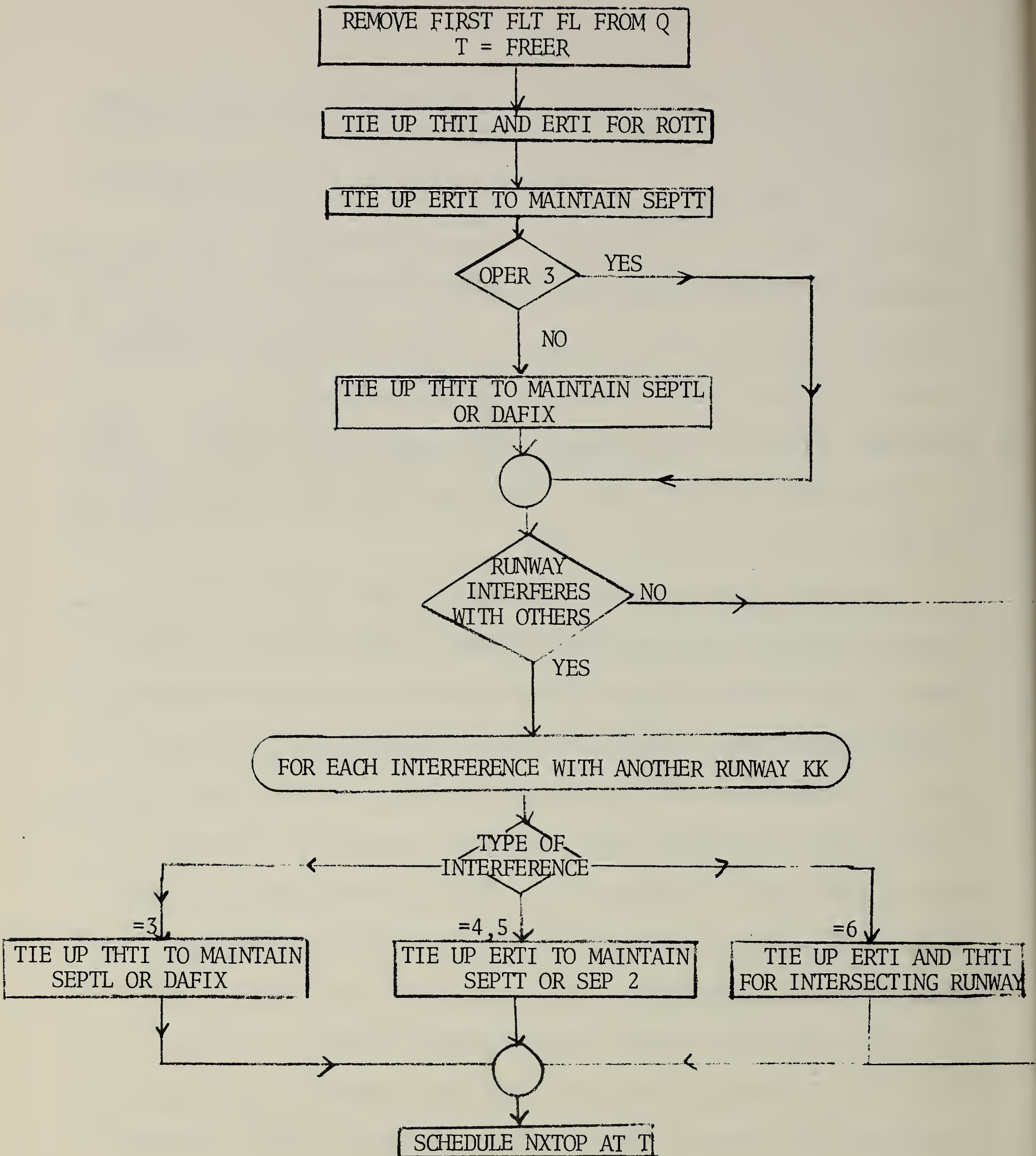

Figure 3.3.9: Flowchart of Event TOFF

By scheduling takeoffs early, landings and takeoffs can be treated in the same manner. As noted above, there is still a residual difficulty when a very slow landing follows a fast takeoff, but for the most part, scheduling takeoffs early permits proper sequencing and scheduling on a dual-use runway.

One may still ask, "why generate takeoffs 15 minutes early?" The figure 15 is of course somewhat arbitrary, but it is necessary to generate takeoffs at least 3 to 5 minutes (depending on the separation rules) before they are scheduled by TOFF. When calculating the tieup duration needed to maintain separation from following aircraft, it is necessary to know the departure path and/or type of the following aircraft. If the second takeoff will immediately diverge from the path of the first, for instance, they need only be separated by 1 minute. Otherwise they need to be separated by a greater time interval. Therefore takeoffs have to be generated as far ahead (in time) of scheduling in TOFF as the greatest time separation required between aircraft.

The careful reader may wish to inquire whether this procedure is indeed not too artificial. We note in response that these time intervals can be interpreted in terms of real events. The 15-minute interval may be thought of as the minimum time ahead of departure at which a flight plan can be filed. Such a minimum time is in fact required at the more congested airports, and as more terminals become congested this practice will become more widespread. Also, with the addition of computer processing of flight plans, a minimum filing time is quite likely. The 4-minute time between scheduling and takeoff may be thought of as the time for the aircraft to leave its gate, taxi to the runway, and complete final checkout. In the model, queuing for takeoff would then occur before leaving the gate, although at most terminals gate space is limited and there are parking ramps for waiting. This is another instance of a situation in which we are interested in the length of a time interval but not in where the aircraft is during that interval. We would be interested in where the aircraft actually is only if this were to affect whether the aircraft could turn onto the runway when the runway is free. The DELCAP model does not include, in its present form, any ground operations. Therefore, the delay figures do not include delays incurred during ground operations. Future model modifications might address this additional source of delay.

To return to our discussion of the TOFF routine, we will now describe the separation rules applying to a takeoff and their implementation.

(1) No two aircraft may simultaneously occupy the same runway. This rule is implemented in the same manner as it was in LAND. The runway threshold and the end of the runway are tied up from start of roll to liftoff.

(2) Separation between departing aircraft depends on the departure paths being followed by those aircraft. The precise rules governing separation between takeoffs are contained in Terminal Air Traffic Control, pp. 96-97. The required separation depends on the distance to the point of divergence of departure paths. The separation is implemented in the model through the use of an array SEPTT which depends on the two

departure paths involved. For every pair of departure paths, SEPTT contains the separation time required between start of roll for two aircraft bound along those paths. All the separation rules are stated in terms of time separation, except for one distance separation. In the present scheme, this rule has to be approximated by a weighted average of the times for takeoffs of various types to fly the required separation distance. The standard values for SEPTT from the preprocessor are: 1 minute for aircraft on different departure paths, and 3 minutes for those on the same path. This assumes that different departure paths diverge immediately. The user may, of course, specify his own SEPTT array depending on the departure routes for the particular terminal he wishes to study. Separation between departures on different runways will be discussed below.

(3) A takeoff must be separated from a succeeding landing. The process here in TOFF is similar to that described in separation (3) of LAND. If the landing has DME, the runway threshold is tied up from start of roll until that time plus

$$(1/S)(SEPTL + 0.5\ S^2\ ROTT/V)$$

where S is the landing speed, V is the liftoff speed of the takeoff, and ROTT is the runway occupancy time of the takeoff. If the landing does not have DME, the runway threshold is tied up from start of roll of the takeoff until the landing could have flown from the departure/arrival fix to touchdown.

Each tieup created is filed in the appropriate set ERTI, for the end of the runway, or THTI for the runway threshold. Along with each tieup, the routine FTIUP is scheduled for when the tieup is no longer in force.

TOFF also ties up points on interfering runways in order to ensure that the required separation from the current takeoff is maintained. Of the six types of interference listed in the description of the LAND routine, four pertain to takeoffs:

3. Takeoffs on one runway must be separated from following landings on the other runway.

4. Takeoffs on one runway must be separated from takeoffs on the other runway by the same time as takeoffs on the same runway.

5. Takeoffs on the two runways are independent if they diverge, but must be separated if they do not.

6. The two runways intersect.

Tieups for types 3 and 4 for different runways are computed in the same manner as separations (3) and (2) above for one runway. Tieups for type 5 are computed in a manner similar to that of separation (2) above, except that a second array SEP2 is used instead of SEPTT. SEP2 contains time separations required between aircraft on different runways which take the same departure path. Type 6 is handled for takeoffs in the same manner as for landings. The threshold and end of the second runway are tied up from the time the takeoff starts its roll until it has passed the intersection.

The remainder of the takeoff routine is the same as the landing routine. NXTOP and PRINT are scheduled, delay is calculated, and the flight is destroyed.

3.3.7 Event FTIUP - This event destroys a tieup as soon as it is no longer in force. A flow chart appears as Figure 3.3.10. These "erasures" free computer storage for new flights and tieups, and make searching the sets in FREER easier. Since the sets OMTI, THTI, and ERTI are ordered by TMAX (the time the tieup is no longer in force), FTIUP only needs to remove and destroy the first tieup in the appropriate set.

This concludes our routine-by-routine description of the simulation program. We will now describe the input required and output produced. Any SIMSCRIPT program requires a "system specifications card" and an initialization deck specifying the number of permanent entities and the size and initial values of each of the main variable and arrays. The preprocessor provides the initialization deck. A user who prefers not to use the preprocessor (and so must prepare his own initialization deck instead) should refer to SIMSCRIPT, A Simulation Programming Language, pp. 115 to 128, for a further description of the initialization deck and its format. We will describe the system specification card here, though, since it must be provided by the user:

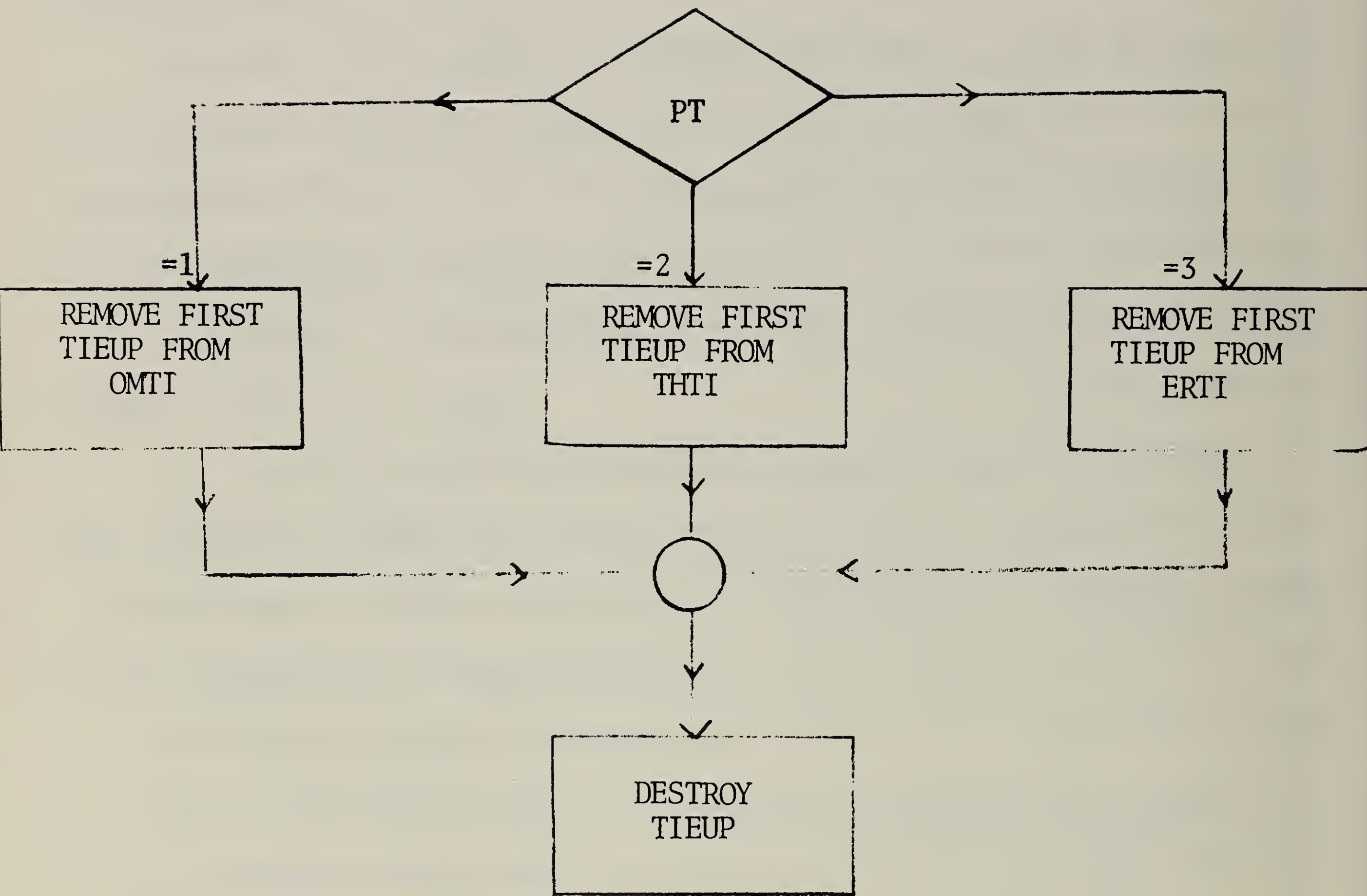

Figure 3.3.10: Flowchart of the Event FTIUP

Columns	Contain
1	1 (one)
6	P - if the user desires printing of the initialization deck. Otherwise, leave blank.
11-12	50
17-18	60 } time unit is hours
23-24	60 }
36	8 (number of tape unit containing initialization deck)
42	8 (number of tape unit containing exogenous events). May be another unit if EXGEN is used, but first event must be BEGIN.

In addition to the initialization deck, the user may provide deterministic traffic for input to the EXGEN routine. This input must be in SIMSCRIPT exogenous-event input format. The first exogenous event must be BEGIN (event type 1), and the rest may be EXGENs (event type 2), The format required is given in Appendix A.

Sample simulation output appeared in the earlier Figure 2.2.1. It includes the number of operations generated, the number performed, total delay, and average delay per aircraft broken down by hour-of-day and by operation. Other outputs are in principle readily available, but since the simulation has never been run in other than a debugging environment, i.e. to determine which outputs are operationally preferred, the current set is all that the present coding provides. Queue statistics, such as average length of queue, could easily be gathered.

Any of the statistics could be gathered by runway or by aircraft type. The present outputs should therefore be regarded as illustrative of model capability, and not as the only ones available.

This concludes our description of the model programs. The next chapter will describe running the model, and outputs from illustrative runs.

4. OUTPUTS FROM ILLUSTRATIVE RUNS

During the course of the current work, the DELCAP model has been run primarily to test its components. The results of several of these runs will be presented here. It should be noted that the scope of the work to date limited "data collection" to use of easily available published data; we were unable even to get a complete set of data for any one airport, and the present phase of this model development effort included neither validity checks nor sensitivity analyses. Therefore, the reader should view the "results" given in this chapter as examples of model capability, the outputs of model-testing runs, rather than of "application" or "production" runs.

We will report three sets of computations. The first describes a parallel runway configuration under three different operational procedures. The second is the 24-hour run based on Atlanta Airport (ATL) which was used as an example in Chapters 2 and 3. The third is a run based on the John F. Kennedy International Airport (JFK). The discussion of each case will be accompanied by a specification of the associated input parameters.

The parallel runways for the first set of runs are assumed to be between 3500 and 5000 feet apart, which under current rules means that a landing on either runway must be separated from all other landings. Takeoffs on one runway do not interfere with landings or takeoffs on the other. Three different operational procedures define the three cases studied in this set of runs:

	Runway 1	Runway 2
Case 1:	Takeoffs	Landings
Case 2:	Both	Landings
Case 3:	Both	Both

The traffic mix and types used for all three cases are given in Figure 4.1. The same stream of traffic is generated in the three cases.

In Case 1, all takeoffs were assigned to runway 1 and all landings to runway 2. In Case 2, 50% of the landings were assigned to each runway but takeoffs only to runway 1. In Case 3, 50% of the takeoffs and landings were assigned to each runway. Other than this, the only difference among the three runs was the operational procedure. In Case 1, OPER = 1 on runway 1 and OPER = 2 on runway 2. In Case 2, OPER = 3 on runway 1 and OPER = 2 on runway 2. In Case 3, OPER = 3 on both runways. (Recall that OPER = 3 means that landings and takeoffs are alternated whenever the appropriate operation can take place immediately; otherwise the operation which can take place first, does take place first.) The outputs of the runs refer to a one-hour period during which the expected numbers of landings and takeoffs are each 30. The simulation was run for one hour previous to recording output in order to preload the system.

Results for the three runs are tabulated in Figures 4.2 and 4.3. Since all landings must be separated by the same 3 miles, permitting landings on both runways could increase throughput only if runway occupancy time were a more critical factor than a 3-mile separation rule. A glance at the data in Figure 4.1 shows that runway occupancy time for either a landing or a takeoff is less than the time for any

TYPE	DESCRIPTION	MIX %	SPEEDS (KNOTS) LANDING	LIFTOFF	RUNWAY OCCUPANCY (SECONDS) LANDING	TAKEOFF
A	Large Jets	6	165	175	59	37
B	Medium Jets and Large Propeller	50	150	160	52	34
C	Small Jets and Medium Propeller	4	135	145	45	31
D	Light Twin Engine	20	120	130	38	28
E	Single Engine	20	105	115	31	24

Figure 4.1 Aircraft and Traffic Description for Run Set I

NUMBER OF OPERATIONS PERFORMED

	LANDINGS	TAKEOFFS	TOTAL
CASE 1	32	28	60
CASE 2	30	11	41
CASE 3	28	27	55

Figure 4.2: Throughputs, First Set of Runs

AVERAGE DELAY PER AIRCRAFT (MINUTES)

	LANDINGS	TAKEOFFS	ALL
CASE 1	11.25	8.74	10.08
CASE 2	13.19	50.03	23.07
CASE 3	17.09	19.22	18.14

Figure 4.3: Mean Delays, First Set of Runs

landing aircraft to fly 3 miles, so that the separation rule is a more stringent requirement than the rule prohibiting two aircraft on the same runway. Hence, no increase in throughput can be gained by allowing landings on both runways. This is illustrated by the results in Figure 4.2, where the number of landings actually decreased when they were allowed on both runways. The decrease stems from the fact that landings must be separated not only from other landings, but from takeoffs on the same runway as well. In Case 1 there are no takeoffs on the same runway as landings, so this rule never comes into play. In Case 2, half the aircraft land on runway 1 and so interact with the takeoffs on that runway. In Case 3 all landings interact with takeoffs and so the landing throughput is cut still further.

In Case 2, the number of takeoffs is drastically reduced. Since takeoffs can only occur on runway 1, and half the landings also occur on that runway, it is not surprising to find the number of takeoffs cut by more than half. When takeoffs are allowed on both runways (Case 3), their number increases to almost its level in Case 1.

Case 1 provides the least delay, as well as the greatest throughput. Since landings must always remain separated by 3 miles from other landings, landing delay can only increase when landings and takeoffs are allowed on one runway. Takeoff delay increases dramatically, to 50 minutes per aircraft, in Case 2 (where only one runway handles takeoffs, and landings also use that runway). When both takeoffs and landings are allowed to use both runways (Case 3), landing delay is increased over Case 2 by about 4 minutes per aircraft, while takeoff delay is reduced about 30 minutes. It is difficult to determine the exact sizes of these delays from the input since they depend on the traffic mix, the runway use distribution, and the distribution of departure paths.

The major conclusion which seems to emerge from this exercise is that -- at least for this particular airport with this particular traffic level -- under the current separation rules, the best on the other. Of course no such rule should be inflexibly imposed on real situations; if the controller is presented with two takeoffs desiring to leave at the same time, and there is a large enough gap in the landing stream, he clearly should allow one of the two takeoffs to use the runway ordinarily reserved for landings.

An additional observation from the preceding analysis is that the distribution of departure fixes all can have a substantial effect putting these items, and they must be taken in account when interpreting the results.

The other two output sets described in this chapter have a common aircraft description, given in Figure 4.4. The first of the two is the Atlanta Airport run which was used as an example earlier. The traffic mix for this run is given in Figure 4.5, and the airport description for our hypothetical version of ATL in Figure 4.6. The results are tabulated in Figure 4.7.

TYPE	SPEEDS (KNOTS)		RUNWAY OCCUPANCY (SECONDS)	
	LANDING	LIFTOFF	LANDING	TAKEOFF
1	145	155	50	33
2	140	148	44	30
3	130	140	39	28
4	115	125	33	23
5	98	110	29	21

Figure 4.4; Aircraft Descriptions for Run Sets II and III

TYPE	LANDING MIX	TAKEOFF MIX
1	19	19
2	58	58
3	2	2
4	14	14
5	7	7

Figure 4.5: Traffic Mix for Run Set II

AIRPORT CONFIGURATION FOR ATL AIRPORT

Number of Runways = 3

RUNWAY 1 (9L) - Takeoffs only

RUNWAY 2 (9R) - Landings only

RUNWAY 3 (15) - Dual Use, Alternating Operations

Runways 9L and 15 intersect at a point 2400 feet from the end of Runway 9L and 1677 feet from the end of Runway 15.

Runways 9R and 15 intersect at a point 5997 feet from the end of Runway 9R and 6520 feet from the end of Runway 15.

Runways 9L and 9R are semi-dependent parallels - Simultaneous arrivals are prohibited.

Figure 4.6: Airport Description for Run Set II

HOUR	OPERATIONS GENERATED LANDINGS	OPERATIONS GENERATED TAKEOFFS	OPERATIONS GENERATED TOTAL	OPERATIONS PERFORMED LANDINGS	OPERATIONS PERFORMED TAKEOFFS	OPERATIONS PERFORMED TOTAL	TOTAL DELAY (MINUTES) LANDINGS	TOTAL DELAY (MINUTES) TAKEOFFS	TOTAL DELAY (MINUTES) ALL	DELAY PER A/C (MINUTES) LANDINGS	DELAY PER A/C (MINUTES) TAKEOFFS	DELAY PER A/C (MINUTES) ALL
1	2	4	6	4	17	21	9.5	728.5	737.9	2.4	42.9	35.1
2	0	4	4	2	4	6	0.	0.	0.	0.	0.	0.
3	19	1	20	16	1	17	35.3	0.	35.3	2.2	0.	2.1
4	1	3	4	4	4	8	11.3	2.8	14.2	2.8	0.7	1.8
5	0	5	5	0	5	5	0.	0.	0.	0.	0.	0.
6	5	1	6	4	0	4	0.	0.	0.	0.	0.	0.
7	4	1	5	4	2	6	0.	0.	0.	0.	0.	0.
8	3	9	12	4	8	12	0.6	2.7	3.4	0.2	0.3	0.3
9	1	20	21	0	10	10	0.	11.6	11.6	0.	0.7	0.7
10	35	14	49	21	17	38	48.9	12.2	61.0	2.3	0.7	1.6
11	33	26	59	33	20	53	452.9	61.2	514.1	13.7	3.1	9.7
12	41	24	65	35	26	61	721.0	68.5	789.5	20.6	2.6	12.9
13	44	44	88	33	27	60	762.2	267.0	1029.3	23.1	9.9	17.2
14	46	47	93	38	33	71	1467.0	986.2	2453.2	38.6	29.9	34.6
15	33	51	84	30	28	58	1563.2	1159.5	2722.7	52.1	41.4	46.9
16	33	53	86	33	29	62	2191.7	2085.1	4276.8	66.4	71.9	69.0
17	62	41	103	34	30	64	2312.2	3061.6	5373.8	68.0	102.1	84.0
18	60	54	114	27	28	55	2003.6	3400.4	5403.9	74.2	121.4	98.3
19	31	25	56	38	34	72	3904.7	5167.0	9071.7	102.8	152.0	126.0
20	13	15	28	37	27	64	4467.4	4608.9	9076.3	120.7	170.7	141.8
21	11	12	23	32	31	63	4779.9	5718.1	10498.0	149.4	184.5	166.6
22	6	11	17	39	38	77	5848.4	8272.8	14121.1	150.0	217.7	183.4
23	10	17	27	28	32	60	1515.4	6208.2	7723.7	54.1	194.0	128.7
24	20	9	29	17	35	52	42.9	4226.0	4268.9	2.5	120.7	82.1
TOTAL	513	491	1004	513	492	1005	32138.1	46048.3	78186.4	62.6	93.6	77.8

Figure 4.7 Sample Model Output

The ATL airport configuration consists of a pair of offset parallels 4400 feet apart, plus a crossing runway. However, about 96% of the traffic uses one or the other of the two parallels. Since the parallel runways are less than 5,000 feet apart, all landings must be separated by 3 miles regardless of which runway they employ. The parallels are operated in the manner of Case 1 of run set I, one for landings and one for takeoffs. The simulation was started at 7 a.m. and was preloaded an hour before delay and throughput were recorded. Therefore the first hour for which delay and throughput are recorded is the one labeled 9 in the output.

Since aircraft are generated about 15 minutes before landing or start-of-roll, the first landings and takeoffs recorded in hour 9 were actually generated during hour 8. Similarly, at the end of the simulation in hour 8, some takeoffs and landings have been generated but not yet performed. Although the expected number of aircraft generated in hour 8 at the beginning of the simulation is equal to that at the end, the actual numbers in any particular sample run need not be the same. This explains the apparent discrepancy of one more takeoff performed than was generated.

Since delay and throughput are recorded at touchdown or start-of-roll, the average delay per aircraft for an hour is calculated as total delay for that hour divided by the number of operations performed in that hour. For this run, the average delay per aircraft was much too high to be tolerated at any actual airport. However, as noted earlier (in Section 2.2.5), the traffic levels employed are taken from total traffic (VFR and IFR) at San Antonio Airport; the true traffic levels and peaking characteristics at ATL may be quite different.

The third set of run outputs to be discussed in this chapter refer to an approximation of John F. Kennedy International Airport (JFK). The traffic mix for this run is given in Figure 4.8. The airport configuration, the most complicated one employed in these model exercises, is portrayed in Figure 4.9; Figure 4.10 shows how it appears on the preprocessor output.

Runways 4L and 4R are close parallels (less than 3,500 feet apart), while runways 31R and 31L are wide parallels (more than 5000 feet apart). Runway 32 (which cannot be located solely from the information given to the simulation) is shown in proper position in Figure 4.10. This is a good example of a case in which more information than is needed by the model would be required to produce pictorial output. The model only needs the interference patterns. We know runways 32 and 31R interfere on both landings and takeoffs, but since they do not intersect we do not not need to know where they are in relation to one another. We also know

Type	Landing Mix	Takeoff Mix
1	54	54
2	35	35
3	1	1
4	6	6
5	4	4

Figure 4.8: Traffic Mix for Run Set III

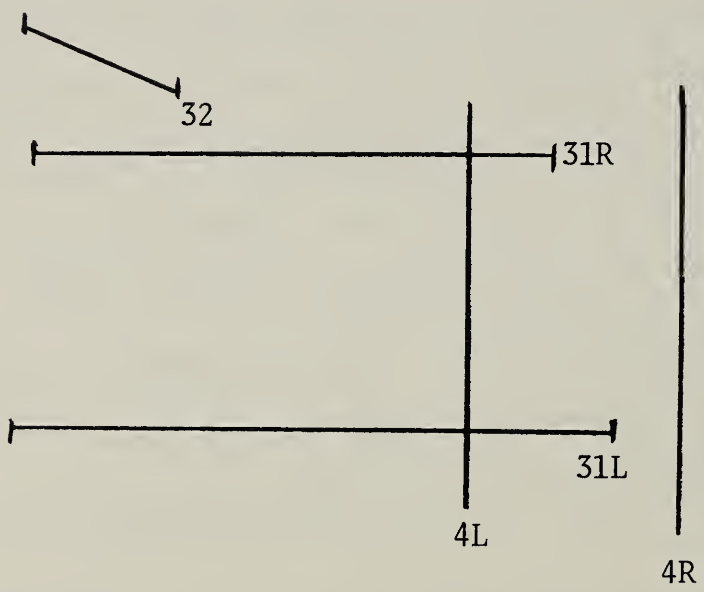

Figure 4. 9 : Pictorial Description of JFK

AIRPORT CONFIGURATION FOR JFK AIRPORT

Number of Runways = 5

RUNWAY 1 (31L) - Takeoffs only

RUNWAY 2 (31R) - Dual Use, Alternating Operations

RUNWAY 3 (4L) - Dual Use, Landings Take Precedence

RUNWAY 4 (4R) - Landings only

RUNWAY 5 (32) - Dual Use, Alternating Operations

Runways 31L and 4L intersect at a point 2096 feet from the end of Runway 31L and 3427 feet from the end of Runway 4L.

Runways 31R and 4L intersect at a point 1288 feet from the end of Runway 31R and 10748 feet from the end of Runway 4L.

Runways 31L and 31R are independent parallels - simultaneous operations are permitted.

Runways 31R and 32 are dependent - No simultaneous operations are permitted.

Runways 4L and 4R are dependent parallels - No simultaneous operations are permitted.

Figure 4.10: Airport Description for Run Set III

runway 4R is parallel to runway 4L, that 4R does not intersect either 31R or 31L, and that 4R and 4L are separated by less than 3500 feet, but do not have to know exactly how far apart 4R and 4L are or if their endpoints are offset.

Results of run set III are given in Figure 4.11. Again this is a 24-hour run, from 8 a.m. one day to 8 a.m. the next with the hour from 7 to 8 the first day run to preload the system. The same traffic distribution was used in run sets II and III,and since the random number sequence and starting times were the same, the exact same traffic sample was obtained. As can be seen from the output, landing delay for JFK is greatly reduced relative to ATL. However, takeoff delay is much too great. This is probably due to the distribution of runway use: 89% of all takeoffs use runway 31L, while another 9% use 32 and only 1% use 31R. (Again it should be noted that these are hypothetical data, which for greater realism should probably be modified to put more takeoffs on runway 31R.) Also, 85% of all takeoffs are bound on the same departure path, tending to increase average takeoff-takeoff separation requirements and hence mean takeoff delays. This latter feature, however, may reasonably reflect the effects of the restrictions governing use of the crowded New York area airspace; real New York data would be needed to verify this.

SUMMARY REPORT FOR THIS RUN

HOUR	OPERATIONS GENERATED			OPERATIONS PERFORMED			TOTAL DELAY (MINUTES)			DELAY PER A/C (MINUTES)		
	LANDINGS	TAKEOFFS	TOTAL	LANDINGS	TAKEOFFS	TOTAL	LANDINGS	TAKEOFFS	ALL	LANDINGS	TAKEOFFS	ALL
1	2	4	6	4	20	24	0.6	7784.5	7785.1	0.1	389.2	324.4
2	0	4	4	2	20	22	0.	8261.3	8261.3	0.	413.1	375.5
3	19	1	20	14	19	33	18.6	7576.3	7594.9	1.3	398.8	230.1
4	1	3	4	6	20	26	1.7	7181.5	7183.2	0.3	359.1	276.3
5	0	5	5	0	20	20	0.	6011.7	6011.7	0.	300.6	300.6
6	5	1	6	3	14	17	0.	1762.5	1762.5	0.	125.9	103.7
7	4	1	5	4	1	5	0.	0.	0.	0.	0.	0.
8	3	9	12	5	9	14	0.6	7.5	8.1	0.1	0.8	0.6
9	1	20	21	0	16	16	0.	14.6	14.6	0.	0.9	0.9
10	35	14	49	20	19	39	6.3	115.5	121.9	0.3	6.1	3.1
11	33	26	59	38	18	56	110.1	60.3	170.4	2.9	3.3	3.0
12	41	24	65	40	23	63	43.6	217.5	261.1	1.1	9.5	4.1
13	44	44	88	42	28	70	62.4	310.0	372.4	1.5	11.1	5.3
14	46	47	93	43	27	70	56.4	879.8	936.2	1.3	32.6	13.4
15	33	51	84	38	26	64	123.9	1327.1	1451.0	3.3	51.0	22.7
16	33	53	86	33	28	61	66.3	1806.9	1873.1	2.0	64.5	30.7
17	62	41	103	44	23	67	211.0	2508.5	2719.5	4.8	109.1	40.6
18	60	54	114	49	24	73	825.6	3215.6	4041.2	16.8	134.0	55.4
19	31	25	56	54	27	81	1105.6	3857.5	4963.0	20.5	142.9	61.3
20	13	15	28	22	21	43	182.1	4531.0	4713.1	8.3	215.8	109.6
21	11	12	23	14	23	37	4.3	5245.5	5249.7	0.3	228.1	141.9
22	6	11	17	10	21	31	0.1	5910.3	5910.4	0.0	281.4	190.7
23	10	17	27	11	25	36	2.2	6453.6	6455.8	0.2	258.1	179.3
24	20	9	29	17	22	39	1.3	7110.7	7112.0	0.1	323.2	182.4
TOTAL	513	491	1004	513	494	1007	2822.6	82149.7	84972.3	5.5	166.3	84.4

Figure 4.11: Results of Run Set III

This is the last of our illustration model run sets. The reader will have observed that the set of outputs presented was rather restricted, including only delay and throughput information, broken down only by operation and hour of day. Any of these statistics could also have been recorded by runway or by aircraft type. If the average number of passengers per aircraft of each type were provided to the model, average delay per passenger could be computed, broken down by any of the above categories. A frequency distribution showing the number of aircraft delayed by up to 5 minutes, by between 5 and 10 minutes, between 10 and 15, etc. can be computed and output in a graphical form if desired. (The exact size of the recording interval for such a distribution would of course be a user-specified parameter.) In addition to delay and throughput statistics, queue-length statistics could also be gathered. The average or the maximum number of aircraft waiting could be recorded by runway and/or by operation and/or by hour of day. All of these items are available with some recoding of both the preprocessor and the simulation. Because the present version of DELCAP has not as yet been run in a production environment, the format and content of output most useful in that environment are not yet known. Designing the "production" output options is one of the areas which must be included in future work on the DELCAP model.

Finally, we wish once again to emphasize that the DELCAP model has so far been run for debugging and logic testing purposes only, in exercises not including the validation checks or sensitivity analyses required for genuine applications. That such further resource-consuming and time-consuming steps could not be executed during the present phase of the effort was not surprising, but without them the work which was accomplished is left in a frustratingly tentative and tenuous status. For example, time did not permit an intensive search for maximally complete and consistent data to be used in validation runs. Such a search should be the first task in any future DELCAP efforts; one cannot confidently interpret or utilize results from an unvalidated model. Once DELCAP has received this basic checkout, sensitivity analyses should be performed to investigate the effects on throughput and delay of such variables as: aircraft mix, operational procedure, runway-use distribution, diurnal traffic distribution, runway occupancy times, aircraft speeds, and separation rules. These analyses will not only provide a valuable insight into the effects on throughput and delay of variation in the airport parameters, but will also provide guidelines for the accuracy needed in collection of data for input to DELCAP.

In a more positive vein, we may note that the DELCAP model has now produced, with an acceptably small expenditure of computer time, delay and throughput figures for a complicated runway configuration with an appropriately high traffic level. The kinds of input data required by the model have been found in the published literature, or at least can in principle be secured without a great data collection effort. The illustrative runs and analyses reported in this chapter have shown that DELCAP can be a useful tool in gaining insight into the roles and interactions of separation rules, airport configurations and operating procedures, and traffic distributions as they affect delay and throughput.

5. RECAPITULATION AND NEXT STEPS

In this report we have described the present version of the DELCAP simulation model and its preprocessor, which are currently operational on the UNIVAC 1108 computer at the National Bureau of Standards. DELCAP has been tested on several different runway configurations, and these tests support our expectation and intention that it be applicable to a wide variety of runway configurations. The model (preprocessor included) takes only about 12 seconds to simulate a day's activity at a major airport.

The DELCAP model is envisioned as a planning and analysis tool. To that end, we adopted the "model as little as you can" philosophy of computer simulation, and designed a preprocessor to permit the user to utilize standard data sets whenever desired and to input other data in an easily prepared format. The simulation model includes only those elements of the airport system which affect throughput and aircraft delays, and is based on the idea of an interference point, one of those points on the path of a landing or takeoff (in the neighborhood of an airport or the surface of the runway) where the presence of one aircraft impedes the free progress of another aircraft because of required separation between the two. The simulation contains two interference points for landings, namely the outer marker and the runway threshold, and one such point for takeoffs, the end of the runway at which the takeoff starts its roll. The model deals in detail only with the times at which an aircraft passes these interference points; flight paths, holding patterns and takeoff queues are not represented explicitly. Only the interferences among aircraft, which dictate when a flight may pass the interference points along its path, are treated in detail in the DELCAP simulation since these interferences suffice to determine the throughputs and the delays which are to be measured by the simulation.

Two factors had to be balanced in the design of the preprocessor: flexibility of input versus ease of input. Since DELCAP is designed as a tool for aiding analysts in the planning process, it must be easy and convenient for use by one whose specialty is airports rather than computers. However, it is expected that planners may wish to vary almost all inputs to the DELCAP simulation. Therefore, the preprocessor was designed to allow the user to choose those types of data he wishes to provide himself, while still providing "standard" sets of data for all input groups so that a complete data set need not be prepared for each run. The formats for user-supplied data are designed to make their preparation as easy as possible, and to allow easy modification where a desire for such modification can be anticipated. Input data are divided into six groups and a user may either insert (all) the data of a group or else utilize the standard data for that group. For those groups which depend on the airport, a user wishing to use standard input must specify from which airport's file these data are to be drawn. This scheme allows a wide variety of types of analyses to be made using DELCAP, by providing a convenient means for varying almost all inputs. However, it limits the data needed for any one run to those which the user wishes to differ from the standard data provided. Thus the preprocessor provides both flexibility and ease of input.

Our documentation has described the current versions of both the DELCAP simulator and its preprocessor. In the course of these descriptions we have noted natural areas for future work on the DELCAP modeling system. We will now list these areas again, to give the reader a better idea both of where we are currently, and of what remains to be done before the mature DELCAP described in the "Mock Instructions" memo can be available.

5.1 Validity and Model Sensitivity

The scope of the current work was too limited to allow adequate validity and model-sensitivity tests. Only basic logic-testing has been done, to insure that the model does in fact perform as it was designed to do. Efforts to confirm that this design yields satisfactory agreement with actual system performance were not included in the present phase. In order to do this, it is necessary to obtain data from existing sources (or to collect them specifically for this purpose), and to run the model to compare model outputs with these collected data. The effort would include preliminary sensitivity analyses to determine what degree of agreement between model outputs and the observed data can reasonably be expected. Such sensitivity analyses will also point out the degree of accuracy needed in collecting model input data in order to achieve desired accuracy in model output. For instance, even the small set of testing runs reported in Chapter 4 shows the need for care in defining the operational procedures, the runway use distribution, and the distribution of departure paths.

We feel that validity testing is the next logical step in bringing the DELCAP model into full operational status. Of the two types of output, throughput figures should be more easily verified than those for delay, since the number of landings and takeoffs in a given period of time can be calculated fairly easily. The delay recorded by DELCAP, however, depends on the description of the "system" included in the model, and may not be the same as any other delay measure now tabulated. Therefore, a special data collection effort might be needed to provide an empirical basis against which to check DELCAP's delay figures; this possibility (and potential alternatives) must be explored systematically.

A model whose performance has not been checked for validity cannot be properly evaluated, adjusted, or elaborated. This is why we recommend validity-testing of the present version as the next step in the process of making DELCAP a useful tool for planning. However, we do not wish to imply that such testing would end at that point. There are several extensions and revisions to the model which, we suggest in the following sections, should be undertaken subsequent to the preliminary validity checks. Further testing against real-world data would be needed to insure that these procedures, too, have been modeled properly.

Validity checks and any associated data collection efforts may themselves indicate model changes. The results of such checks are unlikely to be a clear-cut "yes" or "no"; more probably, some areas will prove to be better represented by the model than are others. Such findings may suggest modifications either in model logic or in the structure of the data required by DELCAP. In this sense, validity checking is an on-going process, which is included implicitly in the other activities suggested in this chapter and which can prove a major stimulus and guide to model improvements.

5.2 Sensitivity Analyses

After preliminary establishment of validity to within appropriate limits, the model may be exercised in examining the sensitivity of delay and throughput to variation in the model inputs. We have already mentioned preliminary model sensitivity work, whose purpose is to ascertain the ranges over which the model variables produce reasonable outputs. Such preliminary work would also address the problem of which variables the model is particularly sensitive to, i.e. which ones produce the most noticable effects in delay and throughput. This first stage of sensitivity analysis, conducted in conjunction with validity checks, would be on the order of "tinkering" with the model, seeing what it will and will not do, how it acts and reacts under various scenarios. The scenarios might or might not be real or realistic; rather, the focus is on the model and how it operates.

With the completion of the first testing cycle, however, the sensitivity analysis may proceed to a second stage, that of testing system sensitivity. That is, with the validity of the model's representation of the airport system established, the model can be used to test that system's reactions to hypothesized patterns of changes in the airport configuration and operating procedures, traffic mix and levels, and separation rules. The model can be used for example to evaluate what traffic levels result in average delay per aircraft of over 20 minutes. The dependence of throughput on aircraft speed can be studied. The object of this phase of sensitivity testing is to find how much of a change in throughput and delay will result from a known percentage change in some input variable. Responses to changes in combinations of variables can also be tested. The results of such sensitivity testing can aid the analyst in understanding the complex interactions in the airport system, and can help him in designing a minimum series of runs to investigate those particular changes at particular airports which he desires to evaluate. The sensitivity runs may eliminate the need for some runs altogether by duplicating common requests. They may also provide advance guidance as to which changes are most likely to achieve the planner's goals.

5.3 Model Modifications

Once validity of the model has been established, and either concurrently with or subsequent to the sensitivity analyses, certain model modifications should be undertaken. They fall into six categories: (1) investigating the possibility of modifying the scheduling of landing and takeoffs (see Section 3.3), (2) revising the output, (3) including the effect of weather on the system, (4) investigating the advisability of using probability distributions rather than single values for some of the variables, (5) the inclusion of VFR traffic, (6) allowing some variables to depend on time-of-day, and (7) adding some ground operations to the model. The seven items appear roughly in what we feel to be the order of their priorities.

The first modification is designed to overcome the difficulty in representing a slow landing following a fast departure. In this case the landing can cross the outer marker before the takeoff leaves its gate, remain adequately separated from the takeoff, and land after the takeoff. This creates a problem in the present model because the landing should be scheduled first, although the takeoff starts its roll before the landing touches down. That is, the landing and takeoff should be scheduled in the opposite order to their occurrence.

Another objective here concerns remedying a second difficulty in the present model's procedure for scheduling landings and takeoffs. The routine FREER scans the tieups pertaining to a particular flight, and LAND or TOFF is scheduled when none of the tieups is any longer in force. However, there may be a gap between the tieups sufficiently large that the present flight could be scheduled during it without affecting previously scheduled flights. If so, the present flight could in reality be scheduled earlier than the present version of the model permits.

Modifying the scheduling procedure to change these imperfections in the present model <u>may</u> turn out to be a difficult task requiring major modeling changes and additions. For this reason it was decided that the simpler version of the model was the one to be produced during the present effort. However, an effort should be made at least to ascertain the difficulties involved in resolving these two scheduling problems.

The <u>second</u> listed area of model modification is the content and form of the output. As mentioned previously, the model has been run only to check out its logic. The current output set was designed to be representative of the types of output readily available from the model. Once the model's validity has been established it will be time to take a further look at what types of output are most desirable. For a well-based decision on this point, it is necessary to have a better idea of the types of applications for which the model will be used. Probably, different sets of output will be most relevant for different applications. In this case a large variety of output would be made available, with the user free to specify which sets he desired. This procedure would require another specification by the user, but this seems clearly worthwhile if it would procure a set of outputs much more tailored to his particular preferences and needs.

The last five model modifications involve the addition of features to the model. The first of these is the representation within the model logic of the effects of weather on the system. At present the only way to mirror changes in weather is to vary the parameters describing the traffic, separation rules or operational procedures. This cannot be done within a model run, since the associated input parameters are presently constant throughout each run. Future modification should include options by which a user could provide weather information, including a wind profile, and should automate the parameter-variations resulting from changes in weather conditions. At this point it seems unlikely that we would want to include automation of weather variations so major as to force the airport to operate in the opposite direction, but the feasibility of even this capability should be investigated. Bad weather also affects the number of missed approaches. In the present version of the model there is no provision for these maneuvers, but the possibility of their inclusion in a future version should be studied.

In the present model version, such parameters as landing speeds and runway occupancy times are single values, one for each aircraft type. Yet two "identical" aircraft, loaded the same, need not have the same exact runway occupancy time or landing speed. These parameters could be represented by probability distributions, presumably with the current parameter values as means.

The present model also assumes that a landing aircraft can be delivered to the outer marker as soon as tieups permit. The degree to which this ideal can be attained depends on the pilot and controller, and is therefore subject to human error. Thus there are many areas in the current model where a single value has been used to represent items which by their very nature admit a range of possible values. For this reason, we suggest that those model variables which (like the ones mentioned above) might best be represented by distributions be identified, and that an appropriate distribution for each be incorporated. Some of these distributions may be made responsive to parameters not now included in the model, such as controllability (human or mechanical) or weather descriptors. This may allow a fuller range of scenarios to be tested by the planner, and should provide a better representation of variations among aircraft.

Yet another model addition would be provision for the inclusion of VFR traffic. All separation rules in the present model involve IFR traffic alone. When dealing with the throughput of an airport, however, it is desirable to include all traffic, VFR as well as IFR. Different separation rules apply to VFR traffic, and such concepts as the outer marker for a landing do not carry over from IFR to VFR operations. Ways of merging the two types of operation within the structure of the DELCAP model should be investigated, but this enlargement may require a major effort.

In the present version of the model such variables as user mix, the runway use distribution, and the operational procedure are constant for the duration of a run. However, it seems desirable to vary these in a manner similar to the current variation in traffic levels. The day might be broken down into several periods, and a different set of parameters be provided for each period. In this way a runway handling landings only could be switched to one handling both landings and take-offs at a time of day when takeoffs predominate. If during certain periods during the day the traffic mix contains fewer small aircraft than at other times, this would be reflected in the model. The coding to accomplish the dependence on time-of-day is relatively minor, so that the increased flexibility is bought at a low price.

The final feature whose addition to the model is proposed is a representation of some ground operations, mainly those (such as on taxiways, exit ramps and holding ramps) which affect the aircraft between its gate and the runway surface. Gate assignment might be included, if its inclusion can be demonstrated to be worth the amount of extra input data required. Some ground operations, such as a plane crossing a runway to get to the terminal, can impede landings and takeoffs. Others may affect sequencing of operations, such as when a takeoff holding area becomes filled and there is no other waiting area available. Still others, such as baggage handling activities, are clearly beyond the proper scope of a model designed to focus on aircraft delay. Those aspects of ground operations which affect the delay and throughput figures calculated by DELCAP should be considered for incorporation in the model; all others should be excluded.

The seven areas of modification described above are the major ones suggested for future work on the DELCAP model. Others may suggest themselves as the work goes on. Several minor modifications have not been included in the list above. For example, more sophisticated sequencing techniques, such as ones depending on the number of landings and takeoffs waiting to use a runway, might be incorporated in the model.

All such additions and modifications should be weighed carefully before they are undertaken. The decision to go ahead must include a balancing of the additional data requirements, if any, against the benefits to accrue from the modification. We reiterate that ease of use is a major goal of the DELCAP model. The more decisions the user must make, on choosing a standard data set versus providing his own inputs, the less convenient the model is to him. If the input is one he would expect to provide (such as some specification of weather conditions), indeed one whose absence would arouse suspicion of the model's outputs, then the extra burden of providing this data is certainly worth it. If the modification requires a great deal of data, and promises little change in the output figures, the modification is much less desirable. All modifications must be studied with this tradeoff in mind, and only those with a favorable "rate of return" should be undertaken.

5.4 Airport Data File

In order to achieve the operation of the DELCAP model as described in the DELCAP Mock Instructions, it is necessary to compile a data file containing preprocessor input data for each airport (or at least each major airport) in the U.S. The current illustrative "file" contains information on just two airports, ATL and JFK, much of it invented either wholly or in part. The process of creating the complete data file can be divided into two parts, one dealing with the form of the data (and any necessary reprogramming of the preprocessor), and the second with the actual gathering of the information.

The changes suggested in section 5.3 may necessitate additional inputs for the DELCAP model. The validity and sensitivity tests may suggest a reworking of the data group categories and changes in preprocessor input formats. The format of the data file itself is likely to be altered to facilitate easier coding. All of these changes involve some recoding of the preprocessor program.

The second activity involved in creating the airport data file is the actual collecting and collating of the data, and preparing them in the form necessary for input to the preprocessor. It is not known at present exactly which of the various data types are available from existing sources, or in what form. Therefore preparation of the file must begin with a comprehensive search of data sources, to ascertain what data are available where, and to establish if possible the consistency of data from different sources. This is a laborious and time-consuming task in itself, but even after data sources are found and checked, there remains the effort of extracting and coding these data in the form required by DELCAP. It is unlikely that all the needed data will already be on record, so that some items will probably have to be collected in the field. Also, since DELCAP is designed for use in the planning process, much of the input file will refer to future years. Therefore it will be necessary to obtain from existing sources (or to project from current data) future traffic levels, traffic mixes, and airport configurations for the desired planning years.

Although preparing the airport data file needed by DELCAP is a major task requiring much effort and time, it should have a payoff in addition to its use for the model. Such a data file could be examined to study the properties of airports in the U.S.. Easily accessible, it will provide people in the FAA with airport information practically at their finger-tips. There will be one, self-consistent source of airport data.

5.5 Conversion of the Model to a Time-Sharing System

The final step, in changing the present version of the model into the version described in the DELCAP Mock Instructions, is that of converting the programs to a time-sharing system. Such a system would enable a user to access the program from a terminal in his own or a nearby office. He would need to know very little about the operation of a computer, probably only how to activate the terminal and call up the program. The program would operate in a conversational mode, asking such questions as "Do you wish to supply airport data?", and if the answer was "No", "Which airport on file do you wish standard data to be chosen from?". The user would type in his replies to each question as it is asked. (The actual form of the questions is of course a matter for study, and must be geared to the backgrounds of the users.) Once all data were furnished, the model would be run to supply the delay and throughput figures requested. The user could then make a few changes in his original data set and run again, or could start over with an entirely new problem. The program runs so rapidly that the results would be available while the user remained at the terminal. Several similar runs could be made in a few minutes to a couple of hours depending on the amount of data to be typed in. This kind of procedure allows analysis to be conducted as the model is run. Questions such as "What if such-and-such a variable is changed?" can be answered on the spot with very little effort. The analysis procedure is therefore both quick and convenient.

This concludes our list of future work to be done on DELCAP. Some of it, such as the validity and sensitivity analyses, is necessary before the model can usefully be applied. Other parts, such as the model modifications in Section 5.3, may or may not be desirable depending on the uses envisaged for the model. Some of those modifications may of course be more desirable than others, and all must be viewed in light of the demands to be made on the model. The data file would bypass the need for each analyst to collect and assemble his own data set, and in addition would encourage more general analyses than would be undertaken were the data not immediately available. Conversion to a time-sharing system is not essential, since the model could be used without it. However, the advantages of having a model at one's fingertips without having to learn much about running on a computer mean that DELCAP would be more accessible to the user and more convenient to his use, providing him with a tool which could really be a major aid in the whole planning process.

6. BIBLIOGRAPHY

6.1 Air Traffic Control Procedures

1. Airman's Information Manual, Part 1, Basic Flight Manual and ATC Procedures, FAA, November 1969

2. Terminal Air Traffic Control, FAA7110.8A, January 1970

6.2 Capacity Studies

3. Airport Capacity Criteria Used In Long-Range Planning, FAA AC 150/5060-3A, December 1969

4. Airport Capacity Handbook, Second Edition, Airborne Instruments Laboratory, June 1969

5. E. N. Hooton, H. Burns, and M.A. Warskow, Operational Development Techniques for Computing Airport Capacity, FAA Report Number RD-68-20, June 1969

6. Analysis of a Capacity Concept for Runway and Final-Approach Path Airspace, NBS report number 10111, November 1969

6.3 Simulation Studies

7. M.R. Abbott, A Summary of Some Digital Simulations Relating to Runway Utilization, Royal Aircraft Establishment Technical Report Number 66087, March 1966

8. J. M. Chaoken and R. H. Fowler, A Simulation of an Airport Runway System, University of California at Berkeley, September 1969

9. R. B. Coulson and V. D. Burgman, An Investigation into Air Traffic Control by a Simulation Method, Australia, March 1950

10. Final Report on Air Traffic Control System Simulation Study, Norden, August 1969

11. General Description of Computer Simulation Model Developed by RDSA/PRC During 1968 and Presently Operable, R. Dixon Speas Associates, December 1968

12. Gordon Jolitz, Simulation and Analysis of ATC/CAS System Design, FAA-NAFEC, November 1968

13. K. W. Kowalke, The Fast Time Simulation Program, IBM, March 1960

14. J. Ray Ruetnik and John H. Thompson, Computer Simulation Model of Terminal Air Traffic and PWI Systems, Kaman AviDyne, March 1969

15. Charles Sheffield, Multiple Aircraft Models in Stare Fast Time Simulation Experiments, World Systems Laboratories, January 1963

16. Simulation Tests of Air Traffic Control Operations in the Denver-Colorado Springs Area, Navigational Aids Evaluation Division FAA, Indianapolis, Indiana, May 1959

6.4 Programming Manuals

17. Harry M. Markowitz, Bernard Hausner, and Herbert W. Karr, SIMSCRIPT A Simulation Programming Language, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1963

18. SIMSCRIPT 1.5, C.A.C.I. 66-INT-3, Consolidated Analysis Centers Inc., Santa Monica, California, 1967

19. UNIVAC 1108 FORTRAN V, Programmers Reference Manual, Sperry Rand Corporation, 1966

6.5 Data Sources

20. L. N. Million J. M. Copeland, J. T. Hornsby, and W. C. Keepers, A Suggested Action Program for the Relief of Airfield Congestion at Selected Airports, FAA, April 1969.

21. Approach Airspace and Runway Capacity Parametric Sensitivity Analysis, Boeing-Airplane Group, May 1970.

22. National System Effectiveness Measurement Program, FAA, 1965 - 1966.

APPENDIX A. INPUT FORMATS AND USER INSTRUCTIONS

This appendix is divided into six sections, (1) preprocessor input formats, (2) sample preprocessor output (simulation input) (3) simulation input (system specification card), (4) exogenous event tape format for explicitly scheduled flights, (5) the airport data file format, and (6) the current sample airport file.

A.1 Preprocessor Input Format

The first two cards must always be provided. Card one contains the times for the beginning and end of the simulation. The hours run from 0. (midnight) to 23. (11 p.m.) and the number of minutes past the hour goes to the right of a decimal point (so that 7:45 p.m. would be 19.45).

The second card contains the INPUT array. The input variables are divided into six groups. The user may provide any of these groups or request that the program supply standard data. Within each of the six groups all or none of the data must be supplied. The INPUT array indicates which of the groups is to be read from the user. Each element of INPUT is a three-character code. If the code is left blank, the program will read user-supplied data. For information which varies from airport to airport, the three-letter code associated with any airport (e.g. JFK, ATL) will cause the program to search its data tape for data representing that airport. If the tape does not contain data on the requested airport, an error message will be printed and the program will stop. For data groups which are independent of airport any code other than three blanks will provide standard data. The variables in each group will follow.

Group I - includes data about each type, or class, of aircraft. It is independent of airport. If INPUT(1) is blank, user must provide one card for each type (up to 10) with average landing and takeoff speeds, average takeoff and landing runway occupancy times, and the presence or absence of distance-measuring equipment. The largest type number must be placed last, followed by an end-of-file card. The next card contains an average turn-off speed for all types.

Group II - includes percentages of each type of aircraft in the takeoff and landing mixes. It is airport-dependent. If INPUT(2) is blank, two cards must be provided: the first with the take-off mix, the second with the landing mix.

Group III - contains the average numbers of takeoffs and landings per hour. It is airport-dependent. If INPUT (3) is blank, user must provide data for every hour to be simulated plus one hour previous to the beginning,up to a total of 24 hours. The takeoff rates come first, twelve numbers to a card, on one or two cards as needed, followed by the landing rates.

Group IV - includes three separation distances,

1) distance to the departure/arrival fix.
2) radar separation between airborne landing aircraft
3) radar separation between a landing and a takeoff from the same runway.

All three distances are in nautical miles. This group is independent of airport. If INPUT (4) is blank, user provides one card with these three distances.

Group V - includes the description of the runways and the operation of the airport. If INPUT (5) is blank, the following cards must be supplied:

1) number of distinct departure paths from the runways.

2) one card for each runway, up to 9, with its heading, its left/right designation if it is one of a set of parallels, its operation code (1, 2, 3, or 4, see later), and the distance to its outer marker, followed by an end-of-file card.

3) one card for each runway containing the time for each type of aircraft to fly from handoff to outer marker of that runway,

4) for each intersection, the numbers of the intersecting runways, the distance, in feet, from the first to the second, and from second to first, followed by an end-of-file card.

5) for each pair of runways which are not independent, the numbers of the runways and the dependence code (1 or 2, see later). [Note -- this may change from VFR to IFR. Two runways which are VFR independent may well be IFR dependent.] These are followed by an end-of-file card.

<u>Group VI</u> -- includes percentages of each type of aircraft using the runways and departure paths. It is airport-dependent. If INPUT(6) is blank, one card for each type must be supplied giving the fraction of total takeoffs of that type using each runway, and the fraction of total landings of that type using each runway. [Note: if there are more than six runways, two cards will be needed for each type.] These are followed by one card for each runway, giving the fraction of all takeoffs from that runway using each departure path.

Card No.	Column Nos.	Variable	No. Decimal Places	FORTRAN Format
1	1 - 7	TBEG - beginning of simulation	2	F7.2
	8 - 14	TEND - end of simulation	2	F7.2
2	1 - 18	INPUT (1 - 6), 3 columns per element	-	6A3
GROUP I				
one per type	1 - 3	number of type (≤ 10)	0	I3
	4 - 6	= 0 if type has DME > 0 if type does not have DME	0	I3
	7 - 13	aver. landing speed(knots)	2	F7.2
	14 - 20	aver. takeoff speed(knots)	2	F7.2
	21 - 27	aver. runway occupancy time - landing -(seconds)	2	F7.2
	28-34	aver. runway occupancy time - takeoff - (seconds)	2	F7.2
#types + 1		end-of-file		
#types + 2	1 - 7	aver. turn off-speed, all types	2	F7.2
GROUP II				
1	6 per type	decimal fraction of take-off mix, of each type	4	12F6.4
2	same	dec. frac. of landing mix of each type,	4	12F6.4
GROUP III				
1, 2	6 per hour	number of planes taking off per hour	1	12F6.1
3, 4	same	# planes landing per hour	1	12F6.1

Card No.	Column Nos.	Variable	No. Decimal Places	Fortran Format
GROUP IV				
	1 - 7	distance to departure/ arrival fix	2	F7.2
	8 - 14	required separation between landing planes in air	2	F7.2
	15 - 21	required separation between arrival and departure	2	F7.2
GROUP V 1	1 - 2	number of departure paths	0	I2
one per runway	1 - 2	number of runways (1 - 9)	0	I2
	3 - 6	heading of runway	0	I4
	7 - 8	left/right designation	-	A2
	9 - 12	operation code: 1-takeoffs only, 2-landings only, 3-both, alternating 4-both, landings preferred	0	I4
	13 - 19	distance to outer marker (naut. miles)	2	F7.2
#rw +2		end-of-file		
one per runway	7 per type	time, in minutes, for each type to fly from handoff to outer marker	2	10F7.2
one per intersection	1 - 2	first runway number	0	I2
	3 - 4	second runway number	0	I2
	5 - 12	distance from end of first to intersection (feet)	0	F8.0
	13 - 20	distance from end of second to intersection (feet)	0	F8.0
		end-of-file		

Card No.	Column Nos.	Variable	No. Decimal Places	Fortran Format
one per inter-ference	1 - 2	first runway	0	I2
	3 - 4	second runway	0	I2
	5 - 6	interference code: 1 - simultaneous dep/arr and dep/dep are permitted, given divergence, but arr/arr is prohibited, 2 - all simultaneous operations prohibited.	0	I2
		end-of-file		
GROUP VI				
one per type	6 per runway	decimal fraction of all takeoffs of type which use each runway, followed by decimal fraction of all landings of type which use each runway	4	12F6.4
one per runway	6 per dep. path	decimal fraction of all takeoffs from each runway going on each path	4	12F6.4

Sample Data Deck

Note: b is used to indicate a blank

```
bbb8.00bb20.00
XXXJFKbbbXXXJFKJFK
```

Note: This card indicates that the user wishes to supply only Group III, takeoff and landing rates. There would follow at most 4 cards.

Group I

```
bb1bb1b165.00b175.00bb59.00bb37.00
bb2bb1b135.00b145.00bb52.00bb34.00
bb3bb0b120.00b130.00bb45.00bb31.00
bb4bb0b110.00b120.00bb38.00bb28.00
bb5bb0b105.00b115.00bb31.00bb24.00
∇bEØF
bb30.00
```

Group II

```
0.3000b.20bbb.15bbb.15bbb.20
b.35bbb.25bbb.15bbb.10bbb.15
```

Group III

```
bbb9.0bbb3.0bbb2.0bbb2.0bbb4.0bbb2.0bbb2.0bbb9.0bb20.0bb23.0bb25.0bb35.0
bb40.0bb38.0bb46.0bb46.0bb51.0bb42.0bb18.0bb16.0bb16.0bbb9.0bbb7.0bbb9.0
bb17.0bbb0.0bb15.0bbb5.0bbb0.0bbb5.0bbb5.0bbb3.0bb17.0bb29.0bb33.0bb39.0
bb47.0bb37.0bb38.0bb40.0bb51.0bb51.0bb38.0bb17.0bb17.0bb11.0bb11.0bb16.0
```

Note that the symbol ∇ stands for a multipunch 7 8. The symbol Ø stands for the letter O.

Group VI

```
1.00bb0.0bbb0.0bbb0.0bbb1.00bb0.0
b.90bb0.0bbb0.10bb0.0bbb0.50bb0.50
b.75bb0.0bbb0.25bb0.0bbb0.55bb0.45
b.50bb0.0bbb0.50bb0.0bbb0.30bb0.70
b.25bb0.0bbb0.75bb0.0bbb0.05bb0.95
b.20bb0.65bb0.15bb0.34bb0.33bb0.33bb0.67bb0.05bb0.28
```

1

1 Even though runway two handles no takeoffs, giving these data does no harm, and makes them available if necessary.

Group IV

bbb4.00bbb3.00bbb2.00

Group V

```
b3
b1bb17bbbbb1bbb8.5 2
b2bb20Rbbbb2bb11.0
b3bb20Lbbbb3bbb8.0
ΔbEØF
bb10.25bb12.18bb14.10bb16.67bb18.33
bbb9.50bb11.25bb13.20bb16.00bb17.90
bbb9.15bb11.75bb13.33bb15.50bb17.40
b1b2bbb8500.bbb9000.
b1b3bbb2275.bbb5430.
ΔbEØF
b1b2b1 3
b1b3b1
b2b3b2
ΔbEØF
```

2. Even though these data are not relevant to runway one, they should be given, and do no harm.

3. Note that interference relates to the physical relationship among runways, not actual interference, since in this case operations on the runways preclude such interference.

A.2 Sample Preprocessor Output (Simulation Input)

```
  1    0 0 R     0   0   0   0 0 0    0         2
  2    0 0 R     0   0   0   0 0 0    0         3
  3    0 0 R     0   0   0   0 0 0    0         5
  4    0 0 R     0   0   0   0 0 0    0         3
  5    0 0 R     0   0   0   0 0 0    0         24
  6    0 1 R     3   2   0   0 0 0    0         10(I2)
1 2 3
  7    0 1 R     3   2   0   0 0 0    0         10(D3.4)
 8.0000  9.0000 18.0000
  8    0 1 R     3   2   0   0 0 0    0         10(I2)
2 1 3
  9    0 2 R     5   3   3   2 0 0 R  F0         12(D1.4)
.1380 .1720 .1380 .1430 .1790 .1430 .1540 .1920 .1540 .1740 .2170 .1740
.2040 .2550 .2040
 10    0 1 R     5   3   0   0 0 0    0         10(I2)
1 1 1 0 0
 11    0 1 R     5   3   0   0 0 0    0         10(D3.4)
145.0000140.0000130.0000115.0000 98.0000
 12    0 1 R     5   3   0   0 0 0    0         10(D3.4)
155.0000148.0000140.0000123.0000110.0000
 13    0 1 R     5   3   0   0 0 0    0         10(D3.4)
  .0140   .0123   .0108   .0092   .0081
 14    0 1 R     5   3   0   0 0 0    0         10(D3.4)
  .0091   .0083   .0077   .0064   .0058
 15    0 0 R     0   0   0   0 0 0    0         25.000
 16    0 0 R     0   0   0   0 0 0    0          4.000
 17    0 0 R     0   0   0   0 0 0    0          3.000
 18    0 0 R     0   0   0   0 0 0    0          2.000
 19    0 1 R     3   2   0   0 0 0    0         10(D3.4)
  .1287   .1338   .1437
 20    0 2 R     2   1   5   3 0 0 R  F0         12(D1.4)
.1900 .7700 .7900 .93001.0000 .1900 .7700 .7900 .93001.0000
 21    0 2 R     5   3   3   2 0 0 R  F0         12(D1.4)
1.00001.00001.00001.00001.00001.0000 .9000 .90001.0000 .9000 .90001.0000
.7000 .70001.0000
 22    0 2 R     5   3   3   2 0 0 R  F0         12(D1.4)
.00001.00001.0000 .00001.00001.0000 .0000 .90001.0000 .0000 .90001.0000
.0000 .70001.0000
 23    0 2 R     3   2   3   4 0 0 R  F0         12(D1.4)
.3300 .67001.0000 .3300 .67001.0000 .7500 .75001.0000
 24   24 1  Z    3   2   0   0 0 0    0
 25    0 1 R     3   2   0   0 0 0    0         10(I2)
1 2 2
 26    0 2 R     3   2   3   2 0 0 R  F0         12(D1.4)
.0000 .0000 .3950 .0000 .0000 .9870 .27601.0730 .0000
 27   28 2  Z    3   2   2   1 0 0    0
 29    0 2 R     3   2   0   0 0 0    0R        10(I2)
3 3 0 0 0 0 0 0 0 0                                              2
3 0 0 0 0 0 0 0 0 0                                              1
1 2 1 0 0 0 0 0 0 0                                              3
 30    0 2 R     3   2   0   0 0 0    0R        10(I2)
6 5 0 0 0 0 0 0 0 0                                              2
6 0 0 0 0 0 0 0 0 0                                              1
6 6 5 0 0 0 0 0 0 0                                              3
```

```
31    0 1 R      3   2    0    0 0 0     0          10(D3.4)
 .0603    .0678    .1357
32    0 2 R      3   4    3    4 0 0 R  F0          12(D1.4)
.0500 .0167 .0167 .0167 .0500 .0167 .0167 .0167 .0500
33   38 1   Z   24   5    0    0 0 0     0
39    0 0 R      0   0    0    0 0 0     0            8.000
40   45 1   Z    3   2    0    0 0 0     0
46    0 0 R      0   0    0    0 0 0     0           32.000
47    0 2 R      2   1   24    5 0 0 R  F0          12(D1.4)
.0909 .3333 .5000 .5000 .2500 .5000 .5000 .0909 .0500 .0417 .0400 .0385
.0250 .0278 .0185 .0185 .0179 .0233 .0526 .0588 .0588 .1111 .0714 .0909
.52699.9999 .0588 .20009.9999 .2000 .2000 .3333 .6667 .0385 .0333 .0263
.0227 .0286 .0278 .0250 .0196 .0189 .0278 .0588 .0588 .1111 .1111 .0526
48    0 0 R      0   0    0    0 0 0     0          8
49    0 2 R      3   4    3    4 0 0 R  F0          12(D1.4)
.0334 .0000 .0000 .0000 .0334 .0000 .0000 .0000 .0334
50   50 1   7    2   1    0    0 0 0     0
 0    0 0        0   0    0    0 0 0     0
```

These cards provide initial values for all system variables and arrays and permanent entities in the SIMSCRIPT simulation program. The format required is rigid and may require changing many cards to make a simple data change. Users interested in a more complete description of the required formats should consult SIMSCRIPT, A Simulation Programming Language, pages 115 through 128.

A.3 Simulation Input - the System Specification Card

Simulation input consists of the tape output by the preprocessor and one card supplied by the user and called the system specification card. Its format is as follows:

Column	Contains
1	the number 1
6	P, if printing of the initialization deck is desired
11 - 12	the number 50
17 - 18	the number 60
23 - 24	the number 60
35 - 36	the tape unit the initialization deck is to be read from; this is unit 8 if the preprocessor is used
41 - 42	the tape unit used for the exogenous events; this is unit 8 if the user is not specifying any explicitly generated flights.

A.4 Exogenous Event Tape Format for Explicitly Generated Flights

The first record must be for the BEGIN event.

Columns	Contain
3	the number 1
6 - 7	the starting hour for the simulation
10	the number 0
12	the number 0

The rest of the records describe explicitly generated flights, one flight per record.

Columns	Contain
3	the number 2
6 - 7	the hour the flight enters the system
9 - 10	the minute of that hour
11 - 12	the second of that minute
13 - 14	1 if flight is a takeoff, 2 if flight is a landing
15 - 16	the runway to be used by this flight
17 - 18	the aircraft type for this flight
19 - 20	the departure path if this is a takeoff

There should be an end of file after the last flight record.

A.5 Airport Data File Format

In all of the following the number of runways is 9, the number of aircraft types is 10, the number of departure paths is 5, and the number of interferences is 10. Data must be provided as if there were 9 runways, for instance, even though there might be only 2. The other 7 values may be zero. They are never used but are only place holders to space the data items properly. The variables referenced here are described in Appendix B.

record number	Contents	FORTRAN format
1	3 letter airport identifying code	A3
2	CTYPE for takeoffs followed in the same record by CTYPE for landings	20F6.3
3 - 4	LAMBD for takeoffs for each hour of the day	12F8.6
5 - 6	LAMBD for landings for each hour of the day	12F8.6
7	number of actual runways	I2
	number of actual departure paths	I2
	operational procedure for each runway	9I2
	heading code for each runway	9I2
	number of interferences for each runway	9I2

record number	Contents	FORTRAN format
8	left/right designator for each runway	9A2
9	Interference code for each runway pair (1 if landings on i interfere with landings on j but not with takeoffs, and 2 if both landings and takeoffs on one runway interfere with operations on the other)	81I1
10	RPT for each runway and interference	90I1
11	TPT for each runway and interference	90I1
12	DOM for each runway	9F6.3
13 - 15	FLYOM for each type for each runway (3 runways per record)	30F4.3
16 - 20	DINT for each pair of runways (2 runways per record)	18F5.3
21	PRWYT for each type of aircraft for runways 1 through 3	30F4.2
22	PRWYL - runways 1 - 3	30F4.2
23	PRWYT - runways 4 - 6	30F4.2
24	PRWYL - runways 4 - 6	30F4.2
25	PRWYT - runways 7 - 9	30F4.2
26	PRWYL - runways 7 - 9	30F4.2
27	PDFIX for each departure path for runways 1 through 4	30F4.2
28	PDFIX - runways 5 - 9	30F4.2

```
ATL
 .190  .770  .790  .930 1.000  .000  .000  .000  .000  .000  .190  .770  .790  .930 1.000  .000  .000  .000  .000  .000
 .090900 .333300 .500000 .500000 .250000 .500000 .500000 .090900 .050000 .041700 .040000 .038500
 .025000 .027800 .018500 .018500 .017900 .023300 .052600 .058800 .058800 .111100 .071400 .090900
 .5269009.999900 .058800 .2000009.999900 .200000 .200000 .333300 .666700 .038500 .033300 .026300
 .022700 .028600 .027800 .025000 .019600 .018900 .027800 .058800 .058800 .111100 .111100 .052600
 3 3 1 2 3 0 0 0 0 0 0 9 915 0 0 0 0 0 0 2 1 3 0 0 0 0 0 0
L R
21000000012000000000200000000000000000000000000000000000000000000000000000000000000000
33100000030200000000100000000000000000000000000000000000000000000000000000000000000000000000000
66600000050600000000500000000000000000000000000000000000000000000000000000000000000000000000000
 8.000 9.00018.000  .000  .000  .000  .000  .000  .000
.138.143.154.174.204.000.000.000.000.000.172.179.192.217.255.000.000.000.000.000.138.143.154.174.204.000.000.000.000.000
.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000
.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000
 .000 .000 .395 .000 .000 .000 .000 .000 .000 .000 .000 .987 .000 .000 .000 .000 .000 .000
 .2761.073 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000
 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000
 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000
 .000 .000 .000 .000 .000 .000 .000 .000 .000
1.001.00 .90 .90 .70 .00 .00 .00 .00 .001.001.00 .90 .90 .70 .00 .00 .00 .00 .001.001.001.001.001.00 .00 .00 .00 .00 .00
 .00 .00 .00 .00 .00 .00 .00 .00 .00 .001.001.00 .90 .90 .70 .00 .00 .00 .00 .001.001.001.001.001.00 .00 .00 .00 .00 .00
 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00
 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00
 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00
 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00
 .33 .671.00 .00 .00 .33 .671.00 .00 .00 .75 .751.00 .00 .00 .00 .00 .00 .00 .00
 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00
JFK
 .540  .890  .900  .960 1.000  .000  .000  .000  .000  .000  .540  .890  .900  .960 1.000  .000  .000  .000  .000  .000
 .090900 .333300 .500000 .500000 .250000 .500000 .500000 .090900 .050000 .041700 .040000 .038500
 .025000 .027800 .018500 .018500 .017900 .023300 .052600 .058800 .058800 .111100 .071400 .090900
 .5269009.999900 .058800 .2000009.999900 .200000 .200000 .333300 .666700 .038500 .033300 .026300
 .022700 .028600 .027800 .025000 .019600 .018900 .027800 .058800 .058800 .111100 .111100 .052600
 5 3 1 3 4 2 3 0 0 0 03131 4 432 0 0 0 0 1 5 6 2 4 0 0 0 0
L R L R
00000000000002000000020000000200000002000000000000000000000000000000000000000000000000
33132000035232000005402000005402000005400000000400000000000000000000000000000000000000000000000
66611000051622000021030000003204000004300000000400000000000000000000000000000000000000000000000
 8.000 9.00013.000 8.000 8.000  .000  .000  .000  .000
.207.214.231.261.306.000.000.000.000.000.207.214.231.261.306.000.000.000.000.000.207.214.231.261.306.000.000.000.000.000
.207.214.231.261.306.000.000.000.000.000.138.143.154.174.204.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000
.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000.000
 .000 .000 .345 .000 .000 .000 .000 .000 .000 .000 .000 .212 .000 .000 .000 .000 .000 .000
 .5641.769 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000
 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000
 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000
 .000 .000 .000 .000 .000 .000 .000 .000 .000
1.00 .98 .90 .10 .00 .00 .00 .00 .00 .001.001.001.00 .15 .00 .00 .00 .00 .00 .001.001.001.00 .15 .00 .00 .00 .00 .00 .00
 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .01 .03 .00 .00 .00 .00 .00 .00
1.001.001.00 .15 .00 .00 .00 .00 .00 .001.001.001.001.001.00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00
1.001.00 .51 .51 .00 .00 .00 .00 .00 .001.001.001.001.001.00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00
 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00
 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00
1.001.001.00 .00 .00 .901.001.00 .00 .00 .101.001.00 .00 .00 .00 .00 .00 .00 .00
 .00 .501.00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00 .00
```

APPENDIX B. MODEL ELEMENTS (ROUTINES, VARIABLES, ETC.)

This appendix is divided into two sections. The first lists each of the routines, events and functions, and provides a phrase describing what each does. The second section lists the names and descriptions of variables used in the model. The variables are listed under the headings of entities, arrays, attributes of event notices, temporary entities, attributes of temporary entities, and sets. The reader should refer to Section 3.1 for the definitions of these terms.

B.1 Routines

Exogenous Events

1. BEGIN - starts the simulation
2. EXGEN - creates explicitly generated flights

Endogenous Events

1. GEN - creates flights in a Poisson manner
2. NXTOP - finds the next operation (landing or takeoff) for a runway
3. LAND - creates tieups resulting from a landing
4. TOFF - creates tieups resulting from a takeoff
5. FTIUP - destroys tieups no longer in force
6. ENDS - prints final output
7. CHOUR - updates current hour
8. PRINT - records delay and throughput

Functions

1. PTYPE - picks an aircraft type
2. PRWAY - picks a runway

3. PDFIX - picks a departure path

4. FREER - finds the first time the current operation can proceed

B.2 Variables

Entities

1. O - operation (1) takeoff, (2) landing

2. RW - runway

3. TYP - aircraft type

4. FIX - departure path

5. H - hour

Variables and Arrays

1. LAMBD(O) - Poisson distribution parameter, the average time (in hours) between operations

2. OPER(RW) - operational procedure

   (1) takeoffs only

   (2) landings only

   (3) dual use, alternate operations

   (4) dual use, landings take precedence

3. DOM(RW) - distance from the outer marker to the runway threshold

4. NPT(RW) - greater than zero if the runway RW interferes with others

5. FLYOM(TYP) - time to fly from handoff to the outer marker

6. DME(TYP) - (0) no DME
   (1) has DME

7. VLAND(TYP) - landing speed

8. VTOFF(TYP) - liftoff speed

9. ROTL(TYP) - runway occupancy time on landing

10. ROTT(TYP) - runway occupancy time on takeoff

11. VTAXI - turnoff speed for landings

12. DAFIX - distance from departure/arrival fix to the runway threshold

13. SEPTL - distance which must separate a landing possessing DME from a preceding takeoff

14. SEPLL - required inter landing separation

15. CALIN - minimum time between the generation of a takeoff and clearance to leave its gate

16. CTYPE (0, TYPE) - cumulative distribution of aircraft mix

17. CRWYT(RW, TYP) - cumulative distribution of runway use for takeoff

18. CRWYL(RW,TYP) - cumulative distribution of runway use for landing

19. CDFIX(RW, FIX) - cumulative distribution of departure paths

20. NEXT(RW) - next type of operation on runway RW. If NEXT is zero, the next operation has not been scheduled

21. LAST(RW) - the latest operation on RW

22. DINT(RW_i, RW_j) - distance from the end of RW_i to its intersection with RW_j

23. TPT(RW, NTPT(RW)) - type of tieup on RW caused by an operation on another runway (see RPT for which other runway):

    (1) to maintain interlanding separation

    (2) to maintain a landing separated from a preceding departure

    (3) to maintain a departure separated from a following landing

    (4) to maintain departure separation between completely dependent runways

    (5) to maintain departure separation on semi-dependent runways

    (6) to maintain separation on intersecting runways

24. RPT(RW, NRPT(RW)) - runway tied up as a result of an operation on RW. Note that RPT and TPT together describe the interference between runways; RPT tells which runway is interfered with and TPT tells how.

25. TDMIN(RW) - minimum time between a takeoff's leaving its gate and starting its roll

26. SEPTT(FIX_i,FIX_j) - required separation time between a departure bound on path FIX_i and one departing the same runway bound on path FIX_j

27. SEP2(FIX_i, FIX_j) - required separation time between an aircraft departing on path FIX_i from one runway and one departing on FIX_j from a different runway where the two runways are semi-dependent

28. IHOUR - current hour

29. NARR(H) - number of landings generated

30. NDEP(H) - number of takeoffs generated

31. NLAND(H) - number of landings performed

32. NTOFF(H) - number of takeoffs performed

33. DELT(H) - total takeoff delay

34. DELL(H) - total landing delay

35. TBEG - time the accounting for delay starts

36. TEND - time the simulation ends

37. GENN(O) - the identity of the GEN currently scheduled for operation O

Attributes of Event Notices

1. RWAY - runway

2. OP - operation

3. PT - point tied up

   (1) outer marker

   (2) runway threshold

   (3) end of the runway

4. DLAY - delay for the current flight

Temporary Entities

1. FLT - flight

2. TIEUP

Attributes of Temporary Entities

1. TYPE(FLT) - aircraft type

2. DFIX(FLT) - departure path

3. TIN(FLT) - first time FLT can cross outer marker or leave its gate

4. TMIN(TIEUP) - beginning of tieup interval

5. TMAX(TIEUP) - end of tieup interval

Sets

1. Q(RW,O) - landing and takeoff queues

2. OMTI(RW) - tieups in force at the outer marker

3. THTI(RW) - tieups in force at the runway threshold

4. ERTI(RW) - tieups in force at the end of the runway

APPENDIX C

LISTING OF THE PREPROCESSOR PROGRAM

```
C     THIS PROGRAM IS A PREPROCESSOR WHICH READS IN THE DATA AND
C     PUTS IT INTO PROPER FORM FOR USE BY THE SIMULATION PROGRAM.
      PARAMETER KRW=9,KTYP=10,KFIX=5,KO=2,KTPT=10
      INTEGER OPER(KRW),DME(KTYP),RPT(KRW,KTPT),TPT(KRW,KTPT)
      REAL LAMBD(KO,24)
      DIMENSION DOM(KRW),NPT(KRW),FLYOM(KTYP,KRW),VLAND(KTYP),ROTL(KTYP)
     1,VTOFF(KTYP),ROTT(KTYP),PTYPE(KO,KTYP),PRWYT(KTYP,KRW),LAST(KRW),
     2PRWYL(KTYP,KRW),PDFIX(KRW,KFIX),DINT(KRW,KRW),INPUT(6),TDMIN(KRW),
     3SEP2(KFIX,KFIX),SEPTT(KFIX,KFIX),INTFR(KRW,KRW),CALIN(KRW),N(60),
     4IHEAD(KRW),ILR(KRW)
      IN=7
C     ------------------
C     TBEG,TEND ARE TIMES OF BEGINNING AND END OF SIMULATION
C     EXAMPLES ARE 0.00 FOR MIDNIGHT AND 17.30 FOR 5.30 P.M.
C     ------------------
      READ(5,705)TBEG,TEND
  705 FORMAT(10F7.2)
      WRITE(6,500)TBEG,TEND
  500 FORMAT('1THIS SIMULATION RUNS FROM'F6.2,' TO'F6.2///)
      TBEG=AINT(TBEG)+(TBEG-AINT(TBEG))/60.
      TEND=AINT(TEND)+(TEND-AINT(TEND))/60.
      IF (TBEG.GT.0.) GO TO 1
      TBEG=TBEG+24.
      TEND=TEND+24.
    1 IF (TEND.LE.TBEG) TEND=TEND+24.
      IBEG=TBEG-1.
      IHOUR=MOD(IBEG,24)+1
      NH=TEND-TBEG+.99
      ND=-1
      IF (NH.LT.24) GO TO 2
      II=1
      JJ=24
      GO TO 4
    2 II=IHOUR
      IEND=TEND-.01
      JJ=MOD(IEND,24)+1
      IF (JJ.LT.II) ND=JJ
C     ------------------
C     INPUT(I) - LEAVE BLANK IF ITH DATA GROUP WILL BE SUPPLIED BY USER
C                ANYTHING ELSE WILL CAUSE PROGRAM TO SUPPLY STANDARD
C                DATA.  FOR GROUPS 2,3,5,6 SET INPUT EQUAL TO AIRPORT ID
C     DATA GROUPS ARE - 1) NUMBER AND DESCRIPTION OF AIRCRAFT TYPES
C                       2) MIX OF AIRCRAFT TYPES
C                       3) LANDING AND TAKEOFF RATES, BY HOUR
C                       4) SEPARATION REQUIREMENTS
C                       5) DESCRIPTION OF AIRPORT AND ITS OPERATION
C                       6) FRACTION OF TYPES USING EACH RUNWAY AND
C                          DEPARTURE PATH
C     ------------------
    4 READ(5,730)INPUT
  730 FORMAT(6A3)
C     ------------------
C     NTYPE - NUMBER OF TYPES
C     DME(I) - SET TO 1 IF TYPE I HAS DISTANCE MEASURING EQUIPMENT
C     VTOFF(I) - TAKEOFF SPEED OF TYPE I, IN KNOTS
C     VLAND(I) - LANDING SPEED OF TYPE I, IN KNOTS
C     ROTL(I) - RUNWAY OCCUPANCY TIME ON LANDINGS FOR TYPE I, IN SECONDS
C     ROTT(I) - SAME FOR TAKEOFFS
C     VTAXI - AVERAGE TURNOFF SPEED FOR ALL TYPES
C             AND TO 0 IF NOT
C     ------------------
```

```
      IF(INPUT(1).EQ.'     ')GO TO 5
      DATA NTYPE,VLAND,VTOFF,ROTL,ROTT,VTAXI,DME/5,145.,140.,130.,115.,
     198.,5*0.,155.,148.,140.,123.,110.,5*0.,.014,.0123,.0108,.0092,
     2.0081,5*0.,.0091,.0083,.0077,.0064,.0058,5*0.,25.,3*1,7*0/
      GO TO 10
    5 READ(5,725,END=6)NTYPE,DME(NTYPE),VLAND(NTYPE),VTOFF(NTYPE),
     1                 ROTL(NTYPE),ROTT(NTYPE)
  725 FORMAT(2I3,4F7.2)
      GO TO 5
    6 READ(5,705)VTAXI
      DO 7 I=1,NTYPE
      ROTL(I)=ROTL(I)/3600.
    7 ROTT(I)=ROTT(I)/3600.
      -----------------
      PTYPE(1,I) - THE FRACTION OF TOTAL TAKEOFFS WHICH ARE OF TYPE I
      PTYPE(2,I) - SAME FOR LANDINGS
      -----------------
   10 IF(INPUT(2).EQ.'     ')GO TO 15
      REWIND IN
   11 READ(IN,600,END=300)I
  600 FORMAT(A3)
      IF(I.NE.INPUT(2))GO TO 11
      READ(IN,605)((PTYPE(I,J),J=1,KTYP),I=1,KO)
  605 FORMAT(20F6.3)
      GO TO 30
   15 DO 16 I=1,2
      READ(5,710)(PTYPE(I,J),J=1,NTYPE)
  710 FORMAT(12F6.4)
   16 CONTINUE
      DO 25 I=1,2
      DO 20 J=2,NTYPE
   20 PTYPE(I,J)=PTYPE(I,J-1)+PTYPE(I,J)
      IF(INT((PTYPE(I,NTYPE)+.01)*100.).NE.100)WRITE(6,800)I
  800 FORMAT(' WARNING - PROBABILITIES OF ALL TYPES FOR OPERATION',I2,
     1' (1-LANDING, 2-TAKEOFF) DO NOT SUM TO ONE')
   25 CONTINUE
      -----------------
      LAMBD(1,I) - AVERAGE NUMBER OF TAKEOFFS DURING ITH HOUR OF THE DAY
      LAMBD(2,I) - SAME FOR LANDINGS
      -----------------
   30 IF(INPUT(3).EQ.'     ')GO TO 35
      IF(INPUT(3).EQ.INPUT(2))GO TO 32
      REWIND IN
   31 READ(IN,600,END=300)I
      IF(I.NE.INPUT(3))GO TO 31
      READ(IN,600)I
   32 DO 33 I=1,KO
      READ(IN,610)(LAMBD(I,J),J=1,24)
   33 CONTINUE
  610 FORMAT(12F8.6)
      GO TO 40
   35 IF(ND.GT.0)GO TO 37
      DO 36 I=1,2
      READ(5,720)(LAMBD(I,J),J=II,JJ)
  720 FORMAT(12F6.1)
      DO 36 J=1,24
      IF(LAMBD(I,J).LE.0.)GO TO 351
      LAMBD(I,J)=1./LAMBD(I,J)
      GO TO 36
  351 LAMBD(I,J)=9.999
   36 CONTINUE
      GO TO 40
```

```
 37 DO 38 I=1,2
    READ(5,720)(LAMBD(I,J),J=IHOUR,24),(LAMBD(I,J),J=1,ND)
    DO 38 J=1,24
    IF(LAMBD(I,J).LE.0.)GO TO 371
    LAMBD(I,J)=1./LAMBD(I,J)
    GO TO 38
371 LAMBD(I,J)=9.999
 38 CONTINUE
C   ------------------
C   DAFIX - DISTANCE OF DEP/ARR FIX FROM RUNWAY
C   SEPLL - RADAR SEPARATION REQUIRED BETWEEN AIRCRAFT ON SAME PATH
C   SEPTL - RADAR SEPARATION REQUIRED BETWEEN A LANDING A/C AND AN A/C
C           TAKING OFF FROM THE SAME RUNWAY
C   ALL THREE DISTANCES ARE IN NAUTICAL MILES.
C   ------------------
 40 IF(INPUT(4).EQ.'    ')GO TO 45
    DAFIX=4.
    SEPTL=2.
    SEPLL=3.
    GO TO 50
 45 READ(5,705)DAFIX,SEPLL,SEPTL
C   ------------------
C   NFIX - NUMBER OF FIXES
C   IHEAD(I) - THE HEADING OF RUNWAY I
C   ILR(I) - FOR PARALLEL RUNWAYS, THE LEFT OR RIGHT DESIGNATION
C   FOR EXAMPLE, IF RUNWAY 2 IS 31R, IHEAD(2)=31 AND ILR(2)=R
C   OPER(I) - OPERATION CODE FOR RUNWAY I
C             CODES ARE 1- TAKEOFFS ONLY, 2- LANDINGS ONLY
C                       3-DUAL USE, ALTERNATING OPERATIONS
C                       4-DUAL USE, LANDINGS TAKE PRECEDENCE
C   DOM(I) - DISTANCE TO OUTER MARKER FOR RUNWAY I
C   FLYOM(I,J)- TIME, IN MINUTES, FOR A TYPE I A/C TO FLY FROM HANDOFF
C               TO OUTER MARKER OF RUNWAY J
C   DINT(I,J) - DISTANCE FROM END OF RUNWAY I TO ITS INTERSECTION
C               WITH RUNWAY J, IN FEET
C   INTFR(I,J) - INTERFERENCE CODE FOR RUNWAYS I AND J
C               CODES ARE  1- LANDINGS ON I AND J INTERFERE AND
C                             MUST BE SEPARATED, BUT SIMULTANEOUS
C                             TAKEOFFS ARE PERMITTED IF THEY DIVERGE.
C                          2- NO SIMULTANEOUS OPERATIONS PERMITTED
C               IF NEITHER, COMPLETE INDEPENDENCE IS ASSUMED.
C   ------------------
 50 IF(INPUT(5).EQ.'    ')GO TO 55
    IF(INPUT(5).EQ.INPUT(3))GO TO 54
    IF(INPUT(5).EQ.INPUT(2).AND.INPUT(3).EQ.'    ')GO TO 52
    REWIND IN
 51 READ(IN,600,END=300)I
    IF(I.NE.INPUT(5))GO TO 51
    READ(IN,600)I
 52 DO 53 J=1,4
    READ(IN,600)I
 53 CONTINUE
 54 READ(IN,615)NRW,NFIX,OPER,IHEAD,NPT
615 FORMAT(66I2)
    READ(IN,620)ILR
620 FORMAT(10A2)
    READ(IN,625)INTFR
    READ(IN,625)RPT
    READ(IN,625)TPT
625 FORMAT(132I1)
```

```
      READ(IN,605)DOM
      READ (IN,630) ((FLYOM(J,I),J=1,KTYP),I=1,KRW)
  630 FORMAT(30F4.3)
      READ (IN,635) ((DINT(I,J),J=1,KRW),I=1,KRW)
  635 FORMAT(18F5.3)
      GO TO 110
   55 READ(5,700)NFIX
      NRW=0
  551 READ(5,735,END=552)I,IHEAD(I),ILR(I),OPER(I),DOM(I)
  735 FORMAT(I2,I4,A2,I4,F7.2)
      NRW=NRW+1
      GO TO 551
  552 DO 56 I=1,NRW
      READ(5,705)(FLYOM(J,I),J=1,NTYPE)
      DO 56 J=1,NRW
      DINT(I,J)=0.
   56 CONTINUE
   57 READ(5,715,END=571)I,J,DINT(I,J),DINT(J,I)
  715 FORMAT(2I2,2F8.0)
      GO TO 57
  571 READ(5,700,END=572)I,J,INTFR(I,J)
  700 FORMAT(10I2)
      GO TO 571
  572 DO 59 I=1,NRW
      DO 58 J=1,NTYPE
   58 FLYOM(J,I)=FLYOM(J,I)/60.
      DO 59 J=1,NRW
   59 DINT(I,J)=DINT(I,J)/6076.
      DO 60 I=1,NRW
      DO 60 J=1,NRW
      IF(INTFR(I,J).EQ.0)GO TO 60
      INTFR(J,I)=INTFR(I,J)
   60 CONTINUE
      DO 90 I=1,NRW
      K=0
      DO 80 J=1,NRW
      IF(INTFR(I,J).EQ.0)GO TO 80
      IF(OPER(I).LT.2.OR.OPER(J).LT.2)GO TO 65
      K=K+1
      RPT(I,K)=J
      TPT(I,K)=1
   65 IF(OPER(I).EQ.2.OR.OPER(J).EQ.2)GO TO 70
      K=K+1
      RPT(I,K)=J
      TPT(I,K)=4
      IF(INTFR(I,J).EQ.1)TPT(I,K)=5
   70 IF(INTFR(I,J).EQ.1)GO TO 80
      IF(OPER(I).EQ.2.OR.OPER(J).EQ.1)GO TO 75
      K=K+1
      RPT(I,K)=J
      TPT(I,K)=3
   75 IF(OPER(I).EQ.1.OR.OPER(J).EQ.2)GO TO 80
      K=K+1
      RPT(I,K)=J
      TPT(I,K)=2
   80 CONTINUE
```

```
      DO 85 J=1,NRW
      IF(DINT(I,J).LE.0.)GO TO 85
      K=K+1
      RPT(I,K)=J
      TPT(I,K)=6
   85 CONTINUE
      NPT(I)=K
      K=K+1
      DO 86 L=K,10
      RPT(I,L)=0
      TPT(I,L)=0
   86 CONTINUE
   90 CONTINUE
      -------------------
      PRWYT(I,J) - THE FRACTION OF ALL TAKEOFFS OF TYPE I A/C
      WHICH USE RUNWAY J
      PRWYL(I,J) - THE SAME FOR LANDINGS
      PDFIX(I,J) - THE FRACTION OF ALL TAKEOFFS FROM RUNWAY I
                   WHICH ARE GOING TO DEPARTURE FIX J
      -------------------
  110 IF(INPUT(6).EQ.'    ')GO TO 115
      IF(INPUT(6).EQ.INPUT(5))GO TO 1142
      IF(INPUT(6).EQ.INPUT(3).AND.INPUT(5).EQ.'    ')GO TO 114
      IF(INPUT(6).EQ.INPUT(2).AND.INPUT(3).EQ.'    '.AND.INPUT(5).EQ.
     1'    ')GO TO 112
      REWIND IN
  111 READ(IN,600,END=300)I
      IF(I.NE.INPUT(6))GO TO 111
      READ(IN,600)I
  112 DO 113 J=1,4
      READ(IN,600)I
  113 CONTINUE
  114 DO 1141 J=1,14
      READ(IN,600)I
 1141 CONTINUE
 1142 DO 1143 II=1,KRW,3
      JJ=II+2
      READ(IN,640)((PRWYT(I,J),I=1,10),J=II,JJ)
      READ(IN,640)((PRWYL(I,J),I=1,10),J=II,JJ)
 1143 CONTINUE
      READ (IN,640) ((PDFIX(I,J),J=1,KFIX),I=1,4)
      READ (IN,640) ((PDFIX(I,J),J=1,KFIX),I=5,KRW)
  640 FORMAT(30F4.2)
      GO TO 145
  115 DO 120 J=1,NTYPE
  120 READ(5,710)(PRWYT(J,I),I=1,NRW),(PRWYL(J,I),I=1,NRW)
      DO 121 I=1,NRW
      READ(5,710)(PDFIX(I,J),J=1,NFIX)
  121 CONTINUE
      DO 122 I=1,NRW
      DO 122 J=1,NTYPE
      IF(PRWYT(J,I).GT. 0. .AND. OPER(I).EQ.2)WRITE(6,830)J,I
  830 FORMAT(' WARNING - THERE IS A POSITIVE PROBABILITY THAT A TYPE',
     1I2,' AIRCRAFT WILL TAKE OFF FROM'/12X,'RUNWAY',I2,', WHICH HANDLES
     2 LANDINGS ONLY')
      IF(PRWYL(J,I).GT.0..AND.OPER(I).EQ.1)WRITE(6,840)J,I
  840 FORMAT(' WARNING - THERE IS A POSITIVE PROBABILITY THAT A TYPE',
     1I2,' AIRCRAFT WILL LAND ON '/12X,'RUNWAY',I2,', WHICH HANDLES TAKE
     2OFFS ONLY')
  122 CONTINUE
```

```
      DO 124 I=1,NRW
      DO 123 J=2,NFIX
  123 PDFIX(I,J)=PDFIX(I,J-1)+PDFIX(I,J)
      IF(INT((PDFIX(I,NFIX )+.01)*100.).NE.100)WRITE(6,850)I
  850 FORMAT(' WARNING - PROBABILITIES OF TAKEOFFS FROM RUNWAY',I2,
     1' GOING TO ALL FIXES DO NOT SUM TO ONE')
  124 CONTINUE
      IF(NRW.EQ.1)GO TO 139
      DO 130 I=1,NTYPE
      DO 125 J=2,NRW
      PRWYL(I,J)=PRWYL(I,J-1)+PRWYL(I,J)
  125 PRWYT(I,J)=PRWYT(I,J-1)+PRWYT(I,J)
      IF(INT((PRWYL(I,NRW   )+.01)*100.).NE.100)WRITE(6,810)I
  810 FORMAT(' WARNING - PROBABILITIES OF A TYPE',I2,' AIRCRAFT LANDING
     1ON ALL RUNWAYS DO NOT SUM TO ONE')
      IF(INT((PRWYT(I,NRW   )+.01)*100.).NE.100)WRITE(6,820)I
  820 FORMAT(' WARNING - PROBABILITIES OF A TYPE',I2,' AIRCRAFT TAKING
     1OFF FROM ALL RUNWAYS DO NOT SUM TO ONE')
  130 CONTINUE
  145 DO 150 I=1,60
  150 N(I)=I
      DO 95 I=1,NRW
      TDMIN(I)=(DOM(I)/VLAND(1))*PTYPE(2,1)
      DO 95 J=2,NTYPE
   95 TDMIN(I)=(DOM(I)/VLAND(J))*(PTYPE(2,J)-PTYPE(2,J-1))+TDMIN(I)
      DO 100 I=1,NRW
      CALIN(I)=FLYOM(I,1)*PTYPE(2,1)
      DO 100 J=2,NTYPE
  100 CALIN(I)=FLYOM(I,J)*(PTYPE(2,J)-PTYPE(2,J-1))+CALIN(I)
      DO 105 I=1,NRW
      LAST(I)=2
      IF(OPER(I).EQ.1)LAST(I)=1
  105 CONTINUE
  139 DO 140 I=1,NFIX
      DO 140 J=1,NFIX
      SEPTT(I,J)=.0167
      IF(I.EQ.J)SEPTT(I,J)=.05
      SEP2(I,J)=0.
      IF(I.EQ.J)SEP2(I,J)=.0334
  140 CONTINUE
      L=0
      M=8
      WRITE(M,900)N(1),N(2)
      WRITE(M,900)N(2),NRW
      WRITE(M,900)N(3),NTYPE
      WRITE(M,900)N(4),NFIX
      WRITE(M,900)N(5),N(24)
      WRITE(M,920)N(6),NRW,N(2),(OPER(I),I=1,NRW)
      WRITE(M,930)N(7),NRW,N(2),(DOM(I),I=1,NRW)
      WRITE(M,920)N(8),NRW,N(2),(NPT(I),I=1,NRW)
      DO 155 I=1,NRW
      IF(NPT(I).EQ.0)NPT(I)=1
  155 CONTINUE
      WRITE(M,940)N(9),NTYPE,N(3),NRW,N(2),((FLYOM(J,I),I=1,NRW),
     1J=1,NTYPE)
      WRITE(M,920)N(10),NTYPE,N(3),(DME(I),I=1,NTYPE)
      WRITE(M,930)N(11),NTYPE,N(3),(VLAND(I),I=1,NTYPE)
      WRITE(M,930)N(12),NTYPE,N(3),(VTOFF(I),I=1,NTYPE)
      WRITE(M,930)N(13),NTYPE,N(3),(ROTL(I),I=1,NTYPE)
      WRITE(M,930)N(14),NTYPE,N(3),(ROTT(I),I=1,NTYPE)
```

```
      WRITE(M,910)N(15),VTAXI
      WRITE(M,910)N(16),DAFIX
      WRITE(M,910)N(17),SEPLL
      WRITE(M,910)N(18),SEPTL
      WRITE(M,930)N(19),NRW,N(2),(CALIN(I),I=1,NRW)
      WRITE(M,940)N(20),N(2),N(1),NTYPE,N(3),((PTYPE(I,J),J=1,NTYPE),
     1I=1,2)
      WRITE(M,940)N(21),NTYPE,N(3),NRW,N(2),((PRWYT(I,J),J=1,NRW),
     1I=1,NTYPE)
      WRITE(M,940)N(22),NTYPE,N(3),NRW,N(2),((PRWYL(I,J),J=1,NRW),
     1I=1,NTYPE)
      WRITE(M,940)N(23),NRW,N(2),NFIX,N(4),((PDFIX(I,J),J=1,NFIX),
     1I=1,NRW)
      WRITE(M,950)N(24),N(24),N(1),NRW,N(2)
      WRITE(M,920)N(25),NRW,N(2),(LAST(I),I=1,NRW)
      WRITE(M,940)N(26),NRW,N(2),NRW,N(2),((DINT(I,J),J=1,NRW),I=1,NRW)
      WRITE(M,950)N(27),N(28),N(2),NRW,N(2),N(2),N(1)
      WRITE(M,960)N(29),NRW,N(2),((RPT(I,J),J=1,10),NPT(I),I=1,NRW)
      WRITE(M,960)N(30),NRW,N(2),((TPT(I,J),J=1,10),NPT(I),I=1,NRW)
      WRITE(M,930)N(31),NRW,N(2),(TDMIN(I),I=1,NRW)
      WRITE(M,940)N(32),NFIX,N(4),NFIX,N(4),((SEPTT(I,J),J=1,NFIX),
     1I=1,NFIX)
      WRITE(M,950)N(33),N(38),N(1),N(24),N(5)
      WRITE(M,910)N(39),TBEG
      WRITE(M,950)N(40),N(45),N(1),NRW,N(2)
      WRITE(M,910)N(46),TEND
      WRITE(M,940)N(47),N(2),N(1),N(24),N(5),((LAMBD(I,J),J=1,24),I=1,2)
      WRITE(M,900)N(48),IHOUR
      WRITE(M,940)N(49),NFIX,N(4),NFIX,N(4),((SEP2(I,J),J=1,NFIX),
     1I=1,NFIX)
      WRITE (M,950) N(50),N(50),N(1),N(2),N(1)
      WRITE(M,970)
      WRITE(M,980)N(1),IBEG,L,L
      ENDFILE M
  900 FORMAT(I4,5X,'0 R',I39)
  910 FORMAT(I4,5X,'0 R',37XF7.3)
  920 FORMAT(I4,5X,'1 R',I6,I4,27X,'10(I2)'/(10I2))
  930 FORMAT(I4,5X,'1 R',I6,I4,27X,'10(D3.4)'/(10F8.4))
  940 FORMAT(I4,5X,'2 R',I6,3I4,5X,'R  F',11X,'12(D1.4)'/(12F6.4))
  950 FORMAT(2I4,I2,'  Z',I5,3I4)
  960 FORMAT(I4,5X,'2 R',I6,I4,18X,'R',8X,'10(I2)'/(10I2,50XI2))
  970 FORMAT(' ')
  980 FORMAT(I3,I4,I3,I2)
      WRITE(6,505)
  505 FORMAT(//35X'AIRCRAFT DESCRIPTION'//16X'TYPE'21X'SPEEDS (KNOTS)'6X
     1'RUNWAY OCCUPANCY (SECONDS)'/39X'LANDING'4X'LIFTOFF'6X'LANDING'6X
     2'TAKEOFF'//)
      DO 160 I=1,NTYPE
      RROTL=ROTL(I)*3600.
      RROTT=ROTT(I)*3600.
      WRITE(6,510)I,VLAND(I),VTOFF(I),RROTL,RROTT
  510 FORMAT('0'16XI2,21XF4.0,7XF4.0,10XF4.0,9XF4.0)
  160 CONTINUE
      WRITE(6,511)
  511 FORMAT(//35X'TRAFFIC DESCRIPTION'//31X'TYPE'5X'LANDING'4X'TAKEOFF'
     1/42X'MIX'8X'MIX'//)
      A=PTYPE(2,1)*100.
      B=PTYPE(1,1)*100.
      WRITE(6,515)N(1),A,B
  515 FORMAT(32XI2,2(7XF4.0)/)
```

```
      DO 165 I=2,NTYPE
      A=(PTYPE(2,I)-PTYPE(2,I-1))*100.
      B=(PTYPE(1,I)-PTYPE(1,I-1))*100.
      WRITE(6,515)I,A,B
  165 CONTINUE
      WRITE(6,520)
  520 FORMAT('1AIRPORT CONFIGURATION')
      IF(INPUT(5).NE.'    ')WRITE(6,521)INPUT(5)
  521 FORMAT(' FOR 'A3,' AIRPORT')
      WRITE(6,525)NRW
  525 FORMAT(///' NUMBER OF RUNWAYS ='I2)
      DO 190 I=1,NRW
      J=OPER(I)
      GO TO (170,175,180,185),J
  170 WRITE(6,530)I,IHEAD(I),ILR(I)
  530 FORMAT('0RUNWAY'I2,' ('I2,A2,') - TAKEOFFS ONLY')
      GO TO 190
  175 WRITE(6,535)I,IHEAD(I),ILR(I)
  535 FORMAT('0RUNWAY'I2,' ('I2,A2,') - LANDINGS ONLY')
      GO TO 190
  180 WRITE(6,540)I,IHEAD(I),ILR(I)
  540 FORMAT('0RUNWAY'I2,' ('I2,A2,') - DUAL USE, ALTERNATING OPERATIONS
     1')
      GO TO 190
  185 WRITE(6,545)I,IHEAD(I),ILR(I)
  545 FORMAT('0RUNWAY'I2,' ('I2,A2,') - DUAL USE, LANDINGS TAKE PRECEDEN
     1CE')
  190 CONTINUE
      DO 195 I=1,NRW
      DO 195 J=1,NRW
      IF(DINT(I,J).LE.0.)GO TO 195
      A=DINT(I,J)*6076.
      B=DINT(J,I)*6076.
      WRITE(6,550)IHEAD(I),ILR(I),IHEAD(J),ILR(J),A,IHEAD(I),ILR(I),B,
     1IHEAD(J),ILR(J)
  550 FORMAT(//'0RUNWAYS'I3,A2,' AND'I3,A2,' INTERSECT AT A POINT'F7.0,
     1' FEET FROM THE'/' END OF RUNWAY'I3,A2,' AND'F7.0,' FEET FROM THE
     2END OF RUNWAY'I3,A2,'.')
  195 CONTINUE
      II=NRW-1
      DO 215 I=1,II
      JJ=I+1
      DO 215 J=JJ,NRW
      IF(INTER(I,J).EQ.0.AND.IHEAD(I).NE.IHEAD(J))GO TO 215
      L=INTER(I,J)+1
      IF(IHEAD(I).NE.IHEAD(J))GO TO 211
      GO TO(200,205,210),L
  200 WRITE(6,555)IHEAD(I),ILR(I),IHEAD(J),ILR(J)
  555 FORMAT(/'0RUNWAYS'I3,A2,' AND'I3,A2,' ARE INDEPENDENT PARALLELS -'
     1/' SIMULTANEOUS OPERATIONS ARE PERMITTED')
      GO TO 215
  205 WRITE(6,560)IHEAD(I),ILR(I),IHEAD(J),ILR(J)
  560 FORMAT(/'0RUNWAYS'I3,A2,' AND'I3,A2,' ARE SEMI-DEPENDENT PARALLELS
     1 -'/' SIMULTANEOUS ARRIVALS ARE PROHIBITED')
      GO TO 215
  210 WRITE(6,565)IHEAD(I),ILR(I),IHEAD(J),ILR(J)
  565 FORMAT(/'0RUNWAYS'I3,A2,' AND'I3,A2,' ARE DEPENDENT PARALLELS -'/
     1' NO SIMULTANEOUS OPERATIONS ARE PERMITTED')
      GO TO 215
```

```
11    GO TO(215,212,213),L
 212 WRITE(6,570)IHEAD(I),ILR(I),IHEAD(J),ILR(J)
 570 FORMAT(/'0RUNWAYS'I3,A2,' AND'I3,A2,' ARE SEMI-DEPENDENT -'/' SIMU
    1LTANEOUS ARRIVALS ARE PROHIBITED')
     GO TO 215
 213 WRITE(6,575)IHEAD(I),ILR(I),IHEAD(J),ILR(J)
 575 FORMAT(/'0RUNWAYS'I3,A2,' AND'I3,A2,' ARE DEPENDENT -'/' NO SIMULT
    1ANEOUS OPERATIONS ARE PERMITTED')
 215 CONTINUE
     STOP
 300 WRITE(6,860)
 860 FORMAT(' WARNING - DATA TAPE DOES NOT CONTAIN INFORMATION ABOUT TH
    1E REQUESTED AIRPORT')
     STOP
     END
```

APPENDIX D

LISTING OF THE SIMULATION PROGRAM

```
N GEN   4        N RWAY   31  2  I    1O        E          Q     2  F         PTYPEI
N NXTOP4         N OP     32  2  I    2RW       E                           PRWAYI
N LAND  4                             3TYP      E                           PDFIXI
N TOFF  4                             4FIX      E                           FREERF
                 N PT     4      I    5H        E
N FTIUP4         T TYPE   1      I    6OPER     1     I
N PRINT4         N DLAY   4      F    7DOM      1     F
T FLT   4        T TIN    3      F    8NPT      1     I
T TIEUP4                              9FLYOM    2     F
N ENDS  2        T TMIN   1      F   10DME      1     I
N CHOUR2         T TMAX   2      F   11VLAND    1     F
                 T DFIX   2      I   12VTOFF    1     F
                                     13ROTL     1     F
                                     14ROTT     1     F
                 T SQ     4      I   15VTAXI    0     F
                                     16DAFIX    0     F
                                     17SEPLL    0     F
                                     18SEPTL    0     F
                                     19CALIN    1     F
                                     20CTYPE    2     F
                                     21CRWYT    2     F
                                     22CRWYL    2     F
                                     23CDFIX    2     F
                                     24NEXT     1     I
                                     25LAST     1     I
                                     26DINT     2     F
                                     27FQ       2     I
                                     28LQ       2     I
                                     29RPT      2     I
                                     30TPT      2     I
                                     31TDMIN    1     F
                                     32SEPTT    2     F
                                     33NARR     1     I
                                     34NDEP     1     I
                                     35NLAND    1     I
                                     36NTOFF    1     I
                                     37DELT     1     F
                                     38DELL     1     F
                                     39TBEG     0     F
                  T POMTI  3      I  40FOMTI    1     I      OMTI1    RTMAX    L
                  T SOMTI  4      I  41LOMTI    1     I      THTI1    RTMAX    L
                  T PTHTI  3      I  42FTHTI    1     I      ERTI1    RTMAX    L
                  T STHTI  4      I  43LTHTI    1     I
                  T PERTI  3      I  44FERTI    1     I
                  T SERTI  4      I  45LERTI    1     I
                                     46TEND     0     F
                                     47LAMBD    2     F
                                     48IHOUR    0     I
                                     49SEP2     2     F
                                     50GENN     1     I
```

```
EVENTS
  2 EXOGENOUS
    BEGIN (1)
    EXGEN (2)
  8 ENDOGENOUS
    GEN
    NXTOP
    LAND
    TOFF
    FTIUP
    CHOUR
    PRINT
    ENDS
END  EVENT LIST
```

```
    EXOGENOUS EVENT BEGIN
    DO TO 1, FOR EACH O I
    CREATE GEN
    STORE GEN IN GENN(I)
    STORE I IN OP(GEN)
    LET T=TIME-LAMBD(I,IHOUR)*ALOG(1.-RANDM)
    CAUSE GEN AT T
1   LOOP
    CREATE ENDS
    CAUSE ENDS AT TEND
    CREATE CHOUR
    CAUSE CHOUR AT TIME+1.
    RETURN
    END BEGIN
```

```
      ENDOGENOUS EVENT GEN
C     GEN  GENERATES LANDINGS AND TAKEOFFS AND
C     ASSIGNS ATTRIBUTES TO THEM.
C     OP(GEN) IS 1- TAKEOFF , OR 2-LANDING
      STORE OP(GEN) IN I
C     GEN SCHEDULES ITSELF TO OCCUR AGAIN AFTER A TIME INTERVAL
C     DEPENDING ON THE RATE OF OPERATION.
      CAUSE GEN AT TIME-LAMBD(I,IHOUR)*ALOG(1.-RANDM)
      CREATE FLT
      LET IT=PTYPE(I)
      STORE IT IN TYPE(FLT)
      LET K=PRWAY(I,IT)
      GO TO (1,2),I
C     TIN(FLT)  IS THE TIME A FLIGHT IS AVAILABLE TO BEGIN FINAL
C     DESCENT (FOR LANDINGS) OR TO BEGIN TAXIING TO RUNWAY(FOR TAKEOFF)
C     CALIN IS A TIME LAG INTRODUCED IN TAKEOFFS, CORRESPONDING TO
C     FLYOM - THE TIME A LANDING TAKES TO FLY FROM
C     HANDOFF TO THE OUTER MARKER.
 1    LET TIN(FLT)=TIME+CALIN(K)
      LET DFIX(FLT)=PDFIX(K)
      IF TIME GE TBEG, LET NDEP(IHOUR)=NDEP(IHOUR)+1
      GO TO 3
 2     LET TIN(FLT)=TIME+FLYOM(IT,K)
      IF TIME GE TBEG, LET NARR(IHOUR)=NARR(IHOUR)+1
 3     IF Q(K,I) IS NOT EMPTY, GO TO 4
      CREATE NXTOP
      LET RWAY(NXTOP)=K
      CAUSE NXTOP AT TIN(FLT)
C     Q(K,I) - IS THE QUEUE OF PLANES WAITING TO TAKFOFF(I=1),
C     OR LAND(I=2) ON RUNWAY K.
 4    FILE FLT IN Q(K,I)
      RETURN
      END GEN


      EXOGENOUS EVENT EXGEN
C     EXGEN GENERATES EXPLICIT DEPARTURES AND ARRIVALS
      SAVE EVENT CARD
      CREATE FLT
      READ I,K,IT,DFIX(FLT)
      FORMAT (4I2)
      STORE IT IN TYPE(FLT)
      GO TO (1,2),I
 1    LET TIN(FLT)=TIME+CALIN(K)
      IF TIME GE TBEG, LET NDEP(IHOUR)=NDEP(IHOUR)+1
      GO TO 3
 2    LET TIN(FLT)=TIME+FLYOM(IT,K)
      IF TIME GE TBEG, LET NARR(IHOUR)=NARR(IHOUR)+1
 3    IF  Q(K,I) IS NOT EMPTY, GO TO 4
      CREATE NXTOP
      LET RWAY(NXTOP)=K
      CAUSE NXTOP AT TIN(FLT)
 4    FILE FLT IN Q(K,I)
      RETURN
      END
```

```
      FUNCTION PTYPE(I)
C     PTYPE CHOOSES AN AIRCRAFT TYPE FOR EACH FLIGHT ACCORDING  TO
C     CTYPF - THE CUMULATIVE DISTRIBUTION OF A/C TYPFS IN THE MIX.
      LET R=RANDM
      DO TO 1, FOR EACH TYP J
      IF R LE CTYPF(I,J), GO TO 2
 1     LOOP
 2    LET PTYPF=J
      RETURN
      END PTYPE
```

```
      FUNCTION PRWAY(I,M)
C     PRWAY CHOOSES A RUNWAY FOR EACH FLIGHT ACCORDING TO
C     CRWYL - THE CUMULATIVE DISTRIBUTION OF PRWYL(SEE PREPROCESSOR).OR
C     CRWYT - SAME AS CRWYL FOR TAKEOFFS.
      LET R=RANDM
      DO TO 3, FOR EACH RW J
      GO TO (1,2),I
 1    IF R LE CRWYT(M,J), GO TO 4
      GO TO 3
 2    IF R LE CRWYL(M,J), GO TO 4
 3    LOOP
 4    LET PRWAY=J
      RETURN
      END PRWAY
```

```
      FUNCTION PDFIX(K)
C     PDFIX CHOOSES A DEPARTURE FIX FOR EACH TAKEOFF ACCORDING TO
C     CDFIX - THE CUMULATIVE DISTRIBUTION OF PDFIX (SEE PREPROCESSOR)
      LET R=RANDM
      DO TO 1, FOR EACH FIX J
      IF R LE CDFIX(K,J), GO TO 2
 1     LOOP
 2    LET PDFIX=J
      RETURN
      END PDFIX
```

```
        ENDOGENOUS EVENT NXTOP
        NXTOP DECIDES WHICH OPERATION WILL BE SCHEDULED NEXT
        ON EACH RUNWAY.
        STORE RWAY(NXTOP) IN K
        DESTROY NXTOP
        IF NEXT NE 0, THE NEXT OPERATION HAS ALREADY BEEN DECIDED UPON.
        IF NEXT(K) NE 0, RETURN
        STORE OPER(K) IN I
        IF I GE 3, GO TO 1
        IF RUNWAY HANDLES ONLY ONE OPERATION, FIND NEXT FLIGHT
        WAITING IN THE QUEUE AND SCHEDULE IT.
        FIND FIRST, FOR EACH FLT IN Q(K,I),IF NONE, RETURN
        STORE FLT IN F
        LET T=FREER(K,I,F)
        GO TO 4
        IF RUNWAY HANDLES BOTH OPERATIONS IN ALTERNATION, LOOK FOR
        NEXT FLIGHT WAITING TO PERFORM THE ALTERNATE OPERATION.
1       LET J=LAST(K)
        IF LANDINGS TAKE PRECEDENCE, ALWAYS CONSIDER THE
        LAST OPERATION TO HAVE BEEN A TAKEOFF.
        IF I EQ 4, LET J=1
        LET T=-1.
        LET IFLAG=0
        LET I=1
        IF J EQ 1, LET I=2
        LET TT=999999.
        FIND FIRST, FOR EACH FLT IN Q(K,I),IF NONE, GO TO 2
        STORE FLT IN F
        LET IFLAG=1
        LET T=FREER(K,I,F)
        IF TIN(F) LS T, GO TO 4
        LET TT=T
        LET I1=I
2       LET I=J
        IF NO FLIGHT AVAILABLE NOW, SEARCH THE OTHER QUEUE.
        FIND FIRST, FOR EACH FLT IN Q(K,I),IF NONE, GO TO 3
        STORE FLT IN F
        LET IFLAG=1
        LET T=FREER(K,I,F)
        IF TIN(F) LS T, GO TO 4
3       IF IFLAG EQ 0, RETURN
        IF NO FLIGHT THERE EITHER, CHOOSE THE FLIGHT WHICH
        WILL BE AVAILABLE EARLIEST.
        IF T LT TT, GO TO 4
        LET T=TT
        LET I=I1
4       LET NEXT(K)=I
        GO TO (5,6), I
5       CREATE TOFF
        STORE K IN RWAY(TOFF)
        CAUSE TOFF AT T
        RETURN
6       CREATE LAND
        STORE K IN RWAY(LAND)
        CAUSE LAND AT T
        RETURN
        END NXTOP
```

```
      FUNCTION FREER(K,I,FLT)
C     FREER CALCULATES THE FIRST TIME AT WHICH FLIGHT FLT CAN PERFORM
C     OPERATION I ON RUNWAY K WITHOUT VIOLATING SEPARATION RULES.
      DIMENSION TR(25)
      LET J=0
       LET T=TIME
      IF TIN(FLT) GT T, LET T=TIN(FLT)
      LET M=TYPE(FLT)
      LET FREER=T
      IF I EQ 2, GO TO 4
      IF ERTI(K) IS EMPTY, RETURN
C     ERTI(K) IS THE SET OF 'TIEUPS' FOR THE END OF THE RUNWAY K.
C     A TIEUP IS A TIME INTERVAL DURING WHICH NO TAKEOFF MAY OCCUPY
C     THE END OF THE RUNWAY DUE TO INTERFERENCE FROM OTHER AIRCRAFT.
      LET TEM=TDMIN(K)
C     TDMIN IS A TIME LAG INTRODUCED INTO THE SCHEDULE OF A TAKEOFF
C     CORRESPONDING TO THE TIME IT TAKES A LANDING TO FLY FROM THE
C     OUTER MARKER TO TOUCHDOWN.  IT MAY BE LOOSELY THOUGHT OF AS
C     TAXIING TIME.
      DO TO 3,FOR EACH TIEUP IN ERTI(K)
      LET TT=TMAX(TIEUP)-TEM
C     THE END OF THE TIEUP, I.E. THE TIME WHEN THE END OF THE RUNWAY
C     BECOMES FREE, IS DISPLACED BACKWARDS TO GIVE THE TIME WHEN
C     THE TAKEOFF MAY BEGIN TAXI.
      IF TT LS T , GO TO 3
      LET J=J+1
      LET TR(J)=TT
 3    LOOP
      GO TO 12
    4 IF OMTI(K) IS EMPTY, GO TO 8
C     OMTI IS THE SET OF TIEUPS FOR THE OUTER MARKER.
      DO TO 5, FOR EACH TIEUP IN OMTI(K)
      LET TT=TMAX(TIEUP)
      IF TT LS T , GO TO 5
      LET J=J+1
      LET TR(J)=TT
 5    LOOP
    8 IF THTI(K) IS EMPTY, GO TO 12
      LET TEM=DOM(K)/VLAND(M)
C     THTI IS THE SET OF TIEUPS FOR THE THRESHOLD OF THE RUNWAY.
      DO TO 9, FOR EACH TIEUP IN THTI(K)
C     THRESHOLD TIEUPS ARE DISPLACED BACKWARDS TO GIVE THE TIME THAT
C     THE LANDING MAY PASS THE OUTER MARKER.
      LET TT=TMAX(TIEUP)-TEM
      IF TT LS T , GO TO 9
      LET J=J+1
      LET TR(J)=TT
 9    LOOP
 12   IF J EQ 0, RETURN
      LET FREER=TR(1)
      IF J EQ 1, RETURN
C     FREER IS SET EQUAL TO THE END OF THE LATEST TIEUP, WHEN THERE IS
C     NO LONGER ANY INTERFERENCE.
      DO TO 21, FOR JJ=(1)(J)
      IF TR(JJ) GT FREER, LET FREER=TR(JJ)
 21   LOOP
      RETURN
      END FREER
```

```
ENDOGENOUS EVENT LAND
LAND CREATES ALL THE 'TIEUPS' WHICH
RESULT FROM AN AIRCRAFT LANDING.
STORE RWAY(LAND) IN K
IF Q(K,2) IS NOT EMPTY, GO TO 9
WRITE ON TAPE 6, TIME,K
FORMAT (' AT TIME',M3.2.2,S2,'LANDING QUEUE FOR RUNWAY',I3,S2,
1   'IS EMPTY')
STOP

FIND THE LANDING TO BE 'SCHEDULED' AND STORE ITS ATTRIBUTES.
REMOVE FIRST FLT FROM Q(K,2)
STORE FLT IN FL
LET M=TYPE(FL)
LET V=VLAND(M)
LET T = FREEB(K,2,FL)
LET TD=T+DOM(K)/V
TIE UP THRESHOLD TO LANDING AIRCRAFT FROM TOUCHDOWN TIME UNTIL
AFTER RUNWAY OCCUPANCY TIME HAS ELAPSED.
CREATE TIEUP
LET TMIN(TIEUP)=TD
LET TMAX(TIEUP)=TD+ROTL(M)
FILE TIEUP IN THTI(K)
CREATE FTIUP
STORE K  IN RWAY(FTIUP)
STORE 2 IN PT(FTIUP)
CAUSE FTIUP AT TMAX(TIEUP)
TIE UP END OF RUNWAY TO DEPARTING AIRCRAFT FROM TOUCHDOWN UNTIL
AFTER RUNWAY OCCUPANCY TIME HAS ELAPSED.
CREATE TIEUP
LET TMIN(TIEUP)=TD
LET TMAX(TIEUP)=TD+ROTL(M)
FILE TIEUP IN ERTI(K)
CREATE FTIUP
STORE K  IN RWAY(FTIUP)
STORE 3 IN PT(FTIUP)
CAUSE FTIUP AT TMAX(TIEUP)
FIND THE FOLLOWING PLANE IN THE LANDING QUEUE
AND STORE ITS ATTRIBUTES.
FIND FIRST, FOR EACH FLT IN Q(K,2),IF NONE, GO TO 11
STORE FLT IN F
LET MM=TYPE(F)
LET S=VLAND(MM)
CREATE A TIEUP WHICH WILL MAINTAIN PROPER RADAR SEPARATION
BETWEEN ARRIVING AIRCRAFT.
CREATE TIEUP
IF S GE V, GO TO 20
IF THE LANDING SPEED OF THE PLANE BEING 'SCHEDULED' IS GREATER
THAN THAT OF THE FOLLOWING PLANE, TIE UP THE OUTER MARKER FROM
THE TIME THE FIRST PLANE PASSES THE OUTER MARKER UNTIL
THE TIME IT TAKES THE SECOND TO FLY THE SEPARATION DISTANCE
HAS ELAPSED.
LET TMIN(TIEUP)=T
LET TMAX(TIEUP)=T+SEPLL/S
FILE TIEUP IN OMTI(K)
CREATE FTIUP
STORE K  IN RWAY(FTIUP)
```

```
    STORE 1  IN PT(FTIUP)
    CAUSE FTIUP AT TMAX(TIEUP)
    GO TO 11
    IF THE LANDING SPEED OF THE FOLLOWING PLANE IS GREATER, TIE UP
    THE THRESHOLD FROM TOUCHDOWN OF THE FIRST UNTIL THE TIME IT TAKES
    THE SECOND TO FLY THE SEPARATION DISTANCE HAS ELAPSED.
 20 LET TMIN(TIEUP) = TD
    LET TMAX(TIEUP)=TD+SEPLL/S
    FILE TIEUP IN THTI(K)
    CREATE FTIUP
    STORE K  IN RWAY(FTIUP)
    STORE 2 IN PT(FTIUP)
    CAUSE FTIUP AT TMAX(TIEUP)
 11 IF OPER(K) LT 3, GO TO 2
    CREATE A TIEUP WHICH WILL MAINTAIN PROPER SEPARATION BETWEEN
    THIS LANDING AND A TAKEOFF ON THE SAME RUNWAY.
    CREATE TIEUP
    LET TMAX(TIEUP)=TD
    IF DME(M) GT 0, GO TO 1
    IF THE LANDING HAS NO DISTANCE MEASURING EQUIPMENT, TIE UP THE
    END OF RUNWAY TO DEPARTURES FROM THE TIME THE LANDING PASSES THE
    D/A FIX UNTIL TOUCHDOWN.
    LET TMIN(TIEUP)=T+(DOM(K)-DAFIX)/V
    GO TO 101
    IF THE LANDING HAS DME, FIND THE TAKEOFF SPEED OF THE
    DEPARTURE AND COMPUTE THE TIEUP NECESSARY TO MAINTAIN SEPARATION.
1    FIND FIRST, FOR EACH FLT IN Q(K,1), IF NONE, GO TO 2
    STORE FLT IN F
    LET MM=TYPE(F)
    LET S=VTOFF(MM)
    LET TMIN(TIEUP)=T+(DOM(K)-(SEPTL+.5*V**2/S*ROTT(MM)))/V
101 FILE TIEUP IN ERTI(K)
    CREATE FTIUP
    STORE K  IN RWAY(FTIUP)
    STORE 3 IN PT(FTIUP)
    CAUSE FTIUP AT TMAX(TIEUP)
2   IF NPT(K)  EQ 0, GO TO 10
    NOW CREATE TIEUPS ON OTHER RUNWAYS, IF SUCH INTERFERENCE EXISTS.
    DO TO 8,FOR J=(1)(NPT(K))
    CREATE TIEUP
    KK IS THE RUNWAY AFFECTED.
    LET KK=RPT(K,J)
    IT IS THE TYPE OF TIEUP TO BE CREATED.
    TIEUP TYPES 1, 2, AND 6 APPLY TO LANDINGS.
    LET IT=TPT(K,J)
    GO TO (3,4,6,6,6,5),IT
    CREATE A TIEUP TO MAINTAIN INTER-ARRIVAL SEPARATION.
  3 FIND FIRST, FOR EACH FLT IN Q(KK,2), IF NONE, GO TO 6
    STORE FLT IN F
    LET MM=TYPE(F)
    LET S=VLAND(MM)
    IF S GE V, GO TO 325
    LET TMIN(TIEUP)=T
    LET TMAX(TIEUP)=T+SEPLL/S
    LET JJ=1
    GO TO 7
325 LET TMIN(TIEUP)=TD
    LET TMAX(TIEUP)=TD+SEPLL/S
```

```
      LET JJ=2
      GO TO 7
C     CREATE A TIEUP TO MAINTAIN DEP/ARR SEPARATION.
    4 LET TMAX(TIEUP)=TD
      IF DME(M) GT 0, GO TO 425
      LET TMIN(TIEUP)=T+(DOM(K)-DAFIX)/V
      GO TO 450
  425 FIND FIRST , FOR EACH FLT IN Q(KK,1), IF NONE GO TO 6
      STORE FLT IN F
      LET MM=TYPE(F)
      LET S=VTOFF(MM)
      LET TMIN(TIEUP)=T+(DOM(K)-(SEPTL+.5*V**2/S*ROTT(MM)))/V
  450 LET JJ=3
      GO TO 7
      TIE UP THE END OF AN INTERSECTING RUNWAY TO TAKEOFFS AND LANDINGS
      UNTIL AFTER THE LANDING  PASSES THE INTERSECTION.
    5 LET TMIN (TIEUP) =TD
      LET A=(VTAXI-V)/ROTL(M)
      LET TEM=.5*A*ROTL(M)**2+V*ROTL(M)
      IF THE LANDING WILL NOT REACH THE INTERSECTION,
      TIE UP UNTIL THE LANDING TURNS OFF.
      IF TEM LE DINT(K,KK), GO TO 51
      LET B=V**2+2.*A*DINT(K,KK)
      LET TUP=TD+(-V+SQRT(B))/A
      GO TO 52
51    LET TUP=TD+ROTL(M)
52    LET TMAX(TIEUP)=TUP
      FILE TIEUP IN THTI(KK)
      CREATE FTIUP
      STORE KK IN RWAY(FTIUP)
      STORE 2 IN PT(FTIUP)
      CAUSE FTIUP AT TMAX(TIEUP)
      CREATE TIEUP
      LET TMIN(TIEUP)=TD
      LET TMAX(TIEUP)=TUP
      LET JJ=3
      GO TO 7
    6 DESTROY TIEUP
      GO TO 8
    7 GO TO(701,702,703),JJ
  701 FILE TIEUP IN OMTI(KK)
      GO TO 705
  702 FILE TIEUP IN THTI(KK)
      GO TO 705
  703 FILE TIEUP IN ERTI(KK)
  705 CREATE FTIUP
      STORE KK IN RWAY(FTIUP)
      STORE JJ IN PT(FTIUP)
      CAUSE FTIUP AT TMAX(TIEUP)
8     LOOP
10    CREATE NXTOP
      STORE K IN RWAY(NXTOP)
      LET NEXT(K)=0
      CAUSE NXTOP AT T
C     DTEM IS THE DELAY ENCOUNTERED BY THIS LANDING.
      LET DTEM=(T-TIN(FL))*60.
      IF TD LS TBEG,GO TO 50
      CREATE PRINT
```

```
C       STORE DATA TO BE RECORDED AT TOUCHDOWN.
        STORE DTEM IN DLAY(PRINT)
        STORE K IN RWAY(PRINT)
        STORE 2 IN OP(PRINT)
        CAUSE PRINT AT TD
 50     DESTROY FLT CALLED FL
        LET LAST(K)=2
        DESTROY LAND
        RETURN
        END LAND
```

```
      ENDOGENOUS EVENT TOFF
      TOFF CREATES THE TIEUPS RESULTING FROM AN AIRCRAFT TAKING OFF.
      STORE RWAY(TOFF) IN K
      DESTROY TOFF
      IF Q(K,1) IS EMPTY, GO TO 16
      FIND TAKEOFF TO BE SCHEDULED AND STORE ITS ATTRIBUTES.
      REMOVE FIRST FLT FROM Q(K,1)
      STORE FLT IN FL
      LET M=TYPE(FL)
      LET V=VTOFF(M)
      LET T=FREER(K,1,FL)
      LET TD=T+TDMIN(K)
      TIE UP THE RUNWAY TO TAKEOFFS AND LANDINGS FOR DURATION OF THE
      RUNWAY OCCUPANCY TIME.
      CREATE TIEUP
      LET TMIN(TIEUP)=TD
      LET TMAX(TIEUP)=TD+ROTT(M)
      FILE TIEUP IN THTI(K)
      CREATE FTIUP
      STORE K  IN RWAY(FTIUP)
      STORE 2 IN PT(FTIUP)
      CAUSE FTIUP AT TMAX(TIEUP)
      CREATE TIEUP
      LET TMIN(TIEUP)=TD
      LET TMAX(TIEUP)=TD+ROTT(M)
      FILE TIEUP IN ERTI(K)
      CREATE FTIUP
      STORE K  IN RWAY(FTIUP)
      STORE 3 IN PT(FTIUP)
      CAUSE FTIUP AT TMAX(TIEUP)
      FIND FIRST, FOR EACH FLT IN Q(K,1), IF NONE, GO TO 2
      STORE FLT IN F
      LET L=DFIX(F)
      LET LL=DFIX(FL)
C     TIE UP THE END OF THE RUNWAY TO THE NEXT TAKEOFF LONG ENOUGH
C     TO MAINTAIN INTER-DEPARTURE SEPARATION. THIS DEPENDS ON WHETHER
      OR NOT THE TWO TAKEOFFS DIVERGE.
      CREATE TIEUP
      LET TMIN(TIEUP)=TD
      LET TMAX(TIEUP)=TD+SEPTT(LL,L)
      FILE TIEUP IN ERTI(K)
      CREATE FTIUP
      STORE K  IN RWAY(FTIUP)
      STORE 3 IN PT(FTIUP)
      CAUSE FTIUP AT TMAX(TIEUP)
  2   IF OPER(K) LT 3, GO TO 5
      FIND FIRST, FOR EACH FLT IN Q(K,2), IF NONE, GO TO 5
      STORE FLT IN F
      LET S=VLAND(TYPE(F))
C     CREATE A TIEUP TO MAINTAIN DEP/ARR SEPARATION.
      CREATE TIEUP
      LET TMIN(TIEUP)=TD
      IF DME(TYPE(F)) EQ 0, GO TO 3
      LET TMAX(TIEUP)=TD+(SEPTL+.5*S**2/V*ROTT(M))/S
      GO TO 4
3     LET TMAX(TIEUP)=TD+DAFIX/S
  4   FILE TIEUP IN THTI(K)
```

```
  CREATE FTIUP
  STORE K IN RWAY(FTIUP)
  STORE 2 IN PT(FTIUP)
  CAUSE FTIUP AT TMAX(TIEUP)
  IF NPT(K) EQ 0, GO TO 15
  CREATE TIEUPS OF OTHER RUNWAYS AS REQUIRED.
  DO TO 14,FOR J=(1)(NPT(K))
  CREATE TIEUP
  KK - RUNWAY AFFECTED
  LET KK=RPT(K,J)
  IT - TYPE OF TIEUP
  ONLY TYPES 3, 4, 5, AND 6 APPLY TO TAKEOFFS.
  LET IT=TPT(K,J)
  GO TO(12,12,6,8,8,10),IT
  CREATE A TIEUP TO MAINTAIN PROPER DEP/ARR SEPARATION.
6 FIND FIRST,FOR EACH FLT IN Q(KK,2), IF NONE,GO TO 12
  STORE FLT IN F
  LET MM=TYPE(F)
  LET S=VLAND(MM)
  LET TMIN(TIEUP)=TD
  LET JJ=2
  IF DME(TYPE(F)) EQ 0, GO TO 7
  LET TMAX(TIEUP)=TD+(SEPTL+.5*S**2/V*ROTT(M))/S
  GO TO 13
7 LET TMAX(TIEUP)=TD + DAFIX/S
  GO TO 13
8 FIND FIRST,FOR EACH FLT IN Q(KK,1), IF NONE, GO TO 12
  STORE FLT IN F
  LET L=DFIX(F)
  LET LL=DFIX(FL)
  CREATE A TIEUP TO MAINTAIN PROPER INTER-DEPARTURE SEPARATION.
  LET TMIN(TIEUP)=TD
  IF THE RUNWAYS ARE DEPENDENT, USE THE SAME SEPARATION AS FOR ONE
  RUNWAY, I.E. THOSE IN THE SEPTT ARRAY.
  LET TMAX(TIEUP)=TD+SEPTT(L,LL)
  IF THE RUNWAYS ALLOW SIMULTANEOUS DEPARTURES WHEN THEY DIVERGE,
  USE THE SEPARATIONS IN THE SEP2 ARRAY.
  IF IT EQ 5,LET TMAX(TIEUP)=TD+SEP2(L,LL)
  LET JJ=3
  GO TO 13
  TIE UP THE END OF AN INTERSECTING RUNWAY TO ALL OPERATIONS UNTIL
  THE TAKEOFF PASSES THE INTERSECTION.
  LET TMIN(TIEUP)=TD
  LET A =V/ROTT(M)
  LET TEM=.5*A*ROTT(M)**2+V*ROTT(M)
  IF THE TAKEOFF IS AIRBORNE BEFORE REACHING THE INTERSECTION,
  TIE UP ONLY UNTIL AIRBORNE.
  IF TEM LE DINT(K,KK), GO TO 51
  LET B=V**2+2.*A*DINT(K,KK)
  LET TUP=TD+(-V+SQRT(B))/A
  GO TO 52
  LET TUP=TD+ROTT(M)
  LET TMAX(TIEUP)=TUP
  FILE TIEUP IN THTI(KK)
  CREATE FTIUP
  STORE KK IN RWAY(FTIUP)
  STORE 2 IN PT(FTIUP)
  CAUSE FTIUP AT TMAX(TIEUP)
```

```
      CREATE TIEUP
      LET TMIN(TIEUP) =TD
      LET TMAX(TIEUP)=TUP
      LET JJ=3
      GO TO 13
   12 DESTROY TIEUP
      GO TO 14
   13 GO TO(14,131,132),JJ
  131 FILE TIEUP IN THTI(KK)
      GO TO 135
  132 FILE TIEUP IN ERTI(KK)
  135 CREATE FTIUP
      STORE KK IN RWAY(FTIUP)
      STORE JJ IN PT(FTIUP)
      CAUSE FTIUP AT TMAX(TIEUP)
   14 LOOP
 15    CREATE NXTOP
      STORE K IN RWAY(NXTOP)
      LET NEXT(K)=0
      CAUSE NXTOP AT T
C     DTEM - THE DELAY INCURRED BY THIS TAKEOFF
      LET DTEM=(T-TIN(FL))*60.
      IF TD LS TBEG,GO TO 50
      CREATE PRINT
C     STORE DATA TO BE RECORDED AT THE TIME THE TAKEOFF TURNS
C     ON TO THE RUNWAY.
      STORE DTEM IN DLAY(PRINT)
      STORE K IN RWAY(PRINT)
      STORE 1 IN OP(PRINT)
      CAUSE PRINT AT TD
 50   DESTROY FLT CALLED FL
      LET LAST(K)=1
      RETURN
 16    WRITE ON TAPE 6, TIME, K
      FORMAT (' AT TIME',D2.4,S2,'TAKEOFF QUEUE FOR RUNWAY',I3,S2,
     1  'IS EMPTY')
      STOP
      END TOFF
```

```
      ENDOGENOUS EVENT FTIUP
C     FTIUP REMOVES TIEUPS FROM THEIR SETS AND DESTROYS THEM WHEN
C     SIMULATED TIME PASSES THE END-LIMIT OF THE TIEUP.
      STORE RWAY(FTIUP) IN K
      STORE PT(FTIUP) IN J
      DESTROY FTIUP
      GO TO(10,20,30),J
   10 REMOVE FIRST TIEUP FROM OMTI(K)
      GO TO 40
   20 REMOVE FIRST TIEUP FROM THTI(K)
      GO TO 40
   30 REMOVE FIRST TIEUP FROM ERTI(K)
   40 DESTROY TIEUP
      RETURN
      END FTIUP


      ENDOGENOUS EVENT PRINT
C     PRINT  RECORDS DATA ON EACH FLIGHT AT THE TIME IT ACTUALLY
C     TOUCHES DOWN OR TURNS ON TO THE RUNWAY, AS THE CASE MAY BE.
      STORE RWAY(PRINT) IN K
      STORE DLAY(PRINT) IN D
      STORE OP(PRINT) IN I
      DESTROY PRINT
C     NTOFF AND NLAND ARE THE TOTAL NUMBER OF TAKEOFFS AND LANDINGS
C     DURING THIS HOUR.
C     DELT AND DELL ACCUMULATE TOTAL DELAY ON TAKEOFFS AND
C     LANDINGS BY HOUR.
      GO TO(10,20),I
   10 LET NTOFF(IHOUR)=NTOFF(IHOUR)+1
      LET DELT(IHOUR)=DELT(IHOUR)+D
      RETURN
   20 LET NLAND(IHOUR)=NLAND(IHOUR)+1
      LET DELL(IHOUR)=DELL(IHOUR)+D
      RETURN
      END PRINT


      ENDOGENOUS EVENT CHOUR
C     CHOUR CHANGES THE HOUR  AND BEGINS GENERATING FLIGHTS AT THE
C     RATE OF OPERATION FOR THE NEW HOUR.
      LET IHOUR=IHOUR+1
      IF IHOUR GT NH, LET IHOUR=1
      LET T=TIME+1.
      IF T LE TEND, CAUSE CHOUR AT T
      DO TO 1, FOR EACH O I
      STORE GENN(I) IN GEN
      CANCEL GEN
      CAUSE GEN AT TIME-LAMBD(I,IHOUR)*ALOG(1.-RANDM)
    1 LOOP
      RETURN
      END CHOUR
```

```
  ENDOGENOUS EVENT ENDS
  WRITE ON TAPE 6
  FORMAT ('1SUMMARY REPORT FOR THIS RUN'///S3,'HOUR',S10,
 1'OPERATIONS GENERATED',S10,'OPERATIONS PERFORMED',S10,
 2'TOTAL DELAY (MINUTES)',S8,'DELAY PER A/C (MINUTES)'/S15,
 3'LANDINGS  TAKEOFFS  TOTAL',S5,'LANDINGS  TAKEOFFS  TOTAL',
 4S5,'LANDINGS  TAKEOFFS  ALL',S7,'LANDINGS  TAKEOFFS  ALL'//)
  LET MARR=0
  LET MDEP=0
  LET MLAND=0
  LET MTOFF=0
  LET BDELT=0.
  LET BDELL=0.
  DO TO 1, FOR EACH H I
  LET NIN=NARR(I)+NDEP(I)
  LET MARR=MARR+NARR(I)
  LET MDEP=MDEP+NDEP(I)
  LET NOP=NLAND(I)+NTOFF(I)
  LET MLAND=MLAND+NLAND(I)
  LET MTOFF=MTOFF+NTOFF(I)
  LET TDEL=DELT(I)+DELL(I)
  LET BDELT=BDELT+DELT(I)
  LET BDELL=BDELL+DELL(I)
  IF NOP EQ 0, GO TO 3
  LET ADEL=NOP
  LET ADEL=TDEL/ADEL
  IF NLAND(I) GT 0, GO TO 4
  LET ADELL=0.
  GO TO 5
4 LET ADELL=NLAND(I)
  LET ADELL=DELL(I)/ADELL
5 IF NTOFF(I) GT 0, GO TO 6
  LET ADELT=0.
  GO TO 2
6 LET ADELT=NTOFF(I)
  LET ADELT=DELT(I)/ADELT
  GO TO 2
3 LET ADEL=0.
  LET ADELL=0.
  LET ADELT=0.
2 WRITE ON TAPE 6, I,NARR(I),NDEP(I),NIN,NLAND(I),NTOFF(I),NOP,
 1DELL(I),DELT(I),TDEL,ADELL,ADELT,ADEL
  FORMAT (S4,I2,S12,I2,S8,I2,S6,I3,S9,I2,S8,I2,S6,I3,S6,D5.1,S3
 1D5.1,S1,D5.1,S5,D5.1,S3,D5.1,S1,D5.1)
1 LOOP
  LET NIN=MARR+MDEP
  LET NOP=MLAND+MTOFF
  LET TDEL=BDELT+BDELL
  LET ADEL=NOP
  LET ADEL=TDEL/ADEL
  LET ADELL=MLAND
  LET ADELL=BDELL/ADELL
  LET ADELT=MTOFF
  LET ADELT=BDELT/ADELT
  WRITE ON TAPE 6, MARR,MDEP,NIN,MLAND,MTOFF,NOP,BDELL,BDELT,TDEL,
 1ADELL,ADELT,ADEL
  FORMAT ('0 TOTAL',S10,I3,S7,I3,S5,I4,S8,I3,S7,I3,S5,I4,S5,D6.1,
 1S2,D6.1,D6.1,S5,D5.1,S3,D5.1,S1,D5.1)
  STOP
  END ENDS
```

APPENDIX E

CONVERSION TO OTHER COMPUTERS

The model described in this report is currently operable on the National Bureau of Standards UNIVAC 1108 under the EXEC II operating system. This computer has 65,536 36-bit words of core storage of which about 53,000 are available under EXEC II. The model consists of two separate programs, the preprocessor and the simulation, which are executed in succession within the same run under EXEC II. The preprocessor output/simulation input tape is rewound between the execution of the two programs by a system utility routine. However, the REWIND instruction could be included in the preprocessor program if desired. For both programs card input is from logical unit 5 and printer output is on 6. The airport data file tape is on unit 7 and the preprocessor output/simulation input tape is on 8. The airport file unit may be altered by changing the value of the variable IN to the desired unit number. The preprocessor output/simulation input unit may be altered by changing the value of the variable M in the preprocessor program and putting the corresponding new value in columns 35 and 36 (and if there are no explicitly generated flights, in columns 41 and 42 also) on the system specification card required as input to the simulation program.

The preprocessor program is written in FORTRAN V (the UNIVAC augmented FORTRAN VI). Several features which have been used in the preprocessor program may not be available in FORTRAN IV compilers on other machines.

1. The PARAMETER statement is used to define the value of an integer constant which may then appear as a DIMENSION size limit. Such variables as the maximum number of runways (KRW), the maximum number of aircraft types (KTYP), the maximum number of departure paths (KFIX), the number of operation types (KO), and the maximum number of interferences for a runway (KTPT) are all defined in a PARAMETER statement. The compiler treats these variables as constants, rather than variables, but their values may be altered by changing only the PARAMETER statement, rather than every instance of the occurrence of the value. The use of the PARAMETER may be circumvented for compilers lacking this capability by using fixed DIMENSION limits and setting KRW, KTYP, KFIX, KO, and KTPT equal to the appropriate constants at at the beginning of the program.
2. The UNIVAC FORTRAN V permits an end of file condition to be detected on input through the use of the READ (unit, format, END = i), where program control transfers to statement number i when an end of file is read. This may be circumvented by using a particular signal sequence of characters and testing after the READ statement.

3. The FORTRAN V allows the use of quote marks enclosing a Hollerith field in a FORMAT statement and also in an arithmetic statement such as

JAY = 'J'.

These may be replaced by the nH form if the compiler does not have this feature.

4. Mixed-mode arithmetic is correctly performed, and thus there are no diagnostics for mixed-mode expressions. Therefore, although care has been taken to avoid such expressions some may have survived.

The simulation program is written in SIMSCRIPT 1.5. The user should check the directions for compiling SIMSCRIPT on his particular machine. On the UNIVAC 1108 SIMSCRIPT 1.5 compiles into SLEUTH, the assembly language, which is then assembled into machine code before the program is executed. SIMSCRIPT 1.5 is available on a variety of computers and is quite standard from one machine to the next since most of the compilers were constructed by the same company. Some things are machine dependent and should be checked when using other computers.

1. Some attributes are packed, two to a computer word. The packing allowed may depend on the computer word size for other machines.

2. The SIMSCRIPT compiler on the UNIVAC 1108 allows both LT and LS although the latter is standard.

3. The 1108 version of the compiler accepts Hollerith strings enclosed in quotes in addition to the standard nH form.

APPENDIX F

SEPARATION BETWEEN A LANDING AND A PRECEDING TAKEOFF

A landing aircraft which possesses DME must be separated by a distance S (currently 2 miles) from a preceding takeoff on the same runway. In this appendix, we will derive an expression for the distance d which must separate the two aircraft when the takeoff starts its roll, in order that the two aircraft remain S apart. We make two assumptions:

(1) The landing speed v_L is constant along to the final approach path.

(2) The takeoff acceleration a is constant along the runway and for a short while after liftoff.

The value a of the constant acceleration can be calculated from the known liftoff speed v_T and the takeoff's runway occupancy time R, as

$$a = v_T/R.$$

Under our two assumptions, the distance s_1 the takeoff has gone in t units of time is

$$s_1 = (1/2)\ a\ t^2 = 0.5\ v_T\ t^2/R.$$

The distance s_2 traveled by the landing in the same time is

$$s_2 = v_L\ t.$$

Since they start out a distance d apart[1] at any time t they are separated by a distance D where

$$D = 0.5\ v_T\ t^2/R - v_L t + d.$$

D must be always greater than or equal to S. Differentiating, we find that D is minimum when $t = R\ v_L/v_T$. Since this is the time that the two aircraft will be closest, we set D = S at that time and solve for d, yielding

---

[1]We assume here and elsewhere in this report that the runway threshold and the end of the runway are geographically the same point. If this were not true an appropriate distance would have to be added or subtracted from d.

$$d = S + .5\, v_L^2\, R/v_T.$$

Therefore, if the landing is this distance d from the end of the runway as the takeoff starts its roll, the two aircraft will never be closer than the required separation S.

APPENDIX G

TIEUP TIME RESULTING FROM INTERSECTING RUNWAYS

In order to compute the time at which an intersection point will become free, it is necessary to know when a landing or a takeoff passes the intersection. In this appendix we will derive an expression for the time t it takes an aircraft to travel from the end of the runway to a point a distance D down the runway, assuming a constant acceleration a along the runway. The value of this acceleration may be calculated from the initial speed v_o, the final speed v_1, and the runway occupancy time R. At any time t the speed v of the aircraft is

$$v = a\,t + v_o.$$

However, at t = R the speed is v_1 so

$$a = (v_1 - v_o)/R.$$

At time $t \leq R$, the distance d traveled by the aircraft is

$$d = 0.5\,(v_1 - v_o)\,t^2/R + v_o\,t.$$

Setting D = d and solving for t yields

$$t = \frac{-v_o \pm \sqrt{v_o^2 + 2D\,(v_1 - v_o)/R}}{(v_1 - v_o)/R}.$$

To ascertain which sign applies here, we note that for D > 0 we must have t > 0, so we choose the positive sign. Therefore since $t \leq R$ we have

$$t = \min\left[\frac{-v_o + \sqrt{v_o^2 + 2D(v_1 - v_o)/R}}{(v_1 - v_o)/R}, R\right]$$

To interpret this for a landing, v_o is the (constant) landing speed and v_1 is the speed at which the aircraft turns off the runway. This yields a negative acceleration, i.e. a deceleration. For takeoffs v_o is zero since the aircraft starts off at rest, and v_1 is the liftoff speed of the aircraft. In this case t simplifies to

$$t = \min \left[\sqrt{2DR/v_1},\ R \right].$$

APPENDIX H

(Mock) Instructions on How to Use the NBS Airport Capacity/Delay Simulation Model

(Note: The Model is referred to here as "DELCAP".)

General Information

DELCAP is physically located within a CDC 6600 computer in the CDC office building located at 11428 Rockville Pike, Rockville, Maryland. In order to access it you use the time-share terminal located in Room 803 in building FOB 10A. The instructions for turning on the terminal equipment in Room 803 and connecting to the Rockville CDC 6600 are contained in a black 3-ring binder, entitled "Computer Connection Instructions," attached by a string to the wall of Room 803. This terminal equipment is comprised of a card-reader, a printer and a Cathode Ray Tube (CRT) display. This equipment and DELCAP are available for use at any time between 7 a.m. and 8 p.m. Monday through Saturday.

What DELCAP is:

DELCAP is a computer model which can be made to represent any of a broad range of actual or hypothesized airport configurations (and traffic mixes). Once you have specified the airport and situation for analysis - which may be done in either of two ways, in the constructive mode or in the on-file mode - you may then ask DELCAP for any or all of the following information:

(a) The hourly throughput of the airport.

(b) The average delay per aircraft using the airport.

(c) The average number of aircraft waiting to take off or land.

These quantities are gathered separately for each hour and over a whole day, and may also be broken down by aircraft type.

How to Provide DELCAP with the Information it Needs:

This may be done by using DELCAP in either its constructive mode or in its on-file mode. Generally, the constructive mode is to be used when analysis is required of a proposed new airport design or runway changes to an existing airport. Any experimental runway configuration may be analyzed in this mode. The on-file mode, by contrast, is to be used when an already-existing airport is to be analyzed. The runway configurations and traffic forecasts for these airports will be permanently stored within DELCAP. For example, you can obtain the capacity and delay conditions at WNA if all traffic except medium-size jets were prohibited, or the throughput and delay conditions at O'Hare in 1985 when subjected to the most recent traffic forecasts for that year.

The Constructive Mode:

To use DELCAP in this mode you would type in the following:

(1) The description of the airport

(1-1) For each runway, its heading and the distance to the outer marker. e.g. runway 1, heading 170 degrees, distance to outer marker 8 miles.

(1-2) For each pair of intersecting runways, the distance from the end of first runway to the intersection with the second, and the distance from the end of the second runway to the intersection with the first, e.g. 3, 1, 5050., 3020. Runways 3 and 1 intersect at a point 5050 feet from the end of runway 3 and 3020 feet from the end of runway 1.

(1-3) For each pair of parallels whether or not they have independent approaches and whether takeoffs on one are independent of landings on the other.

(2) The description of the traffic using the airport.

(2-1) Aircraft mix by type of aircraft e.g. 9% large jets, 37% medium jets and large propeller, 26% medium propeller and small jets, 18% light twin engine, 10% single engine.

(2-2) Number of arrivals and number of departures per hour for each hour of the day

(2-3) Any explicit arrivals or departures to be added to the Poisson generated flights resulting from (2-2).

(2-4) Distribution of use of departure paths for each runway.

(3) Description of certain airport parameters.

(3-1) Mix of runway use by aircraft type for landings and separately for departures e.g. 40% of large jets land on runway 1 and 60% on runway 2. 20% of large jets take off from runway 1 and 80% from runway 2.

(3-2) Minimum spacing rules

a. Interlanding e.g. 3 miles for IFR approaches

b. Landing following a takeoff e.g. 2 miles

c. Between takeoffs - one for those using the same departure path, and another for those using different departure paths for the same runway e.g. 1 minute between departures using different paths and 3 miles between departures using the same path. With noise abatement procedures in force the 1 minute might be increased to 2 minutes.

(3-3) Aircraft type characteristics.

e.g. final approach speed, typical runway occupancy time for each type of aircraft.

The On-File Mode:

To use DELCAP in this mode you must type in the following:

(1) Airport designator e.g. (JFK)

(2) Any changes to filed items (2) or (3) of the Constructive Mode Parameters.