FTL REPORT R80-1

A LINEAR PROGRAMMING SOLUTION TO THE GATE ASSIGNMENT PROBLEM AT AIRPORT TERMINALS

Rami Sabet Mangoubi

June 1980

Flight Transportation Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

ABSTRACT

This research solves the flight-to-gate assignment problem at airports in such a way as to minimize, or at least reduce, walking distances for passengers inside terminals. Two solution methods are suggested. The first is a heuristic algorithm which assigns the "most crowded" aircraft (i.e., most on-board passengers) to the best gate, while the second consists of formulating the problem as a linear program.

A flight schedule of one day at Terminal No. 2 of Toronto International Airport is used to test and compare the two methods. The algorithm offers an assignment solution with a 27% reduction in the expected walking distance when compared to the original assignment at the airport. The linear program's assignment gives a 32% reduction. The heuristic algorithm is, therefore, only 5% suboptimal for the sample problem. In addition, its associated computational expenses, less than $10 per run, are by far cheaper than those of the linear program with expenses as high as $400 per run. Such excellent, or even acceptable, performance by the algorithm cannot be guaranteed for all problems. A strategy which helps decide when to use which approach is therefore suggested.

CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 The Problem

The airport terminal is the area where passenger servicing and processing take place. In planning for that area, one of the major considerations in the airport planner's mind should be the quality of service offered to passengers. The enormous growth in air transportation, which occurred during the last two decades, necessitated the enlargement of existing airport terminals as well as the founding of new ones, in order to satisfy growing demands. Careful terminal planning, as well as efficient management, are, therefore, of crucial importance if the passenger is to receive a quality service.

Though hard to measure, an important criterion for the quality of service is the distance the passenger is required to walk inside the terminal before reaching either his aircraft or the baggage claim area. In planning new installations, therefore, designers make considerable efforts to minimize the traveller's walking distances. Trying to address the problem, planners introduced new concepts in terminal building architecture, each one of them offering its own special advantage. For instance, in the satellite pier concept, gates are grouped together in satellites, thus facilitating the movement for transfer passengers if the connecting flights are assigned to gates in the same satellite group. The satellite concept is a modified version of the finger pier concept and offers the advantage of more space for the easy assembly of passengers.

Both satellite and finger pier designs are centralized processing concepts. Centralized processing permits a large passenger processing capacity without excessive land-area usage. In the gate arrival concept, however, each gate has its own processing facility, thus shortening the waiting time for passengers and reducing the level of congestion in any one area. In the gate arrival concept, there are gates in a central position and thus, more accessible from public transportation than other gates which are located further. The central gates can be used for scheduled flights, or any flights with higher priority (such as those normally boarded by elderly or frequently travelling businessmen), while the more distant gates can be used for charters, V.I.P.'s and other flights.

While the choice of the proper terminal design is important in easing the burden of long walking distances on air passengers, efficient operational procedures are also essential to improving the situation. Such procedures become even more crucial when present installations are either undergoing expansion in order to meet the anticipated growth in air travel, or are to serve as permanent buildings with no anticipated plans for modern replacements. One such procedure, and the one with which this research is concerned, is the assignment of scheduled flights to airport gates, with the objective of a reduced walking distance for the passenger in mind.

Traditionally, aircraft are assigned to gate positions to satisfy various operating requirements such as available servicing equipment, ramp crew scheduling, etc. Rarely is any consideration given to the number of passengers on the plane and how far they have to walk, whether to the baggage claim area from the aircraft, from the check-in counter to the gate, or from one gate to the other. The purpose of this research, therefore, is to suggest solutions to the gate assignment problem from the point of view of the passenger's walking distance.

1.2 A Brief Review of Past Research

Passenger terminal servicing and processing have been the subject of much research, and numerous terminal designs as well as handling approaches have been reported in the literature. The amount of research concerned with flight assignment to gates and to passenger walking distances is, however, limited.

J. P. Braaksma [1977] demonstrates that significant savings in walking distances can be had through appropriate gate allocations. He shows that the walking distance for users of Toronto Terminal No. 2 at Toronto International

Airport was reduced from 923 feet per passenger in 1973 to 744 feet in 1974 and 800 feet in 1975. This improvement is a direct result of a change in gate assignment policy by Air Canada, the terminal's sole user. Table 1.1 contains a small statistical summary of Braaksma's results. It is shown, for instance, that the median walking distance in 1973 was 890 feet per passenger while, in 1974 and 1975, the median was 744 feet and 800 feet respectively. Other percentiles are also contained in the table.

In another effort to address the same problem, J. Bustinduy [1977] suggests several gate assignment algorithms for implementation at major airports. Mangoubi [1978] tested these algorithms and found that one particular algorithm, that which assigns the best gate to the "crowdest" (i.e. most passengers on-board) aircraft performs better than the other algorithms suggested, when tested at Toronto Terminal No. 2. This algorithm, which Bustinduy calls "Crowdest-Come-Best-Serve", performed even better than another algorithm which the same author calls "optimal"! Nevertheless, the "Crowdest-Come-Best-Serve" algorithm still does not give an optimal solution to the problem, i.e., it does not give a minimum average walking distance per passenger.

	1973	1974	1975
85th Percentile	1,300	1,100	1,165
Mean Distance	923	744	800
50th Percentile	890	660	765
15th Percentile	480	380	430

Table 1.1 Various Statistics on Passengers' Walking Distance at Toronto Terminal No. 2 (Source: Braaksma [1977])

1.3 Outline of Research and Contributions

The present work aims at finding an optimal solution to the flight-to-gate assignment problem at airport terminals. The objective is a minimum average walking distance per passenger. Passengers connecting to other flights, as well as passengers originating or terminating their itinerary, are considered. Since, as mentioned in the last section, the "Crowdest-Come-Best-Serve" heuristic algorithm does not suggest an assignment with an optimal walking distance, a mathematical programming approach is introduced to solve the problem. The results from the mathematical program are compared against those of this algorithm. Finally, the computational costs for both the algorithm and the mathematical program are also compared.

Chapter 2 of this research discusses the "Crowdest-Come-Best-Serve" algorithm. Section 2.1 states and describes the algorithm and also briefly discusses the other algorithms which Bustinduy [1977] suggests. Section 2.2 contains a proof showing that the "Crowdest-Come-Best-Serve" algorithm does not necessarily offer an optimal assignment; and section 2.3 describes the input data necessary for the computer implementation of the algorithm, as well as the various assumptions taken.

Chapter 3 introduces the linear programming formulation of the problem. The model is described in

Section 3.1. In Section 3.2, a hypothetical problem is solved which, because of its small size, helps the reader visualize the shape of the linear program's constraint matrix. Section 3.3 discusses the computer implementation of the linear program. The section briefly introduces SESAME, the software optimization procedure used as well as the model generating program which builds, out of the necessary data input, the objective function and the constraint matrix. For purposes of comparison, the data assumptions used in the LP are exactly the same as those for the heuristic algorithm.

Chapter 4 presents and compares results of the two solution methods for Terminal No. 2 at Toronto International Airport*. In Section 4.1, some statistical analysis and comparisons are shown. Section 4.1 also briefly discusses the postprocessor program written to present the output information. A comparison of the costs of the two solutions is given in Section 4.2. Advice on the use of the LP versus the heuristic methods is also presented. Finally, conclusions and suggestions for further research appear in Chapter 5.

*The Data for this airport was made available to the M.I.T. Flight Transportation Laboratory by J. P.Braaksma, Assistant Professor in the Department of Civil Engineering at Carleton University, Ontario, Canada.

2. THE CROWDEST-COME-BEST-SERVE ALGORITHM

Bustinduy [1977] suggested several heuristic algorithms which assign flights to gates in such a way as to reduce passenger walking distances. One of these algorithms, the "Crowdest-Come-Best-Serve", performed better than any of the others when tested by Mangoubi [1978] on one day of scheduled flights at Toronto Terminal No. 2.

2.1 Description of the Heuristic Algorithms

The "Crowdest-Come-Best-Serve" algorithm assigns the best available gate, i.e., the gate with the shortest average walking distance per passenger, to the aircraft with the largest number of on-board passengers. For each scheduled flight, free gates are stored in a set G. Set S, a subset of set G, contains only those gates in G which can serve the flight category and its aircraft type. In the test case used, however, no distinction is made between the two sets, S and G. In other words, at Toronto Terminal No. 2, any free gate can serve any flight. The steps of this algorithm are as follows:

Step 1. Number the gates in a serial order and state them in a set G.

Step 2. Consider the "crowdest" arriving aircraft.

Step 3. Create a set S in order to store all gates which can serve that flight's aircraft.

Step 4. Try the first gate in set G.

Step 5. If set G is exhausted (there are no gates left), go to Step 8, else continue.

Step 6. If the gate can serve the flight's type of aircraft, store it in S and go to Step 7 else try the next gate and go to Step 5.

Step 7. Next to the gate number, store the average passenger's walking distance for the flight. Check next gate and go to Step 5.

Step 8. In set S, choose the gate with the minimum associated average walking distance. Assign it to the flight.

Step 9. Clear sets S and G.

Step 10. consider the next arriving flight. If all flights are exhausted, go to Step 13, else continue to Step 11.

Step 11. Check to see which gates are free at the flight's arrival time. Store these gates in set G after numbering them (in any order).

Step 12. Go to Step 4.

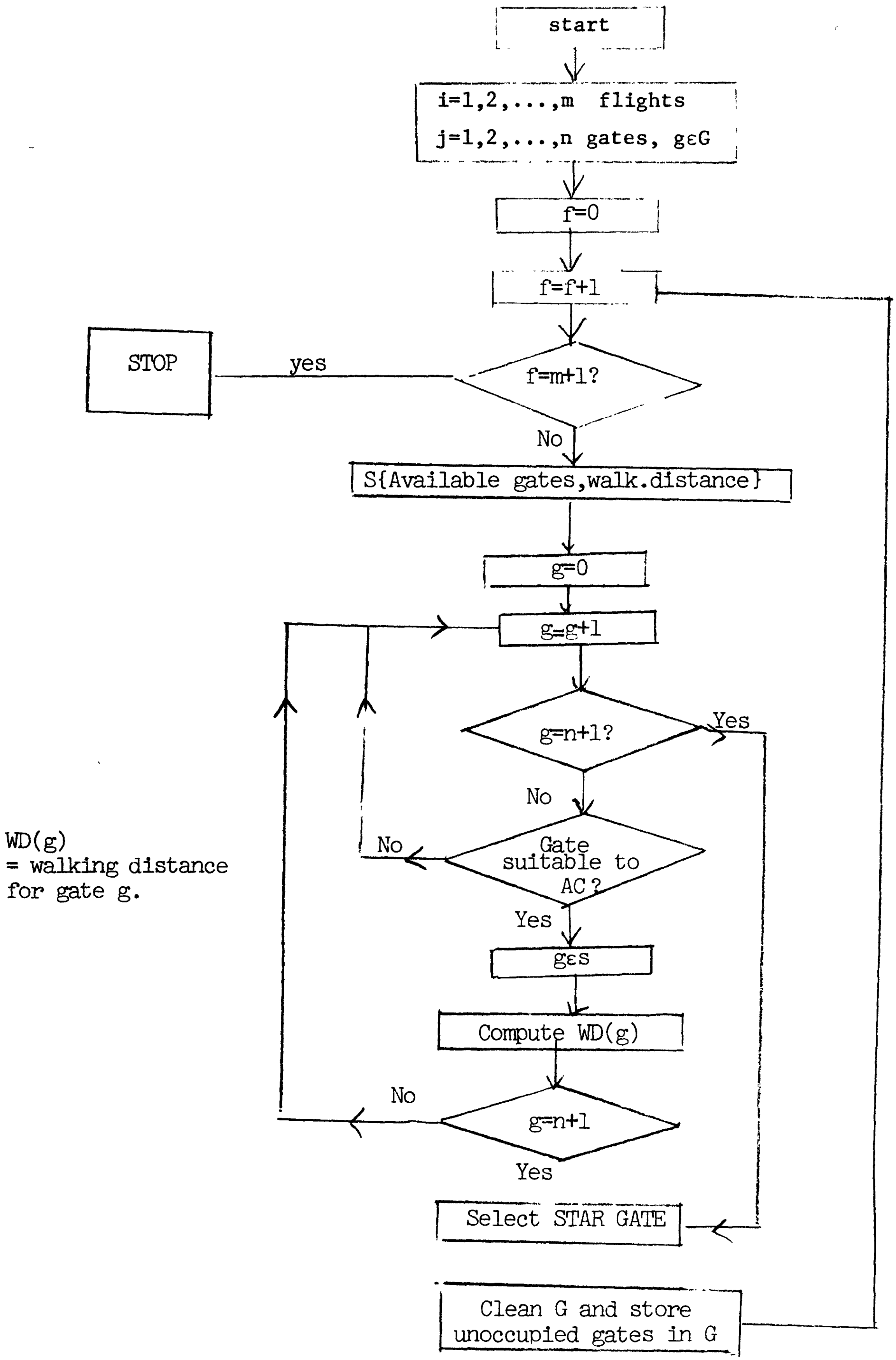
Step 13. Stop.

Figure 2.1 shows a flow chart description of this algorithm.

Another algorithm suggested by Bustinduy is the "First-Come-First-Serve" algorithm. Here, the first scheduled flight, instead of the "crowdest", is assigned to the best available gate. One can conclude a priori, that since the only priority consideration for the "First-Come-Best-Serve" algorithm is the scheduled time of arrival of a flight, that it can never suggest an assignment with a smaller walking distance than that of the "Crowdest-Come-Best-Serve".

Bustinduy suggests a third algorithm which looks ahead at all future scheduled flights before giving a final assignement to the next arriving flight. Briefly, the algorithm works as follows. It assigns the first scheduled

Figure 2.1 Flow Chart for the "Crowdest-Come-Best-Serve" Algorithm



flight to a gate. Given this assignment, the algorithm looks ahead and assigns the remaining flights to the best available gates on a first-come-first-serve basis. The total distance walked by all passengers is tallied. The first scheduled flight is then assigned to another gate and the walking distance of all passengers is once again tallied. All available gates which can serve that flight are in turn assigned to it in that manner. When all gates are exhausted, the gate assignment yielding the lowest average walking distance is given permanently to that flight. With the next scheduled flight, the whole process repeats itself. The algorithm stops when all scheduled flights are permanently assigned to a gate.

Mangoubi [1978] tested the three algorithms. In the test, all scheduled flights from one representative day of Terminal No. 2 at Toronto International Airport were used. The results of the test indicated that, of all the three algorithms, the assignment given by the "Crowdest-Come-Best Serve" algorithm yields the highest savings in average walking distance per passenger. This saving amounts, on the average, to about 27% of the walking distance resulting from the original assignment given to the flights by Air Canada.

Nevertheless, the "Crowdest-Come-Best-Serve" algorithm is not optimal, as will be shown in the following section. The results of the "Crowdest-Come-Best-Serve" algorithm, however, will be compared in Chapter 5 with those of the linear program introduced in Chapter 4.

2.2 Proof of the Algorithm's Suboptimality

This section contains a proof by counter example that the "Crowdest-Come-Best-Serve" algorithm does not necessarily provide an optimal gate assignment policy with respect to the average walking distance per passenger; hence, the motivation for the linear programming model introduced in the next chapter.

Consider, for instance, an airport schedule as follows: A Boeing 747 landing at 10:00 o'clock with 200 passengers on board and planning to take off three hours later at 13:00 o'clock, with the same number of passengers. Within these three hours, three Boeing 727 aircraft are also scheduled to be on the ground, but in such a way as not to conflict with each other. (For instance, the first B727 would arrive at 10:00 A.M. and depart at 10:40, the second would arrive at 10:45 A.M. and depart at 11:30 A.M., and the third would arrive at 12:00 and leave anytime.) Assume also that each of these B727's lands and takes off with 120 passengers on board.

The short time table for this hypothetical airport is shown in Table 2.1, along with the total number of passengers each plane serves. Assume that two gates exist at the airport, Gate A and Gate B, with walking distances shown in Table 2.2.

Flight	AC	Arrival	Departure	Pax
1	B727	10:00	10:40	240
2	B747	10:00	13:00	400
3	B727	10:45	11:30	240
4	B727	12:00	13:20	240

Table 2.1 Scheduled Flights Information for Example Given in Section 2.2

Gates	Walking Distance (ft)
A	650
B	800

Table 2.2 Average Walking Distances for Gates A and B

If a "Crowdest-Come-Best-Serve" policy is adopted, the Boeing 747 would be assigned to Gate A, since the Jumbo is the single largest scheduled aircraft and Gate A offers the shortest average walking distance in the airport. All of the Boeing 727's are thus assigned to Gate B because each of them, separately, conflicts with the Jumbo. One can see that such an assignment policy leads to a smaller number of B747 travellers (400) walking a shorter distance than the larger total of 720 passengers from the three Boeing 727's. Table 2.3 lists both the optimal assignment and the "Crowdest-Come-Best-Serve" assignment, along with the corresponding walking distances. Table 2.4 indicates that the shortest average walking distance per passenger (597 feet) does not result in the "Crowdest-Come-Best-Serve" algorithm, which gives 633 feet per passenger as an average walking distance.

Two conclusions can be drawn from this example. First, that a drawback of the algorithm lies in the fact that though the crowdest aircraft is offered the best gate, the policy takes no account of the length of time the aircraft is occupying the gate, and thus preventing other aircraft from utilizing it. Second, the degree of the algorithm's suboptimality needs not be of any significance (In this example, a difference of only 36 feet per passenger). How far from optimal the algorithm is, depends, of course, on

Flight	AC	PAX	Algorithm's Gate	Assignment Walking Distance	Optimal Gate	Assignment Walking Distance
1	B727	240	B	800	A	650
2	B747	400	A	650	B	800
3	B727	240	B	800	A	650
4	B727	240	B	800	A	650

Table 2.3 Gates and Walking Distances for Both the "Crowdest-Come-Best-Serve" and the Optimal Assignment Policies for the Example Problem

Assignment Policy	Average Walking Distance per Passenger
Crowdest-Come-Best-Serve	633 feet
Optimal	597 feet

(Total Number of passengers: 1,320)

Table 2.4 Average Walking Distances for all Passengers for the Two Assignment Policies the structure of the airport and the nature of its flights' schedule. For these reasons, the results of the algorithm will be compared in Chapter 5 against those of the linear program for Toronto Terminal No. 2.

The purpose of the above example is simply to demonstrate a drawback of the algorithm. In the actual test case, passengers can be of three types: arriving, departing or connecting. In addition, flights can be domestic, transborder, (U.S.) or international. A description of all the information necessary for the implementation of the algorithm on the computer is found in the report by Mangoubi [1978]. It is repeated in the next section for the sake of completion. The data are exactly identical to those used to test the linear programming formulation of the problem, though the input format is different.

2.3 Data Used to Solve the Problem

In order to test the "Crowdest-Come-Best-Serve" algorithm on the computer, a program which simulates the operational conditions of the algorithm was written. Each flight's characteristics and the terminal's layout constitute the information required to implement the algorithm (as well as the mathematical program to be described in the next chapter).

2.3.1 Flight and Passenger Information

As mentioned earlier, Toronto Terminal No. 2 at Toronto International Airport was selected for testing the algorithm and the mathematical program. A weekday from the summer of 1975 was selected and the flight's number, aircraft type, arrival and departure times, as well as the flight category and the gate actually assigned were tabulated. The flight's category consists of a number indicating whether the flight is domestic, 0, transborder (U.S.), 1, or international, 2, . The information described in this subsection. and the next one appears at the end of Appendix A (following the computer program which implements the heurestic algorithm).

A constant load factor of 65 percent was assumed for all aircraft using Terminal No. 2. Table 2.5 lists the various aircraft using the terminal, their capacity and their assumed seat occupation.

A constant load factor implies an equal number of arriving and departing passengers. The number of connecting or transfer passengers, given in Braaksma [1977], was estimated at about 30% of arriving passengers at Toronto. For example, flight number 136136, with a Boeing 747, lands with 248 passengers on board and takes off with an equal number of departing passengers (in addition to those transferring to it from other flights).

AIRCRAFT	CAPACITY	OCCUPATION
B747	382	248
L10	262	170
D8S	210	137
DC8	140	91
72S	135	88
727	135	88
D9S	110	72
DC9	90	59

Table 2.5

Summary of Aircraft Data for Toronto Terminal No. 2

Of the arriving passengers, it is assumed that 30% or 74 intend to board another flight at Toronto Terminal No. 2. These conncecting passengers, therefore, do not need to check in and go directly to their new departure gate.

One can thus conclude that 50% of all passengers are departing, 35% are arriving and 15% are connecting.

Finally, no restriction is assumed on the use of gates by any particular type of flight or aircraft (In any case, any computer implementation can be easily modified to accomodate such a constraint).

2.3.2 Walking Distance

Several approaches exist for measuring the walking distance travelled by airport passengers. Braaksma [1976] developed an elaborate method for collecting pedestrian traffic flow data in airport terminals. Turning away from traditional interview surveys which, in any case, yield fragmented bits of information, Braaksma's method consists of handing a card to each passenger as he enters the terminal; either at the gate for the unloading passenger (arriving or transfer) or at the door for the departing passenger. During his stay, the passenger keeps the card, which is time-stamped at various check points. As he leaves the terminal, the passenger delivers the card.

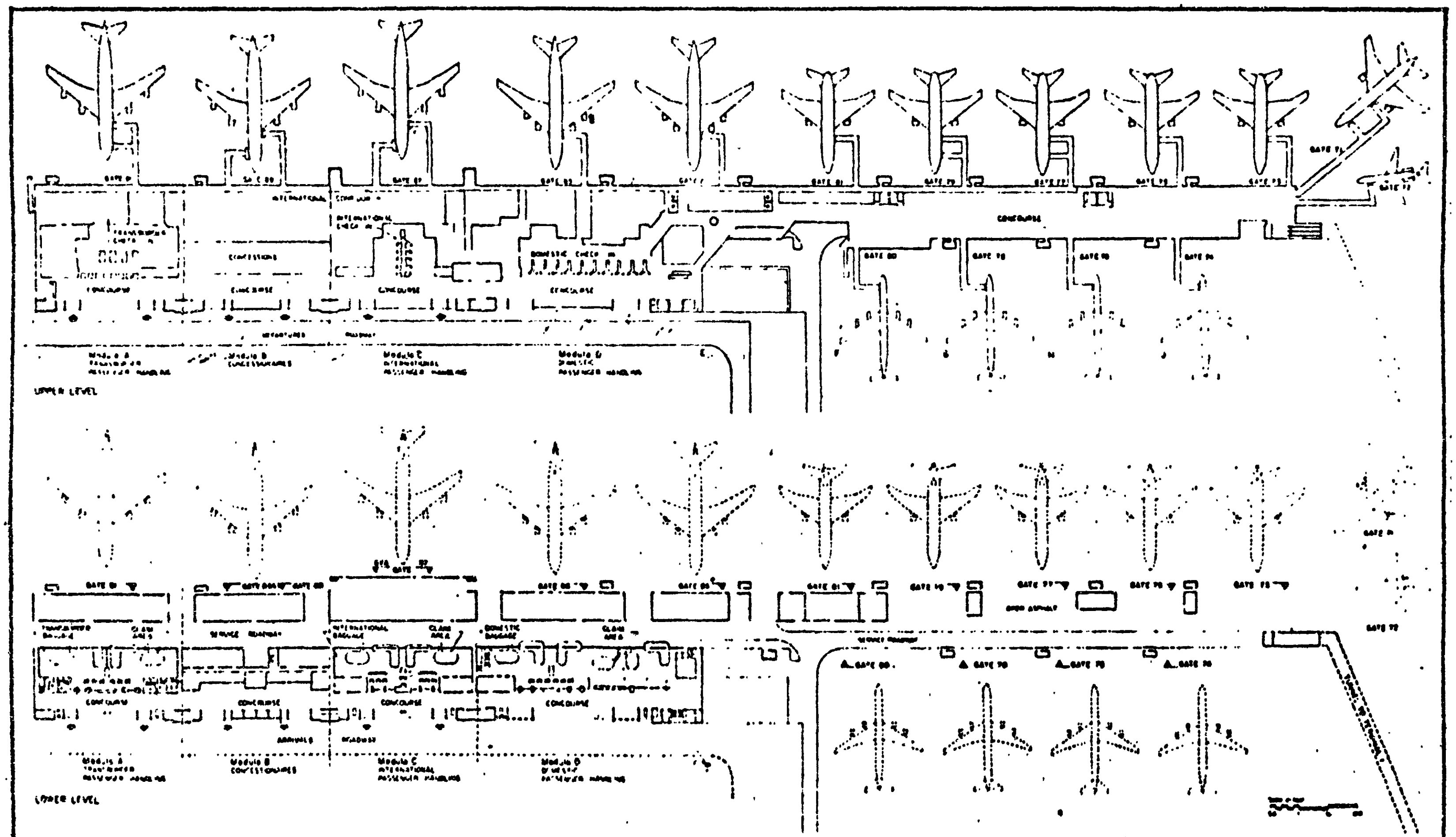
When tested for two days at Winnipeg International Airport, this technique proved successful as only 2% of the cards delivered were unaccounted for. It also produced data so comprehensive that they can yield volumes, flow rates, occupancies, queueing lengths, service times,... etc. Statistical distributions describing these various quantities can then be built and passengers' patterns can thus be better understood, enabling the airport to improve upon the service level offered to the passengers.

Though comprehensive in its nature, this method, called time-stamping, measures the actual distance traversed by the passenger, as opposed to the distance he has to walk, which this research is trying to minimize. A more direct approach was thus used and distances were measured with the help of the diagram in Figure 2.2 of Toronto Terminal No. 2, as well as accompanying explanation found in the other report by Braaksma [1977].

Table 2.6 lists the walking distance for non-transfer on non-connecting passengers in each flight category. The six columns in the table contain each gate's walking distance, for arriving and departing passengers, for each of the three categories of flights, domestic, transborder and international. In the case of departures, the distance represents the rectilinear walking distance between the check-in point and the gate, while in the case of arrivals, the distance is between the gate and the baggage claim point.

Figure 2.2

Plan of Terminal 2 at Toronto International Airport
(Departures Shown Above, Arrivals Shown Below)
(reproduced from [Braaksma,1977])



GATE	ARRIVALS DOMESTIC	ARRIVALS TRANSBORDER	ARRIVALS INTERNATIONAL	DEPARTURES DOMESTIC	DEPARTURES TRANSBORDER	DEPARTURES INTERNATIONAL
71	1287	2367	1727	1303	2261	1737
72	1269	2350	1710	1285	2244	1720
73	1285	2365	1725	1301	2259	1735
74	1106	2193	1553	1112	2087	1543
75	1102	2182	1542	1118	2076	1552
76	926	2013	1373	932	1907	1363
77	919	1929	1289	935	1823	1299
78	746	1833	1193	752	1727	1183
79	739	1749	1109	755	1643	1119
80	566	1670	1030	582	1564	1020
81	556	1566	926	572	1460	936
83	509	1343	703	349	1237	713
85	594	1068	428	434	962	438
87	855	807	347	695	701	177
89	1109	553	601	949	447	329
91	1363	299	855	1203	193	583
93	1662	598	1154	1502	492	882
95	1845	781	1337	1685	675	1065
97	510	418	828	350	312	668
99	957	418	828	797	312	568

Table 2.6

Walking Distances for Non-Transfer Passangers

[in feet]

The matrix in Table 2.7 displays the intergate distances. Again, connecting or transfer passengers are assumed to walk in a rectilinear manner. In addition to these distances, two probabilities are essential to compute the average walking distance for this third category of passengers. First, the transfer probability, as first mentioned in Section 2.3.1, is estimated at about 30% of arriving passengers at Toronto International Airport. Second, also essential is a distribution indicating the probability p_{kj} that a transfer passenger arriving at Gate k will depart from Gate j. Several approaches can be used to obtain this probability. The first is the "time-stamping" approach described earlier and suggested by Braaksma. The second approach consists of derived distributions based on prior knowledge of the passenger's trip origin and destination, the potential flight for the particular O.D. traffic, as well as rather questionable a priori assumptions on gate assignments for these future flights. The third approach, and the easiest, assumes a random gate assignment. In other words, if the probability of disembarking from Gate k and transferring to Gate j is the same for all gates, then,

$$p_{kj} = p = \frac{1}{N} \qquad \forall k,j = 1,\ldots,N \qquad \text{(Eq. 2.1)}$$

N being the number of gates at the airport.

GATE	71	73	73	74	75	76	77	78	79	80	81	83	85	87	89	91	93	95	97	99
71	0	10	20	310	270	530	420	720	610	910	800	1040	1280	1560	1830	2100	2370	2640	2910	3180
72		0	30	300	230	540	430	730	620	920	810	1050	1290	1570	1840	2110	2380	2650	2920	3190
73			0	310	200	510	400	700	590	890	780	1020	1200	1540	1810	2080	2350	3070	3340	3610
74				0	110	200	330	220	500	330	690	930	1170	1450	1720	1990	2660	2530	2800	3070
75					0	110	200	500	390	690	580	820	1000	1340	1610	1880	2150	2420	2690	2960
76						0	110	190	300	220	490	730	970	1250	1520	1790	2060	2330	2600	2870
77							0	110	190	490	380	620	860	1140	1410	1680	1950	2220	2490	2760
78								0	110	190	300	540	780	1060	1330	1600	1870	2140	2410	2680
79									0	300	190	430	670	950	1220	1490	1760	2030	2300	2570
80										0	110	350	590	870	1140	1410	1680	1950	2200	2490
81											0	240	480	760	1030	1300	1570	1840	2110	2380
83												0	240	520	790	1060	1330	1600	1870	2140
85													0	280	550	820	1090	1360	1630	1900
87														0	270	540	810	1080	1350	1620
89															0	270	540	810	1080	1350
91																0	270	540	810	1080
93																	0	270	540	810
95																		0	270	540
97																			0	270
99																				0

This Side is symmetric
to the other one.

Table 2.7

Matrix of Inter-gate distances

[in feet]

Because of its simplicity, the third approach will be employed. This approach is most valid in this case since no knowledge exists concerning flight connection patterns at Toronto Terminal No. 2.

The expected walking distance d_k^T for a transfer passenger unboarding at Gate k then becomes

$$d_k^T = \sum_{j=1}^{N} P_{kj} W_{kj} = \frac{1}{N} \sum_{j=1}^{N} W_{kj} \quad \forall k=1,..,N \qquad (2.2)$$

where W_{kj} is the kj th element of the intergate distance matrix shown in Table 2.7.

Cases where patterns of connecting flights are usually known can also be accounted for. For instance, if flight A serves a large number of passengers transferring to flight B, then the computer program simulating the algorithm can be easily modified to incorporate a constraint insuring that flights A and B are assigned to nearly gates. In addition, Braaksma's time stamping method can be used to find which flight pairs usually serve the same large number of passengers.

A listing of the computer program used to implement the "Crowdest-Come-Best-Serve" algorithm appears in Appendix A. This listing includes the input data bases containing information on Toronto Terminal No. 2.

3. SOLVING THE PROBLEM AS A LINEAR PROGRAM

The previous chapter describes a heuristic algorithm solution to the walking distance problem at airport terminals. Furthermore, it is shown in Section 2.2 that the algorithm may not necessarily offer an optimal solution. In order to obtain an optimal solution, therefore, a linear programming approach is introduced in this chapter.

3.1 Formulation of the Linear Program

(A) The Objective Function

The objective is to minimize the average walking distance per passenger, or the total of all distances walked by passengers,

$$\text{Min } Z = \sum_{j=1}^{N} \sum_{i=1}^{M} \{P_i \, d_j \, x_{ij}\} \qquad (3.1)$$

where M is the total number of flights,

N is the total number of gates,

P_i is the total number of passengers boarding to or unboarding from flight i ,

d_j is the expectation of the measured airport terminal walking distance per passenger.

and the decision variable

$$X_{ij} = \begin{cases} 1 & \text{if flight i is assigned to gate j} \\ 0 & \text{otherwise} \end{cases}$$

Here, X_{ij} is a binary variable. If, for instance, flight 1 is not assigned to gate 3, $x_{13} = 0$ and the product term $P_1 d_3$ vanishes.

The number of passengers on any flight, P_i, depends as in the case of the "Crowdest-Come-Best-Serve" algorithm, on the type of carrier used by that flight. If flight i is a Boeing 747, for instance, then under the assumed 65% load factor, $P_i = 248$ (See Table 2.5 in Section 2.1.1).

The mean distance d_j a passenger using gate j has to walk is a weighted sum of the walking distance for the three types of passengers: arriving, departing, and transferring. Thus,

$$d_j = .35d_j^a + .5d_j^a + .15d_j^T \qquad (3.2)$$

where the superscripts a, d, and t denote, respectively, arriving, departing and transferring distances. The weighting factors .35, .5, and .15 represent the probabilities that the random passenger is respectively, arriving, departing or connecting. These probabilities are derived and explained in Section 2.3.1. Finally, each distance in Equation 3.2 can be obtained from one of the entries of either Table 2.5 or 2.6 in Section 2.3.2.

Equation 3.1 gives more importance to one flight over the other only if that flight carries more passengers. Other factors of importance can be introduced in the objective function. If, for instance, the terminal's management feels that flights normally carrying buisness-men are more important than other flights,then a scaling factor can be added to the product $p_i d_j$. More succintly, the objective function would become

$$\text{Min } Z = \sum_{i=1}^{N} \sum_{j=1}^{N} \gamma_i P_i \, d_j \, x_{ij} \qquad (3.3)$$

where γ_i is the importance factor for flight i. The linear program will then reduce more the average walking distance of flights with higher importance factors. Since no knowledge exists concerning how the management at Toronto International views the various flights, the objective function of equation 3.1 will be used.

(B) <u>The Constraints</u>

Two classes of constraints exist for the gate assignment problem at airports: those which are physical and inherent to the problem and those which depend on the airport management or the airline using the terminal. The first class of constraints are necessary for the flight-to-gate assignment to meet the following two conditions:

1. Every flight must be assigned to exactly one gate, and
2. No two airplanes can occupy the same gate concurrently.

The second class of constraints deals with problems which vary from one airport to the other. For instance, certain gates can only serve one flight category, such as international flights, or some aircraft types are too big for certain gates.

<u>Constraints inherent to the assignment problem</u>:

1. Every flight must be assigned to exactly one gate:

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad \forall\ i = 1,\ldots,M \tag{3.4}$$

For each flight i, the sum of all gates j assigned to that flight must equal 1 . There are as many of those constraints as there are flights, M.

2. No two flights may occupy the same gate concurrently:

To formulate this constraint, a set covering method is used. Assume that flights are indexed in order of their arrival time. For each flight i, define the set L(i) , whose elements are themselves flights, as follows:

$$L(i) = \{\ell \mid t_\ell^a + t_\ell^g \geq t_i^a,\ \ell=1,\ldots i-1\}$$

$$= \{\ell \mid t_\ell^a + t_\ell^g \geq t_i^a\ ,\ \ell \varepsilon L(i-1)\} \tag{3.5}$$

where t_ℓ^a = Arrival time for flight ℓ

and

t_ℓ^g = ground stay time of flight ℓ.

Note that $t_\ell^a + t_\ell^g$ is actually the departing time for flight ℓ. Since flights are indexed in their order of arrival, the set L(i) thus consists of all flights landing before flight i and still on the ground when that flight arrives. This set is defined recursively. That is, of all flights preceeding flight i, one needs only consider those belonging to L(i-1) , together with flight i-1 itself, in order to construct the set L(i) . Note also that L(0) is the empty set.

The conflict constraints are thus described as follows:

$$\sum_{\ell \in L(i)} X_{\ell j} + X_{ij} \leq 1 \quad \forall i=1,\ldots,M \quad j=1,\ldots,N \tag{3.6}$$

Equation 3.6 says that if any flight ℓ conflicts in time with flight i, it cannot be assigned to the same gate j. These constraints come in inequality form in order to express the fact that some gates do not necessarily have to be used at all times.

The conflict sets generate at most a total of ([M-1]xN) constraints where, as before, M is the total number of flights and N is the total number of gates. Thus, in addition to the first M constraints, there are ([M-1]xN) total constraints. For the case of Toronto Terminal No. 2, the total number of constraints is

$$([M-1]xN)+M = (138x20) + 138 = 2,878$$

A simple example, however, will demonstrate that many of these constraints can be redundant and should, therefore, be dropped.

Assume that the pth arriving flight conflicts only with the three previous flights. Then $L(p) = \{p-3,p-2,p-1\}$ and the corresponding conflict constraint for any gate j , is

$$\sum_{\ell \varepsilon L(p)} x_{\ell j} + x_{pj} = x_{p-3,j}+x_{p-2,j}+x_{p-1,j}+x_{p,j} \leq 1 \qquad (3.7a)$$

Assume further that the p+1st flight arrives and none of the four flights already on the ground leaves. That is $L(p+1) = \{p-3,\ldots,p\}$. For each gate, then

$$\sum_{\ell \varepsilon L(p+1)} x_{\ell j} + x_{p+1,j} =$$

$$= x_{p-3,j}+x_{p-2,j}+x_{p-1,j}+x_{p,j}+x_{p+1,j} \leq 1 \qquad (3.7b)$$

Here, $L(p) \subset L(p+1)$ and it is clear that any solution satisfying equation 3.8b will automatically satisfy equation 3.8a. The constraints generated by the pth flight can therefore be dropped. For an airport with 20 gates, this means 20 less constraints. The above type of redundancy in constraints occurs when one or more flights land before any flight on the ground takes off. The following theorem shows that if a series of flights land consecutively without any departures occurring between them, then the corresponding conflict sets are nested:

Theorem: If $L(i) \subset L(i+k)$, for any $k=2,\ldots,M-i+1$,

then $L(i) \subset L(i+1) \subset \ldots \subset L(i+k)$

Proof: Assume that $L(i+r) \subset L(i+r+1)$ for some $r = 0,\ldots,k-1$. Then $\exists \ell = f$ such that $f \in L(i+r)$ but $f \notin L(i+r+1)$. From the definition of the sets $L(i)$, this means that

$$t_f^a + t_f^g < t_{i+r+1}^a$$

and since the flights are indexed in their arrival order, $t_{i+k}^a \geq t_{i+r+1}^a$ and

$$t_f^a + t_f^g < t_{i+k}^a$$

or $f \notin L(i+k)$. This contradicts the hypothesis that $L(i)$ is a subset of $L(i+k)$ and thus completes the proof.

Q.E.D.

This simple theorem actually helps recognize redundant constraints. If, for instance, $L(3) \subset L(7)$, then the constraints generated by the third through sixth flight are redundant and their omission will not alter the set of feasible solutions to the linear program. The example in the next section will illustrate by how much does the elimination of such redundant constraints reduce the computational burden associated with the problem.

Additional Constraints

In addition to the two types of constraints inherent to the assignment problem, other additional constraints, which depend on the individual airport, are now introduced.

3. Flights are to be assigned to nearby gates.

The desire to have such a constraint arises when it is known that two or more flights serve the same large number of connecting passengers. Because of the assumption of random gate assignment explained in Section 2.3.2 , the LP does not necessarily position connecting flights in nearby positions. Namely, it is assumed that a transfer passenger landing in gate k is equally likely to find his connecting flight at any other gate. This assumption, however, is not always valid. In the case where two or more flights serve the same transfers, passenger movements occur in group, that is, from the landing flight's gate to one or more specific gates. The expected walking distance d_k^t of equation 2.2 (Section 2.3.2), whose derivation assumes random assignment, is therefore not valid when such situations occur.

Braaksma's time-stamping approach, explained in Section 2.3.2, can be used to discover if any two or more flights actually serve the same transferring passengers. If it is found, for instance, that flights r and ℓ are serving a large number of the same passengers, then the program as originally formulated should first be solved. If these flights are assigned to gates too distant, then the folloiwng can be done. Fix one of the flights, say flight ℓ, to the gate assigned to it by the linear program, say gate z . Thus, fix $X_{\ell z} = 1$ and add the following constraint:

$$\sum_{j=1}^{N} X_{\ell j} W_{zj} \leq D \qquad (3.8)$$

where D is the maximum distance permitted between the two flight's gates and W_{zj} is the intergate distance between gates z and j . Since this constraint was introduced when the problem was already optimal, the additional number of iterations required to satisfy this constraint and return to an optimal basis would be negligible.

The method described above would bring flight r to a gate within a distance D of flight ℓ's, or gate z . If, as a result of introducing this constraint, the value of the optimal solution is greatly increased (which also mean a very high shadow price for the right hand-side D), then the described procedure should be tried by reversing the two flights' roles. In other words, after returning to the original optimal basis, one should fix fight r to its gate and attempt to bring flight ℓ nearby.

Looking at the shadow price information given by the program may also be helpful. This information normally accompanies the output to the linear program. If the right-hand-side for which the high shadow price is valid has an upper bound rather close to D, and if the shadow price drops significantly beyond that range, then relaxing the constraint equation 3.9 by increasing the value of D to a value slightly above the upper bound of the right-hand-side range, would improve the optimal solution. The disadvantage, of course, would be that the two flights are placed further apart than originally desired, i.e., at a distance greater than D .

If several pairs of flights like flights r and ℓ exist, then for each pair, a constraint equation like that of 3.8 should be introduced along with the fixing of one of its flights to its gate.

Finally, it is possible to set a constraint fixing the two aircraft to close-by gates prior to solving the problem. This constraint, written in equation 3.10, however, is not linear and cannot be easily implemented on the computer.

$$\sum_{j=1}^{N} \sum_{i=1}^{N} X_{\ell z} W_{zs} X_{rs} \leq D \qquad (3.10)$$

4. Subdivision of the airport into separate airline areas:

Most U.S. airports are divided into several areas where each area is reserved for the exclusive use of a particular airline. If S airlines are using the terminal, then the set j of all gates and the set I of all flights can be partitioned as follows:

$$I = \{I_1, \ldots, I_s, \ldots I, _S\} \qquad (3.11a)$$

and

$$\{J = J_1, \ldots, J_s, \ldots, J, _S\} \qquad (3.11b)$$

Each pair of subsets I_s of I and J_s of J can then be treated treated as separate airports, i.e., since the I's and the J's are both mutually exclusive, the problem can be subdivided into S linear programs.

However, proponents of shared airport terminal facilities argue, justifiably, that if walking distances are to be significantly reduced, the practice of dividing the airport into airline areas must be abandoned.

5. Restricting the use of some aircraft at specified gates.

This type of consideration can be taken into account by simply setting the appropriate decision variable to zero. For instance, if gate 73 does not have the facilities for jumbo jets, then, set $X_{\ell 73} = 0$, for all flights ℓ with a B747 .

Other considerations also exist and can, in most cases, be easily incorporated as constraints into the linear program.

3.2 Solving an Example Program for a Small Airport

In order to best visualize the shape of the constraint matrix A, a small problem is solved in this section. The hypothetical airport consists of three gates. Five flights are to be served within one hour. Table 3.1 lists the average walking distance assumed for each gate d_j while the necessary flight information appears in Table 3.2 . Furthermore, all flights are eligible to be assigned to any gate.

The diagram of Figure 3.1 helps recognize the conflicts sets L(i), i = 1,...5. In this diagram, the time table for the airport is shown. The third flight arrives before any of the first two flights already on the ground leave. The conflict set for the third flight L(3), is therefore a superset of L(2), the conflict set for the second flight. More succintly

$$L(3) = \{1,2\} \supset L(2) = \{1\}$$

The elements of each conflict set are, of course, flights. Following the reasoning of the last section, any solution which satisfies the conflict constraints generated by the third flight should thus satisfy those generated by the second flight.

GATE	AVERAGE WALKING DISTANCE PER PASSENGER d_j (in feet)
1	1000
2	2400
3	3000

Table 3.1 Average Gate Walking Distance per Passenger (in feet) for Hypothetical Airport

FLIGHT	ARRIVAL TIME	DEPARTURE TIME	PASSENGERS
1	00:00	00:25	400
2	00:10	00:40	200
3	00:20	00:50	100
4	00:30	00:44	100
5	00:45	00:100	250

Table 3.2 Flight Information for Example Problem

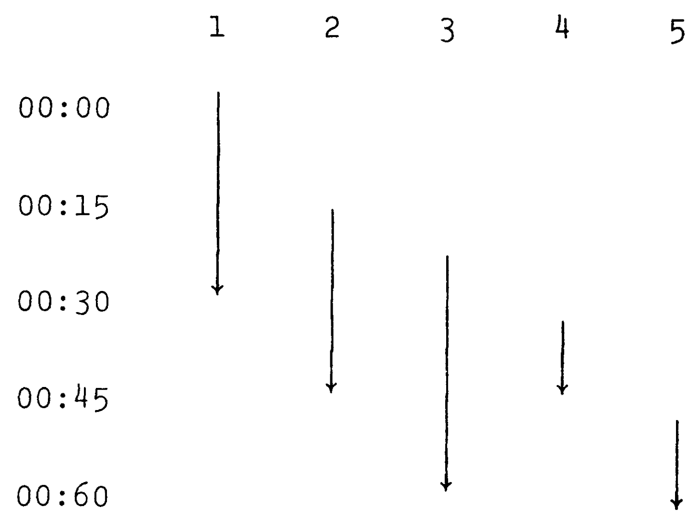


Figure 3.1 Diagram showing conflict sets L(i), i=1 to 5 for example problem

Now, the first flight leaves before the fourth flight arrives. Hence, $\{1\} \notin L(4)$ and $L(3) \not\subset L(4)$. The constraints generated by the third flight are not, therefore, redundant. Similarly, the fourth flight leaves before the fifth flight arrives and $L(4) \not\subset L(5)$.

A look at the formulation presented now verifies the assertions of the last two paragraphs.

$$\text{Min } Z = \sum_{i=1}^{5} \sum_{j=1}^{3} d_j \, p_i \, X_{ij}$$

S.T.

<u>1st Type of Constraints</u>: $\sum_j X_{ij} = 1 \quad \forall_i$

$$
\begin{array}{lllll}
X_{11}+X_{12}+X_{13} & & & & = 1 \\
 & X_{21}+X_{22}+X_{23} & & & = 1 \\
 & & X_{31}+X_{32}+X_{33} & & = 1 \\
 & & & X_{41}+X_{42}+X_{43} & = 1 \\
 & & & & X_{51}+X_{52}+X_{53} = 1
\end{array}
$$

<u>2nd Type of Constraints:</u> $\sum_{\ell \in L(i)} X_{\ell j} + X_{ij} \leq 1 \quad \forall i,j$

$L(1) = \emptyset$ ≤ 1

$L(2) = \{1\}$ ≤ 1

$$X_{11} + X_{21} \leq 1$$
$$X_{12} + X_{22} \leq 1$$
$$X_{13} + X_{23} \leq 1$$

$L(2) \subset L(3)$ redundant constraints

$L(3) = \{1,2\}$

$$X_{11} \quad X_{21} + X_{31} \leq 1$$
$$+X_{12} + X_{22} + X_{32} \leq 1$$
$$+X_{13} + X_{23} + X_{33} \leq 1$$

$L(4) = \{2,3\}$

$$X_{21} + X_{31} + X_{41} \leq 1$$
$$+X_{22} + X_{32} + X_{42} \leq 1$$
$$+X_{23} + X_{33} + X_{43} \leq 1$$

$L(5) = \{3\}$

$$+X_{31} + X_{51} \leq 1$$
$$+X_{32} + X_{52} \leq 1$$
$$X_{33} \quad X_{53} \leq 1$$

$X_{ij} = 0,1 \qquad \forall_i = 1,\ldots,5 \qquad j = 1,..3$

One can obtain a solution to this problem by inspection. The optimal solution appears in Table 3.3. The average walking distance per passenger is also shown for each flight. The optimal value of the objective function, i.e., the minimum total of all walking distances is 15,300 feet, or an average of 1,450 feet per passenger.

This problem was also solved on SESAME. Two remarks are noteworthy. The first one concerns the redundant constraints. The problem was solved twice on SESAME. Once with the redundant constraints and once without them. It was found that dropping the redundant constraints reduced the number of simplex iterations from fourteen to seven. Originally, the constraints numbered ([M-1]XN)+M=(4x3)+5 = 17 . If the three redundant conflict constraints generated by the second flight (see Figure 3.1) are dropped, 14 constraints would be left. Thus, a reduction of 3 constraints gave a 50% reduction in the number of iterations. Such improvement

---

SESAME is an interactive computer software package used to solve this problem for Toronto Terminal No. 2. This system has been designed at the Computer Research Center of the National Bureau of Economic Research (NBER) and and is used in conjunction with the VM/CMS Operating System of the IBM 370 computer.

FLIGHT	GATE	AVERAGE WALKING DISTANCE	OBJECTIVE FUNCTION TERM
1	1	1,000	400,000
2	2	2,400	480,000
3	3	3,000	300,000
4	1	1,000	100,000
5	1	1,000	250,000

Table 3.3 Optimal Gate Assignment and Walking Distances for Each Flight For Example Problem

in the computational efficiency of a solution is common especially when degeneracies, and therefore cycling, are eliminated. A decrease in the execution time and cost should be expected since these two factors grow exponentially with the number of constraints.

The second remark regards the integrality of the decision variable x_{ij} . The simplex procedure gives an integral optimal solution ($x_{ij} = 0$ or 1, for $i = 1$ to M, $j = 1$ to N). A sufficient condition for obtaining an integral optimal solution is the total unimodularity of the constraints matrix A. A matrix is totally unimodular when the determinant of everyone of its submatrices equals 0, -1, or 1. Hoffman and Kruskal [1956] proved that every extreme point of the convex polyhedra $\{x \mid Ax \leq b\}$ is integral if and only if the matrix A is totally unimodular. Unimodularity exists, for instance, in the constraint matrices of transportation problems.

Because the optimal solution is integral, no need exists to utilize any integer programming technique such as the Branch and Bound Algorithm or the Subgradient Optimization Algorithm. Unimodularity is also of interest because the solution to the linear program for Toronto Terminal No. 2 is integral. It remains to be determined, however, whether a formulation similar to the one described in Section 3.1 always leads to a unimodular matrix A.

3.3 Implementation of the Model on the Computer

The linear program defined in Section 3.1 was solved for the schedule of Toronto Terminal No. 2 using the interactive software package SESAME. Within SESAME itself, several procedures exist. One of these procedures, called DATAMAT, is actually a computer language used in conjunction with SESAME. DATAMAT is used for model generation, problem revision, parametric studies and report generation. To develop the linear programming model for the gate assignment problem, a program was written in the DATAMAT language. The flight and passenger information for Toronto Terminal No. 2, as well as the gate distances, are contained in two tables which serve as input to the model generator (also called the preprocessor). The preprocessor program appears in Appendix C.

For the present study, the preprocessor generated constraints of the first two types derived in equation 3.4 and 3.5 in Section 3.1. These constraints, which are inherent to the assignment problem, are: 1) Every flight must be assigned to exactly one gate and (2) No two aircraft may occupy the same gate concurrently. Constraints which depend on the individual airport can be programmed into the same model. The input data bases for the model are cited in Section 2.3.

The flight schedule used to test this model generated 1,318 constraints and 4,078 variables. The number of constraints indicates that there are 59 non-nested conflict sets. Each one of these sets generates 20 constraints, one for each gate. There are thus 59 x 20 = 1,180 conflict constraints. The remaining 138 constraints correspond to those of the first type.

Of the 4078 variables, 2760 are decision variables (X_{ij}'s), corresponding to every possible combination from 138 flights and 20 gates. The remaining 1318 variable are slack and artificial variables, one for each constraint in the model.

4. RESULTS

The flight-to-gate allocations vary in accordance with the particular method of solution used to solve the problem. The two solution methods give different results and accrue different costs. This chapter first discusses and compares the results of the two methods against the actual flight-to-gate assignments. Next, a discussion on the cost associated with each method follows. Due to the high computational cost of implementing the linear program and to the shortage of available data, only one test was made. As mentioned in Section 2.3, the data for this test consisted of one day in the summer of 1976 at Terminal No. 2 of Toronto International Airport. The chapter ends with a discussion surrounding the use of the algorithm vs. the LP.

4.1 Comparison of the Two Methods of Solution

In order to compare, analyze and tabulate the results of each of the two solution methods, the algorithm and the linear program, a computer program was written in the Datamat Language. This postprocessor lists for each flight the gate and the corresponding walking distance for each of the three assignment policies: Air Canada's actual assignment, the heuristic algorithm and the linear program. The postprocessor program produces a separate flight-by-flight listing of walking distances for each of the three categories of passengers: arriving, departing and transferring. A fourth listing gives the weighted mean walking distance for all three categories.

In addition, the program supplies statistical distributions for the mean walking distance of each of the three categories of passengers, as well as for the weighted average walking distance. A listing of the postprocessor program appears in Appendix D.

Solutions to the flight-to-gate assignment problem appear in Appendix E. Table E.1 gives the overall mean walking distance and gate position for each flight under each of the three assignment policies, while Tables E.2 - E.4 give the same information for each individual category of passengers separately. In addition Tables E.5- E.8 list the statistical distributions of the walking distances. These tables were used to build the four graphs of figures 4.1 through 4.4.

Figure 4.1 shows the cumulative distribution of the weighted average walking distances for all passengers resulting from each of the three assignment policies. The cumulative percentage of passengers is plotted against the average walking distance. Since the objective is the minimization of the walking distance, the distribution located to the extreme left will give the best results. This distribution is, as expected, the results of the linear program. The LP offers a mean walking distance of 608 ft. while the original (Air Canada's) airport assignment gives a mean of 803 feet, a difference of 195 feet, or a savings of 32%. The "Crowdest-Come-Best-Serve" algorithm offers an assignment with a mean of 632 feet per passenger; that is, a saving of 27% over the original assignment. In the case of Toronto Terminal No. 2, therefore, the algorithm is only 5 percent suboptimal. This information is summarized in Table 4.1a.

The graph also indicates that under the original assignment, 99 percent of the passengers walked an expected distance of 1,300 feet or less. If the algorithm's assignment is implemented, the same percentage of passengers would have walked 1,100 feet or less. The same distance for the linear program measures 1,083 feet. Table 4.1b shows various percentiles for each policy.

Cumulative distributions for each of the three categories of passengers are shown in Figures 4.2, 4.3, and 4.4. The greatest savings in walking distance goes to the departing passenger, or 34% under the linear program's assignment and 31% under the algorithm's. This is due to the fact that departing passengers comprise the largest single category of passengers or 50% of a total number of 28,378 air travellers. Their walking distance, therefore, carries the heaviest single weight on the objective

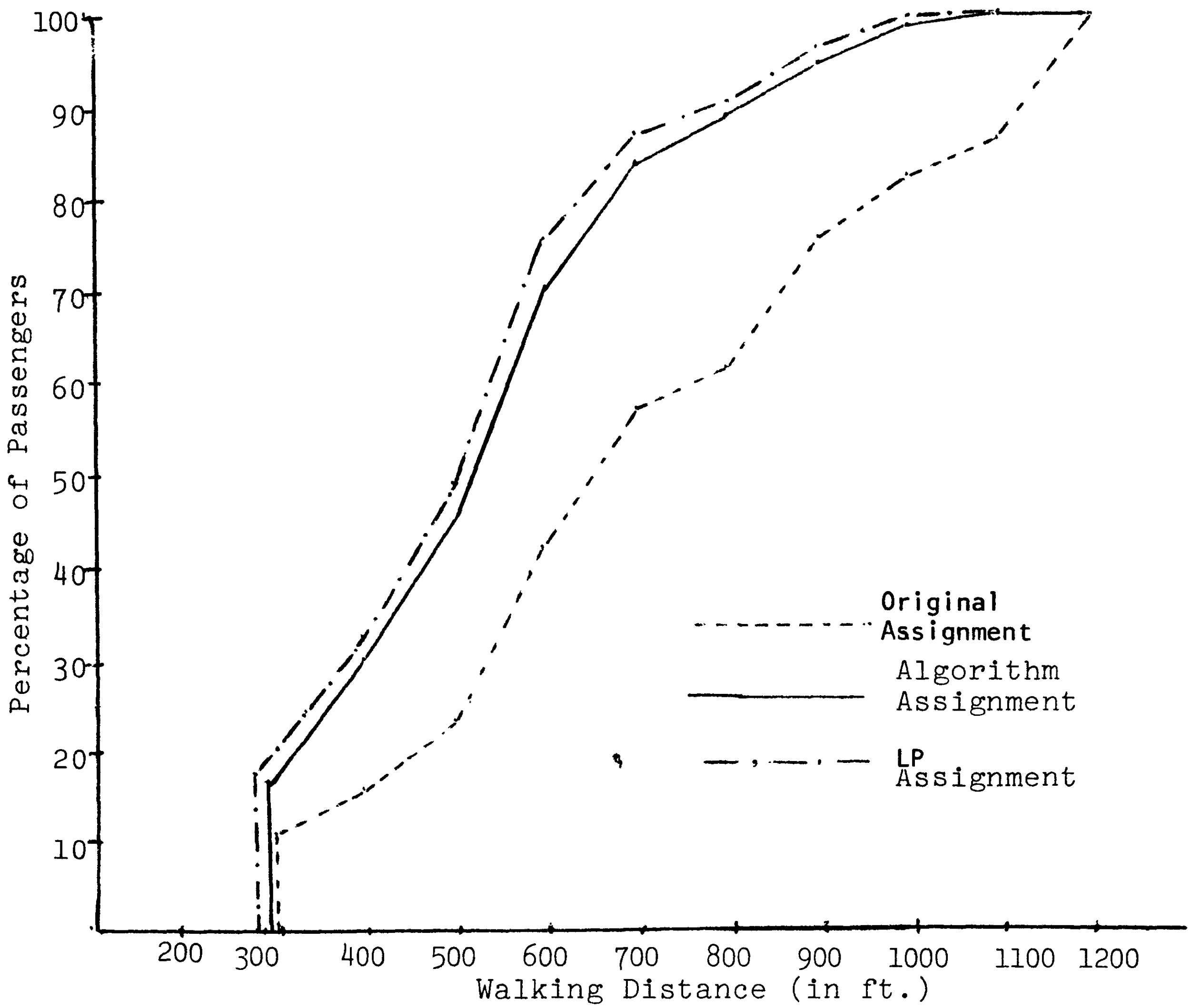


Fig. 4.1 Cumulative Distributions of the Overall Mean Walking Distance for All Passengers under each of the Three Different Assignment Policies

		MEAN SAVINGS	PERCENTAGE SAVINGS
		(Compared to Original)	
Original	803	—	
Algorithm	632	171	27%
Linear Program	608	195	32%

Table 4.1a Mean and Mean Saving in the Expected Distance for All Passengers (in feet) under the Three Assignment Policies

	Percentile			
	25th	50th	75th	99th
Original	617	750	1,000	1,300
Algorithm	460	617	735	1,100
Linear Program	450	600	700	1,083

Table 4.1b Percentiles of Expected Walking Distances for All Passengers Under the Three Assignment Policies

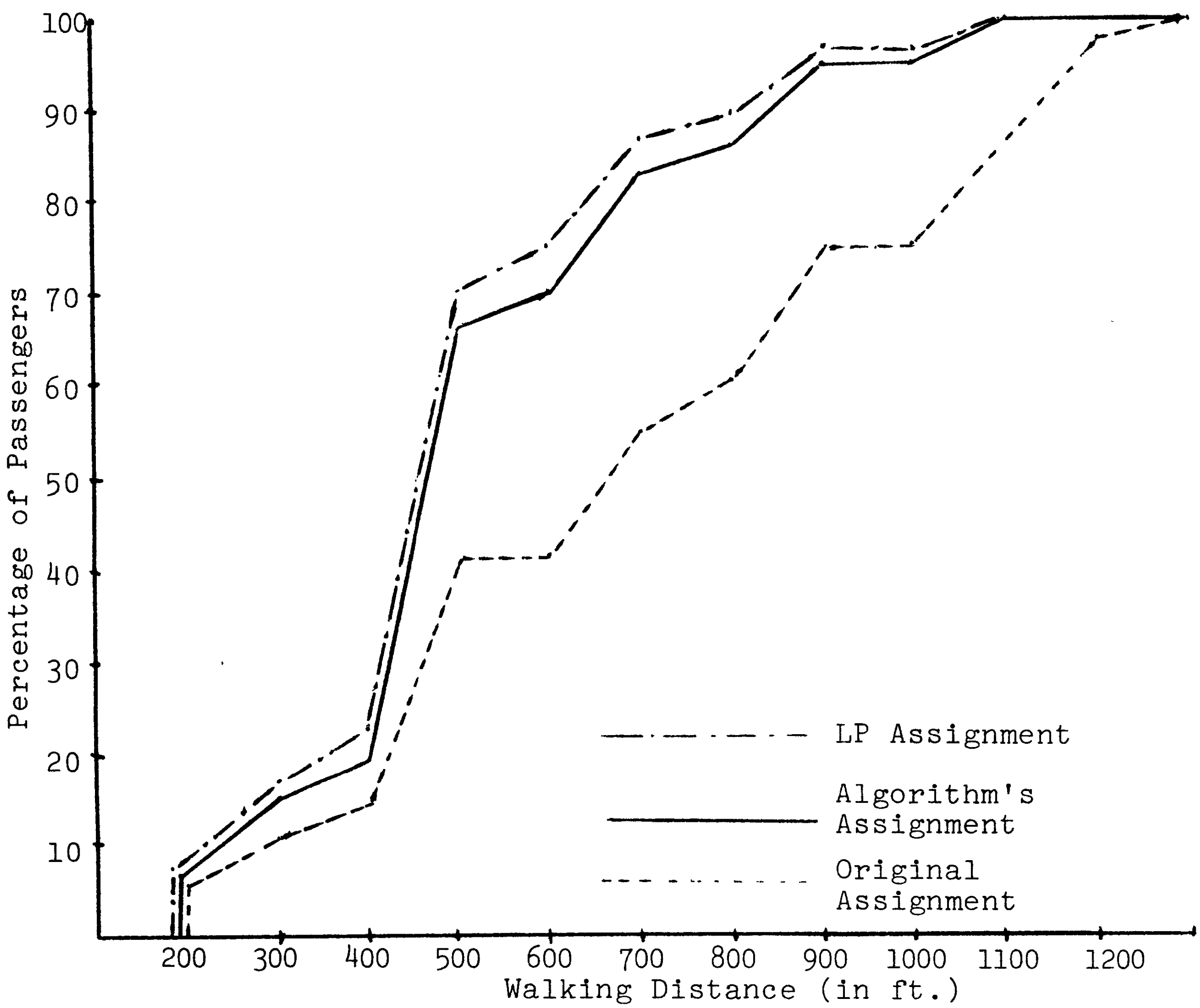


Fig. 4.2 Cumulative Distributions of the Expected Walking Distance for Arriving Passengers under Each of the Three Assignment Policies

	Mean	Mean Savings	Percentage Saving
		(Compared to Original)	
Original	784	——	
Algorithm	608	176	22%
Linear Program	582	202	26%

Table 4.2a Mean and Mean Saving in Expected Distance for Arriving Passengers (in feet) Under the Three Assignment Policies

	Percentile			
	25th	50th	75th	99th
Original	540	765	1,000	1,300
Algorithm	517	567	743	1,200
Linear Program	507	540	700	1,200

Table 4.2b Percentiles of Expected Walking Distances for Arriving Passengers Under the Three Different Assignment Policies

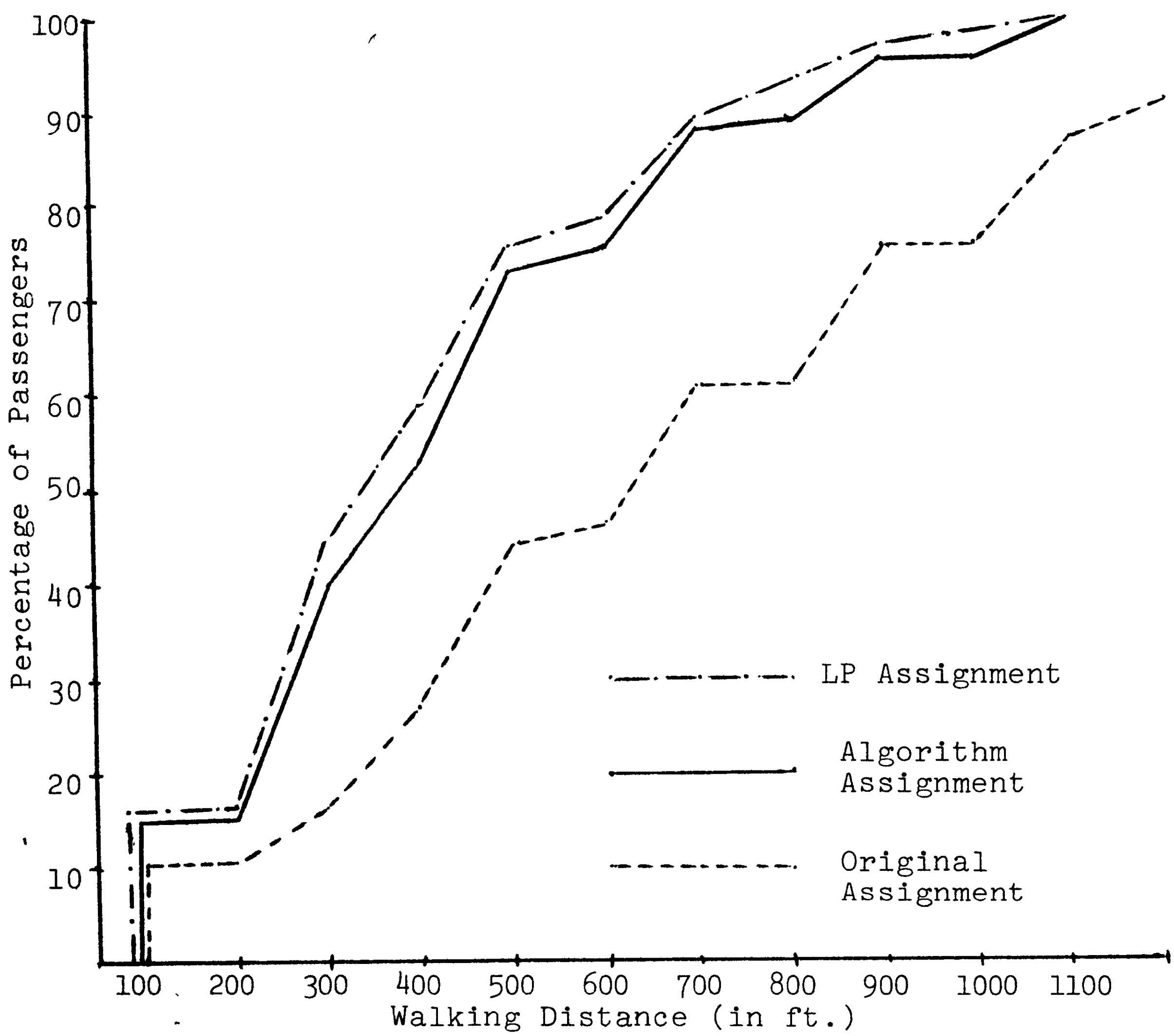


Fig. 4.3 Cumulative Distribution of the Expected Walking Distance for Departing Passengers under Each of the Three Assignment Policies

function. Figures 4.2 and 4.3 show the cumulative distributions for arriving and departing passengers while Tables 4.2 and 4.3 summarize the statistics for these graphs.

Figure 4.4 shows the distribution in walking distances for transfer passengers under each policy. The three graphs have similar distributions and therefore, transfer passengers do not necessarily gain any savings as a result of a change in assignment policy. In fact, the linear program gives a 1% increase over the original assignment in the expected walking distance of a transfer passenger and the algorithm gives a 4% increase. Tables 4.4a and 4.4b summarize these results. Two potential explanations can be given. First, connecting passengers comprise only 15% of the total number of passengers. This low ratio is reflected in the average walking distance for any passenger derived in equation 3.2 (rewritten below)

$$d_j = .35d_j^a + .5d_j^d + .15d_j^t \qquad (3.2)$$

Second, even if connecting passengers are given a heavier weight in the objective function, the improved numerical results, if any occur, would not necessarily reflect the actual situation. It was mentioned in Section 3.1 that the random gate assumption is valid only in the absence of any information concerning connecting flights. These are flights which serve the same large number of transfer

	Mean	Mean Saving	Percentage Saving
		(Compared to Original)	
Original	744	—	—
Algorithm	512	232	31%
Linear Program	492	252	34%

Table 4.3a Mean and Mean Saving in Expected Walking Distance for Departing Passengers under Each of the Three Assignment Policies

	Percentile			
	25th	50th	75th	99th
Original	483	720	1,000	1,400
Algorithm	335	467	636	1,173
Linear Program	220	433	583	1,167

Table 4.3b Percentiles of Expected Walking Distance for Departing Passengers Under Each of the Three Policies

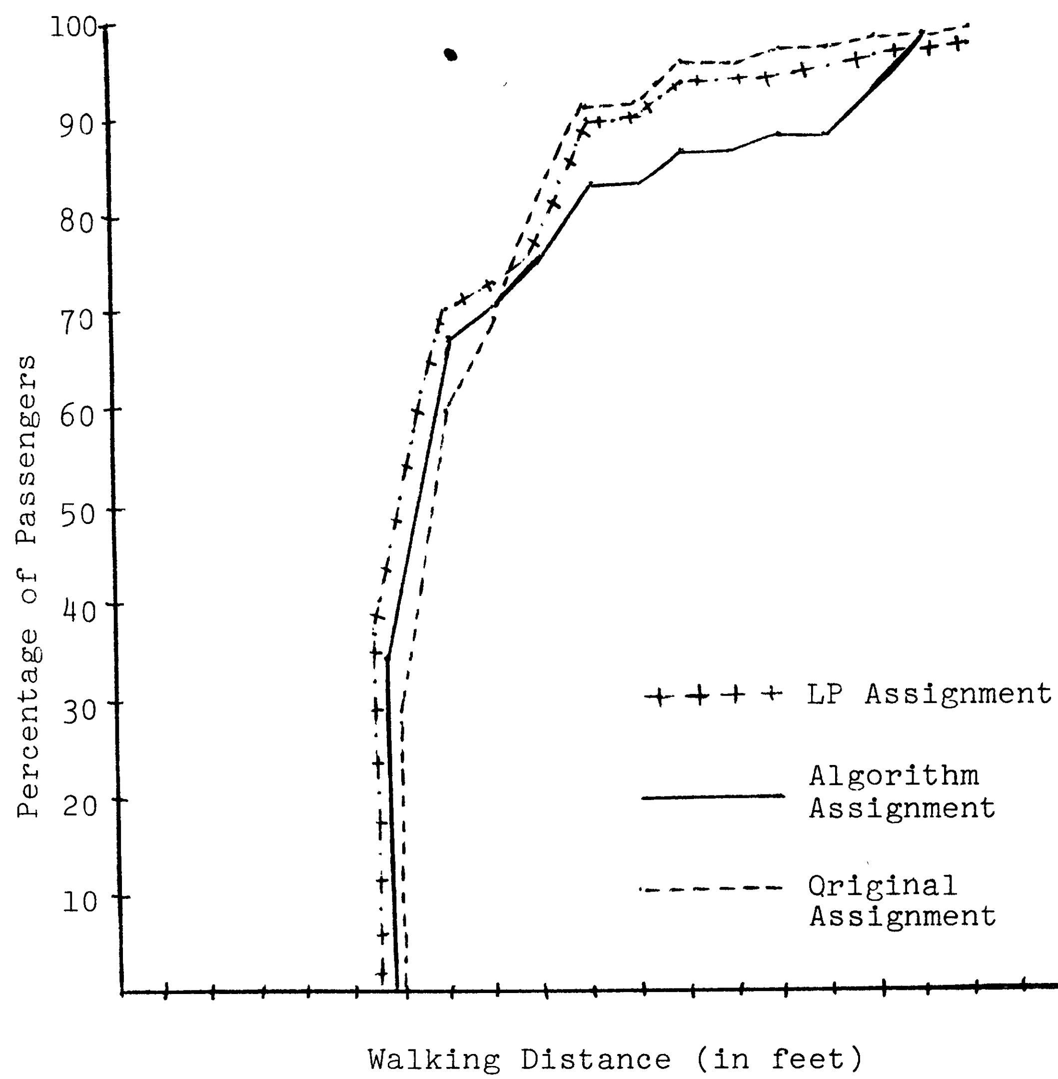


Fig. 4.4 Cumulative Distributions of the Expected Walking Distance for Transfer Passengers Under Each of the Three Assignment Policies

	Mean	Mean Difference	Percentage Difference
		(Compared to original)	
Original	1045	—	
Algorithm	1091	-46	-4%
Linear Program	1062	-17	-1%

Table 4.4a Mean and Mean Difference in Walking Distance for Transfer Passengers Under each of the Three Assignment Policies

	25th	50th	75th	99th
Original	900	930	1,120	1,900
Algorithm	900	920	1,150	2,100
Linear Program	900	920	1,100	2,100

Table 4.4b Percentiles of Expected Walking Distances for Transfer Passengers under the Three Assignment Policies

passengers. Such passengers leave their landing gate to a specific other gate or gates in order to board their next plane. Contrary to the implications of the random gate assignment assumption, any transfer passenger in this situation does not have his next flight assigned to any of the twenty gates at the terminal with equal probability.

Braaksma's "time-stamping" approach can be used to recognize if any two or more flights serve the same transfer passengers. Once such information is known, it is essential to insure that these flights are positioned in nearby gates. This can be done by adding one or more constraints as explained in Section 3.1.

4.2 Computational Costs

Though both the algorithm and the LP have similar results, the difference in the cost of computation is substantial. The computer program which simulates the heuristic algorithm was written in Fortran IV on an IBM/370 VS1 batch facility. The linear program was implemented on SESAME, a subenvironment of the CMS operating system, which also operates on the IBM/370. The reader should note that though the computer used to implement both the algorithm and the LP is the same, the operating systems are different.

The LP was implemented twice, once with no initial basic feasible solution and the second time, using the algorithm's assignment solution as an initial basis. In the first case, the simplex method took 1,296 iterations to arrive at optimality and in the second, the number of iterations was reduced to 605. The reason for the disparity is that in the first case, a very large number of iterations is necessary to eliminate the primal infeasibilities (or the artificial variables added to the equality constraints) while in the second case, a primal feasible basis already exists.

The simplex method is but the last of three steps essential to obtaining an optimal solution. The first step is the model construction. As mentioned in Section 3.3, the constraint matrix size is 1,318 rows and 4,078 columns. The second step consists of copying the model from the active file into a permanent model file.

Implementation of the algorithm costs approximately $3.15. The total CPU time is 3.40 seconds and the total storage space-time used is 4,231 knet sec. In addition, other costs such as printing exist. Table 4.5a contains an item-by-item cost list for running the computer program used.

For running the linear program, the resources used and the costs vary with the time of day and number of users in the system. Table 4.5b shows cost estimates for each of SESAME's steps. The numbers in this table are round <u>on purpose</u>. Different costs can be obtained during different computer runs. The only certain conclusion that the reader

VS1 Resource	Cost
CPU Time 3.40 seconds @ $1.667/sec.	.57
Virtual Core 4.231 knet sec. @ $.00014/KNS	.59
Subtotal	1.16
802 printer lines @ $1.55 per 1,000 lines	1.24
Subtotal	2.40
Adjustment for day shift and standard priority	.75
	3.15

Table 4.5a Resource Utilization and Their Costs for the "Crowdest-Come-Best-Serve" Algorithm (1979-1980)

	Cost (No initial feasible basis)	Cost (Algorithm's basis Used)
Model Development	$150	$150
Model Permanent File Rewriting	$120	$120
Simplex Method	$210	$ 40
TOTAL	$480	$310

Table 4.5b Very Approximate Costs for Running the Linear Program

can draw from Table 4.5b is the following: while the heuristic algorithm's costs amount to less than $10, the linear program's costs are between $300 and $500.

Though the expenses associated with the heuristic algorithm are negligible, its solution is suboptimal. There is no guarantee that the excellent performance of the algorithm in the case of Toronto Terminal No.2 is reproducible. In fact, the only way to determine the algorithm's degree of suboptimality (5% in Toronto's case) is to solve the linear program and compare the answers. A priori, these results, however, may not justify the added costs. A reasonable approach, therefore, could be the following:

1. First, solve the "Crowdest-Come-Best-Serve" algorithm and obtain a solution.

2. If the savings from the algorithm's assignment proves to be satisfactory, then no need exists to solve the linear program.

3. If the heuristic algorithm's assignments do not offer sufficient savings in passengers' walking distances, and if by inspecting the solution many improvements can be detected, then the linear program should be solved. Of course, the algorithm's assignment should be used as an initial basic feasible solution in the linear program.

Once the model is developed and stored in a pera-nent file using DATAMAT, then the Simplex procedure of any software package can be used. It is possible, for example, to utilize the IBM MPSX/370 package, which may be more efficient, and therefore, less expensive. Finally, since DATAMAT performs a large number of disk input-output (I/O) operations, a very large storage (I M bytes or more) and the largest permissible block size must be used in order to keep the associated costs as low as possible.

5. CONCLUSION

The present work aimed at solving the flight-to gate assignment problem at airport terminals in such a way as to minimize, or at least reduce, the expected walking distance per passenger. Two solution methods were used. The first is the "Crowdest-Come-Best-Serve" algorithm which simply allocates the best gate to the aircraft with the largest number of on-board passengers. The second method consists of formulating the problem as a linear program. Both methods were tested on a flight schedule from one day during the summer of 1976 at Terminal No. 2 of Toronto International Airport.

The algorithm's assignment gave an expected walking distance of 632 feet per passenger for a random passenger, as opposed to 784 feet under the original airport assignment, a saving of 27%. The linear program's assignment offered an optimal walking distance of 582 feet per passenger, or a saving of 32%. Results were also obtained for each of the three categories separately. Though the walking distance for the connecting passengers did not significantly change when either of the two solution methods were used (mainly because of the low ratio of connecting passengers to total passengers), means to improve the situation were suggested.

Though the algorithm, which is the cheaper of the two solution methods, performed at a 95% optimal level at Toronto, such excellent results cannot be guaranteed for every case. For this reason, a strategy which helps the analyst decide between the algorithm and the linear program was presented.

Both the algorithm and the linear program can be useful for other applications. For instance, other objective functions such as minimizing congestion in any one area of the airport can be formulated and used with the linear programming model. Also, the same model could possibly be used for optimizing core memory allocation in a computer, or for bus stations in some large metropolitans such as Tel Aviv and Rome.

Finally, deviations from schedule can be incorporated into either the algorithm or the linear program.

APPENDIX A

COMPUTER PROGRAM IMPLEMENTING THE "CROWDEST-COME-BEST-SERVE" ALGORITHM

(Written in Fortran IV)

FILE: ALGO   VS1JOB   A

```
//LODA JOB LOD,
// PROFILE='DEFER', MEMORY=150K,
// TIME=(0,10)
//*PASSWORD DJEBEL
// EXEC FTG1CLG,PRINT='PRINT'
//FORT.SYSIN DD *
C      DECLARATIONS
C
       DATA BLANK/'    '/
       DIMENSION AC(10),ISEAT(10),IFLTNO(150),IAC(150),ILF(150),
      1          IARRT(150),IDEPT(150),ITRANS(150),ICAT(150),IGATE(25),
      2          IGTIME(25,150),IWALK(25,6),ITWALK(25,25),IGT(25),
      3          IFA(25),IFD(25),IFT(25),IFWA(25),
      4          IAGATE(150),ISGATE(150),IFLTA(150),ICGATE(150),
      5          IFAA(25),IFDA(25),IFTA(25),IFWAA(25)
       DO 10 I=1,25
       IFA(I)=0
       IFD(I)=0
       IFT(I)=0
       IFWA(I)=0
    10 CONTINUE
C
C      INPUT AIRCRAFT DATA
       NAC=1
   100 READ(5,110) AC(NAC),ISEAT(NAC)
   110 FORMAT(A4,I4)
       IF(ISEAT(NAC).NE.777) GO TO 120
       NAC=NAC-1
       GO TO 200
   120 NAC=NAC+1
       GO TO 100
C
C      INPUT FLIGHT DATA
   200 NFLT=1
C FORMAT & READ REP
   205 READ(5,210) ISEQN,IFLTA(NFLT),IFLTNO(NFLT),ACTYPE,
      1            IARRT(NFLT),IDEPT(NFLT),ICAT(NFLT),IAGATE(NFLT)
   210 FORMAT(I4,I4,I3,A4,I5,I5,I2,I3)
C FOL CRD ADD
       ICAT(NFLT)=ICAT(NFLT)+1
       IF(IFLTNO(NFLT).NE.0) GO TO 215
       NFLT=NFLT-1
       GO TO 300
C      CHECK AIRCRAFT TYPE
   215 IAC(NFLT)=0
       DO 220 J=1,NAC
       IF(AC(J).EQ.ACTYPE) IAC(NFLT)=J
   220 CONTINUE
       IF(IAC(NFLT).NE.0) GO TO 240
       WRITE(6,230) IFLTA(NFLT),IFLTNO(NFLT)
   230 FORMAT(' INCORRECT AIRCRAFT TYPE CN FLIGHT NUMBER',I4,I3,
      1       'FLIGHT IGNORED')
       GO TO 205
   240 ILF(NFLT)=65
       ITRANS(NFLT)=30
```

```
      NFLT=NFLT+1
      GO TO 205
C
C      INPUT GATE DATA
  300 NGATE=1
C     ARRIVING AND DEPARTING DISTANCES
  310 READ(5,320) IGATE(NGATE), (IWALK(NGATE,J), J=1,6)
  320 FORMAT(I3,6I5)
      IF(IGATE(NGATE).NE.0) GO TO 330
      NGATE=NGATE-1
      GO TO 340
  330 NGATE=NGATE+1
      GO TO 310
C     DISTANCES BETWEEN GATES - TRANSFER WALKING DISTANCE
  340 DO 370 I=1,NGATE
      READ(5,350) (ITWALK(I,J),J=1,NGATE)
  350 FORMAT(20I4)
  370 CONTINUE
      DO 390 I=1,NGATE
      DO 360 J=1,NGATE
      ITWALK(J,I)=ITWALK(I,J)
  360 CONTINUE
      WRITE(6,351) (ITWALK(I,J),J=1,NGATE)
  351 FORMAT(1X,20I6)
  390 CONTINUE
C
C
C
      WRITE (6,394)
  394 FORMAT(////,20X,'LARGEST COME BEST SERVE')
      WRITE(6,457)
  457 FORMAT(////,
     1       1X, ' FLT   AC   ARR   DEP   GTE KTE    ARR    DEP  TRA
     2ACT  CAL   DIF     RAT     ACT  CAL  DIF      RAT     ACT  CAL
     3DIF  RAT')
C
C     INITIALIZE GATE AVAILABILITY
  400 DO 410 I=1,NGATE
      IGTIME(1,I)=0
  410 IGTIME(2,I)=-1
C
C
C
      DO 500 I=1,NFLT
      JG=0
      DO 213 K=1,NGATE
  213 IF(IAGATE(I).EQ.IGATE(K)) JG=K
      IF(JG.EQ.0) WRITE(6,272) I
  272 FORMAT(1X, ' INCORRECT GATE NUMBER FOR FLT IDX',I4)
      IF(JG.EQ.0) STOP
C     CALCULATE PASSENGER LOADS
      TRANS=ITRANS(I)/100.
      F=ILF(I)/100.
      IPA=ISEAT(IAC(I))*F*(1.0-TRANS)
      IPD=ISEAT(IAC(I))*F
```

```
      IPT=ISEAT(IAC(I))*F*TRANS
C     INITIALIZE GATE ASSIGNMENT
      MINDIS=1000000
      NEARBY=1
C     GATE ASSIGNMENT
      DO 420 J=1,NGATE
C     CHECK GATE AVAILABILITY
      IF (I.EQ.1) GO TO 416
      IP=I-1
      DO 411 L=1,IP
      IF (IGATE(J).NE.ISGATE(L)) GO TO 411
      IF (IARRT(I).GE.IARRT(L).AND.IARRT(I).LE.IDEPT(L)) GO TO 420
      IF (IDEPT(I).GE.IARRT(L).AND.IDEPT(I).LE.IDEPT(L)) GO TO 420
      IF (IARRT(I).LE.IARRT(L).AND.IDEPT(I).GE.IDEPT(L)) GO TO 420
  411 CONTINUE
C     COMPUTE AVERAGE WALKING DISTANCE FOR GATE J
416   IDA=IWALK(J,ICAT(I))
      IDD=IWALK(J,(ICAT(I)+3))
      IDT=0
      DO 412 K=1,NGATE
  412 IDT=IDT+ITWALK(J,K)/NGATE
      IPDA=IDA*IPA
      IPDD=IDD*IPD
      IPDT=IDT*IPT
      IDIST=(IPDA+IPDD+IPDT)/(IPA+IPD+IPT)
C     SELECT MINIMUM WALKING DISTANCE
      IF(IDIST.GT.MINDIS) GO TO 420
      NEARBY=J
      ISGATE(I)=IGATE(J)
      MINDIS=IDIST
  420 CONTINUE
C     CHECK TO SEE THAT A GATE HAS BEEN ASSIGNED TO THE FLIGHT
      IF(MINDIS.NE.1000000) GO TO 450
      WRITE(6,430) IFLTNO(I)
  430 FORMAT(' FLIGHT ',I4,'COULD NOT BE ASSIGNED TO ANY AVAILABLE ',
     1       'GATE. ARRIVAL DELAYED UNTIL FIRST AVAILABLE GATE.')
      NEARBY=1
      IWAIT=IGTIME(2,1)
      DO 440 J=2,NGATE
      IF(IGTIME(2,J).GT.IWAIT) GO TO 440
      NEARBY=J
      IWAIT=IGTIME(2,J)
  440 CONTINUE
  450 IGTIME(1,NEARBY)=IARRT(I)
      IGTIME(2,NEARBY)=IDEPT(I)
      ICGATE(I)=NEARBY
      IDA=IWALK(NEARBY,ICAT(I))
      IDAA=IWALK(JG,ICAT(I))
      IDD=IWALK(NEARBY,(ICAT(I)+3))
      IDDA=IWALK(JG,ICAT(I)+3)
      IDT=0
      IDTA=0
      DO 455 K=1,NGATE
      IDTA=IDTA+ITWALK(JG,K)/NGATE
  455 IDT=IDT+ITWALK(NEARBY,K)/NGATE
```

```
      JDEPT=IDEPT(I)
      IF(IDEPT(I).GT.2400) JDEPT=IDEPT(I)-2400
      IDIFA=IDAA-IDA
      IDIFD=IDDA-IDD
      IDIFT=IDTA-IDT
      RATA=FLOAT(IDA)/FLOAT(IDAA)
      RATD=FLOAT(IDD)/FLOAT(IDDA)
      RATT=FLOAT(IDT)/FLOAT(IDTA)
      WRITE(6,460) IFLTNO(I),AC(IAC(I)),IARRT(I),JDEPT,
     1              IAGATE(I),IGATE(NEARBY),
     2              IPA,IPD,IPT,
     3              IDAA,IDA,IDIFA,RATA,
     4              IDDA,IDD,IDIFD,RATD,
     5              IDTA,IDT,IDIFT,RATT
  460 FORMAT(/,1X,I4,1X,A4,2X,I4,1X,I4,2X,2I4,3X,3I5,4X,2I5,I6,
     11F8.3,3X,2I5,I6,1F8.3,3X,2I5,I6,1F8.3)
      K1=IDA/100
      K2=IDD/100
      K3=IDT/100
      NA=IDAA/100
      ND=IDDA/100
      NT=IDTA/100
      IFA(K1)=IFA(K1)+IPA
      IFD(K2)=IFD(K2)+IPD
      IFT(K3)=IFT(K3)+IPT
      IFAA(NA)=IFAA(NA)+IPA
      IFDA(ND)=IFDA(ND)+IPD
      IFTA(NT)=IFTA(NT)+IPT
      IWA=(IDA*IPA+IDD*IPD+IDT*IPT)/(IPA+IPD+IPT)
      IWAA=(IDAA*IPA+IDDA*IPD+IDTA*IPT)/(IPA+IPD+IPT)
      K4=IWA/100
      NWK=IWAA/100
      IFWAA(NWK)=IFWAA(NWK)+IPA+IPD+IPT
      IFWA(K4)=IFWA(K4)+IPA+IPD+IPT
  500 CONTINUE
      WRITE(6,510)
  510 FORMAT(/,' HISTOGRAM')
      DO 900 I=1,25
  900 WRITE(6,910) IFA(I),IFD(I),IFT(I),IFWA(I),IFAA(I),IFDA(I),IFTA(I),
     1             IFWAA(I)
  910 FORMAT(1X,8I10)
      STOP
      END
/*
//GO.SYSIN DD *
 DC9  90
 D9S 110
 DC8 140
 D8S 210
 727 135
 72S 135
 L10 262
 747 382
 777 777
  67 857857 747 1545 1645 2 87
```

FILE: ALGO   VS1JOB   A

76	136136	747	1625	1730	0	81
92	870870	747	1650	1750	2	83
109	149149	747	1800	1930	0	83
97	871871	747	1815	1910	2	85
123	856856	747	1945	2100	2	87
7	000608	L10	0000	0725	0	77
25	000243	L10	0000	0830	0	75
31	000105	L10	0000	0915	0	81
6	164164	L10	0710	0815	0	79
22	791791	L10	0825	0910	1	91
24	117117	L10	0830	0920	0	77
35	123123	L10	0940	1030	0	77
60	110624	L10	1410	1630	0	73
63	106247	L10	1445	1715	0	75
64	250141	L10	1445	1750	0	77
94	137137	L10	1810	1900	0	81
106	437165	L10	1910	2100	0	77
116	148148	L10	2010	2100	0	83
121	792792	L10	2025	2100	1	91
125	154154	L10	2110	2200	0	75
129	160160	L10	2120	2210	0	77
137	621621	L10	2220	2310	0	81
143	248248	L10	2320	2400	0	73
2	000310	D8S	0000	0700	0	76
17	000920	D8S	0000	0800	1	91
19	960960	D8S	0805	0900	2	87
26	603992	D8S	0850	1100	2	85
50	122249	D8S	1240	1420	0	77
67	813813	D8S	1520	1625	2	83
77	790790	D8S	1625	1725	1	91
85	921872	D8S	1700	1900	1	87
90	891891	D8S	1745	1840	2	99
89	873161	D8S	1745	1945	2	79
110	878878	D8S	1820	1930	2	89
102	793793	D8S	1840	1930	0	91
111	881881	D8S	1940	2120	2	85
113	807807	D8S	1955	2100	2	99
118	244244	D8S	2015	2100	0	81
126	993993	D8S	2110	2215	2	91
1	000440	DC8	0000	0655	0	79
43	902902	DC8	0930	1030	1	91
75	147147	DC8	1620	1700	0	74
74	961961	DC8	1615	1715	2	85
82	903903	DC8	1655	1800	1	89
141	156156	DC8	2310	2400	0	77
4	000400	727	0000	0700	0	80
16	000402	727	0000	0800	0	78
12	441796	72S	0755	0905	2	95
18	401404	727	0805	0900	0	80
23	103103	72S	0830	0915	0	71
30	403406	727	0905	1000	0	78
38	246246	72S	1005	1045	0	81
39	405408	727	1005	1100	0	80
44	407410	727	1105	1200	0	78
47	409412	727	1205	1300	0	80

FILE: ALGO VS1JOB A

51	724725	72S	1300	1410	1	91
53	411414	727	1305	1400	0	78
55	465454	72S	1310	1420	0	81
59	413416	727	1405	1500	0	80
66	415418	727	1505	1600	0	78
73	417420	727	1605	1700	0	80
79	455460	72S	1645	1745	0	76
86	419422	727	1705	1800	0	78
91	726729	72S	1750	1855	1	93
95	421424	727	1810	1900	0	80
100	797797	72S	1835	1925	2	97
105	423426	727	1910	2000	0	78
115	425428	727	2005	2100	0	80
117	461464	72S	2010	2110	0	79
124	427427	727	2105	2155	0	78
127	162162	72S	2115	2155	0	81
138	241241	72S	2130	2240	0	79
136	429429	727	2205	2300	0	80
5	000701	D9S	0000	0700	1	87
9	000721	D9S	0000	0730	1	93
13	000341	D9S	0000	0755	0	73
29	000982	D9S	0000	0900	2	89
36	720705	D9S	0445	1050	1	89
8	612612	D9S	0730	0800	0	76
10	238107	D9S	0740	0930	0	72
14	700774	D9S	0800	0855	1	93
20	308308	D9S	0815	0845	0	73
21	362444	D9S	0815	0900	0	76
27	346365	D9S	0855	0945	0	75
32	342642	D9S	0930	1035	0	74
33	605600	D9S	0930	1050	0	76
41	625654	D9S	1025	1115	0	83
42	773778	D9S	1030	1230	1	93
45	704385	D9S	1140	1545	1	76
48	368315	D9S	1230	1320	0	79
49	102102	D9S	1240	1310	0	83
52	344349	D9S	1305	1405	0	74
54	777780	D9S	1305	1500	1	93
56	706709	D9S	1320	1410	1	89
65	647650	D9S	1450	1550	0	71
68	646646	D9S	1530	1605	0	74
70	351351	D9S	1555	1625	0	76
71	601658	D9S	1555	1650	0	79
72	779713	D9S	1600	1725	1	93
78	609446	D9S	1640	1750	0	72
80	710727	D9S	1645	1745	1	95
83	983784	D9S	1655	1805	2	97
81	649387	D9S	1655	1815	0	71
88	655655	D9S	1730	1800	0	75
96	604604	D9S	1810	1855	0	77
98	489233	D9S	1820	1925	0	72
99	382389	D9S	1830	1920	0	73
101	353353	D9S	1840	1915	0	74
103	163163	D9S	1845	1920	0	71
104	781786	D9S	1905	1955	1	95

FILE: ALGO VS1JOB A

112	653329	D9S	1955	2055	0	72	
114	716719	D9S	2000	2120	1	93	
119	330357	D9S	2015	2120	0	71	
120	354331	D9S	2015	2120	0	73	
122	394355	D9S	2035	2130	0	76	
128	152333	D9S	2120	2225	0	74	
131	728309	D9S	2135	2315	1	85	
132	356356	D9S	2155	2240	0	76	
133	783788	D9S	2155	2255	1	93	
135	396397	D9S	2205	2255	0	73	
140	334334	D9S	2305	2350	0	79	
142	332332	D9S	2310	2400	0	74	
144	467467	D9S	2320	2400	0	75	
145	789789	D9S	2345	2400	1	89	
3	000361	DC9	0000	0700	0	74	
11	000442	DC9	0000	0740	0	85	
15	000303	DC9	0000	0800	0	71	
34	450450	DC9	0830	0935	0	79	
28	360363	DC9	0900	0950	0	73	
37	541373	DC9	0950	1215	0	71	
40	312371	DC9	1010	1115	0	72	
46	366347	DC9	1205	1315	0	75	
57	370317	DC9	1330	1530	0	72	
58	481522	DC9	1340	1450	0	83	
61	374383	DC9	1430	1530	0	79	
62	485526	DC9	1440	1605	0	81	
84	348327	DC9	1700	1745	0	73	
87	542542	DC9	1715	1800	0	74	
93	384486	DC9	1805	1940	0	76	
107	324324	DC9	1920	1950	0	75	
108	463391	DC9	1930	2100	0	74	
130	535535	DC9	2120	2215	0	72	
134	469469	DC9	2200	2245	0	83	
139	398398	DC9	2300	2400	0	76	

71	1287	2367	1727	1303	2261	1737
72	1269	2350	1710	1285	2244	1720
73	1285	2365	1725	1301	2259	1735
74	1106	2193	1553	1112	2087	1543
75	1102	2182	1542	1118	2076	1552
76	926	2013	1373	932	1907	1363
77	919	1929	1289	935	1823	1299
78	746	1833	1193	752	1727	1183
79	739	1749	1109	755	1643	1119
80	566	1670	1030	582	1564	1020
81	556	1566	926	572	1460	936
83	509	1343	703	349	1237	713
85	594	1068	428	434	962	438
87	855	807	347	695	701	177
89	1109	553	601	949	447	329
91	1363	299	855	1203	193	583
93	1662	598	1154	1502	492	882
95	1845	781	1337	1685	675	1065
97	510	418	828	350	312	668
99	957	418	828	797	312	568

```
FILE: ALGO     VS1JOB   A

  10  20 310 270 530 420 720 610 910 8001040128015601830210023702640291031 80
      30 300 230 540 430 730 620 920 81010501290157018402110238026502920 3190
         310 200 510 400 700 590 890 78010201260154018102080235030703340 3610
             110 200 330 220 500 330 690 9301170145017201990266025302800 3070
                 110 200 500 390 690 580 82010601340161018802150242026902960
                     110 190 300 220 490 730 97012501520179020602330260028 70
                         110 190 490 380 620 86011401410168019502220249027 60
                             110 190 300 540 7801060133016001870214024102680
                                 300 190 430 670 950122014901760203023002570
                                     110 350 590 870114014101680195022202490
                                         240 480 76010301300157018402110 2380
                                             240 520 7901060133016001870 2140
                                                 280 550 820109013601630 1900
                                                     270 540 81010801350 1620
                                                         270 540 8101080 1350
                                                             270 540 810 1080
                                                                 270 540  810
                                                                     270  540
                                                                          270
                                                                            0

/*
/*EOJ ********
```

APPENDIX B

RESULTS OF THE "CROWDEST-COME-BEST-SERVE" ALGORITHM

This appendix contains the output to the computer program of Appendix A. The content of each column in the output is as follows:

	Heading	Content
First Column	FLT	Flight number
Second "	AC	Aircraft type
Third "	ARR	Flight's arrival time
Fourth "	DEP	Flight's departure time
Fifth "	GTE	Original gate assignment
Sixth "	KTE	Algorithm's gate assignment
Seventh "	ACT	Walking distance under original assignment for arriving passengers.
Eighth "	CAL	Walking distance under algorithm's assignment for departing passengers
Ninth "	PIF	Difference in the walking distances listed in the two previous columns
Tenth "	RAT	Ratio of the algorithm's walking distance to the original walking distance for the arriving passengers.

	Content
Eleventh through fourteenth columns	Same as 7th through 10th columns, but for departing passengers
Fifteenth through nineteenth column	Same as 7th through 10th columns, but for transfer passengers

FLT	AC	ARR	DEP	GTE	KTE	ARR	DEP	TRA	ACT	CAL	DIF	RAT	ACT	CAL	DIF	RAT	ACT	CAL	DIF	RAT
857	747	1545	1645	87	87	173	248	74	347	347	0	1.000	177	177	0	1.000	994	994	0	1.000
136	747	1625	1730	81	83	173	248	74	556	509	47	0.915	572	349	223	0.610	838	862	-24	1.029
870	747	1650	1750	83	87	173	248	74	703	347	356	0.494	713	177	536	0.248	862	994	-132	1.153
14[illegible]	747	1800	1910	83	83	173	248	74	509	509	0	1.000	349	349	0	1.000	862	862	0	1.000
871	747	1815	1910	85	87	173	248	74	428	347	81	0.811	438	177	261	0.404	910	994	-84	1.092
856	747	1945	2100	87	87	173	248	74	347	347	0	1.000	177	177	0	1.000	994	994	0	1.000
634	L10	0	725	77	83	119	170	51	919	509	410	0.554	935	349	585	0.373	905	862	43	0.952
243	L10	0	830	75	85	119	170	51	1102	594	509	0.539	1118	434	684	0.388	1006	910	96	0.905
105	L10	0	915	81	81	119	170	51	556	556	0	1.000	572	572	0	1.000	838	838	0	1.000
154	L10	710	815	79	80	119	170	51	739	566	173	0.766	755	582	173	0.771	855	881	-26	1.030
791	L10	825	910	91	91	119	170	51	299	299	0	1.000	193	193	0	1.000	1237	1237	0	1.000
117	L10	830	920	77	83	119	170	51	919	509	410	0.554	935	349	586	0.373	905	862	43	0.952
123	L10	940	1130	77	83	119	170	51	919	509	410	0.554	935	349	586	0.373	905	862	43	0.952
624	L10	1410	1630	73	85	119	170	51	1285	594	691	0.462	1301	434	867	0.334	1221	910	311	0.745
247	L10	1445	1715	75	81	119	170	51	1102	556	546	0.505	1118	572	546	0.512	1006	838	158	0.833
141	L10	1445	1750	77	80	119	170	51	919	566	353	0.616	935	582	353	0.622	905	881	24	0.973
137	L10	1810	1900	81	85	119	170	51	556	594	-38	1.068	572	434	138	0.759	839	910	-72	1.086
165	L10	1910	2100	77	85	119	170	51	919	594	325	0.646	935	434	501	0.464	905	910	-5	1.006
148	L10	2010	2100	83	83	119	170	51	509	509	0	1.000	349	349	0	1.000	862	862	0	1.000
792	L10	2025	2100	91	91	119	170	51	299	299	0	1.000	193	193	0	1.000	1237	1237	0	1.000
1[illegible]4	L10	2110	2200	75	83	119	170	51	1102	509	593	0.462	1118	349	769	0.312	1006	862	144	0.857
1[illegible]0	L10	2120	2210	77	85	119	170	51	919	594	325	0.646	935	434	501	0.464	905	910	-5	1.006
621	L10	2220	2310	81	83	119	170	51	556	509	47	0.915	572	349	223	0.610	838	862	-24	1.029
248	L10	2320	2400	73	83	119	170	51	1285	509	776	0.396	1301	349	952	0.268	1221	862	359	0.706
310	D8S	0	700	76	80	95	136	40	926	566	360	0.611	932	582	350	0.624	960	881	79	0.918
920	D8S	0	800	91	91	95	136	40	299	299	0	1.000	193	193	0	1.000	1237	1237	0	1.000

960	DBS	805	900	87	87	95	136	40	347	347	0	1.000	177	177	0	1.000	994	994	0	1.000
992	DAS	850	1100	85	85	95	136	40	428	428	0	1.000	438	438	0	1.000	910	910	0	1.000
249	DAS	1240	1420	77	83	95	136	40	919	509	410	0.554	935	349	586	0.373	905	862	43	0.952
813	DAS	1520	1625	83	89	95	136	40	703	601	102	0.855	713	329	384	0.461	962	1100	-239	1.276
790	DAS	1625	1725	91	91	95	136	40	299	299	0	1.000	193	193	0	1.000	1237	1237	0	1.000
872	DAS	1700	1900	87	97	95	136	40	807	418	389	0.518	701	312	389	0.445	994	1925	-331	1.836
878	DAS	1820	1930	89	89	95	136	40	601	601	0	1.000	329	329	0	1.000	1100	1100	0	1.000
793	DAS	1840	1930	91	81	95	136	40	1363	556	807	0.408	1203	572	631	0.475	1237	838	399	0.677
881	DAS	1940	2120	85	89	95	136	40	428	601	-173	1.404	438	329	109	0.751	910	1100	-190	1.209
807	DAS	1955	2100	99	99	95	136	40	828	828	0	1.000	568	568	0	1.000	2069	2069	0	1.000
244	DAS	2015	2100	81	81	95	136	40	556	556	0	1.000	572	572	0	1.000	838	838	0	1.000
993	DAS	2110	2215	91	87	95	136	40	855	347	508	0.406	583	177	406	0.304	1237	994	243	0.804
443	DCB	0	655	79	97	63	90	27	739	510	229	0.690	755	350	405	0.464	855	1825	-970	2.135
992	DCB	930	1030	91	91	63	90	27	299	299	0	1.000	193	193	0	1.000	1237	1237	0	1.000
961	DCB	1615	1715	85	99	63	90	27	428	828	-400	1.935	438	563	-130	1.297	910	2069	-1159	2.274
147	DCB	1620	1700	74	79	63	90	27	1106	739	367	0.668	1112	755	357	0.679	1075	855	220	0.795
993	DCB	1655	1800	89	85	63	90	27	553	553	0	1.000	447	447	0	1.000	1100	1100	0	1.000
156	DCB	2110	2400	77	85	63	90	27	919	594	325	0.646	935	434	501	0.464	905	910	-5	1.006
403	727	0	700	80	79	61	87	26	566	739	-173	1.306	582	755	-173	1.297	881	855	26	0.970
402	727	0	800	78	78	61	87	26	746	746	0	1.000	752	752	0	1.000	905	905	0	1.000
796	72S	755	905	95	89	61	87	26	1337	601	736	0.450	1065	329	736	0.309	1610	1100	510	0.683
404	727	805	900	80	97	61	87	26	566	510	56	0.901	582	350	232	0.601	881	1825	-944	2.072
133	72S	830	915	71	80	61	87	26	1287	566	721	0.440	1303	582	721	0.447	1171	881	290	0.752
406	727	905	1000	78	97	61	87	26	746	510	236	0.684	752	350	402	0.465	905	1825	-920	2.017
246	72S	1005	1045	81	81	61	87	26	556	556	0	1.000	572	572	0	1.000	838	838	0	1.000
408	727	1005	1100	80	80	61	87	26	566	566	0	1.000	582	582	0	1.000	881	881	0	1.000
410	727	1105	1200	78	83	61	87	26	746	509	237	0.682	752	349	403	0.464	935	862	43	0.952
412	727	1205	1300	80	85	61	87	26	566	594	-28	1.049	582	434	148	0.746	881	910	-29	1.033
725	72S	1303	1410	91	91	61	87	26	299	299	0	1.000	193	193	0	1.000	1237	1237	0	1.000
414	727	1305	1400	78	85	61	87	26	746	594	152	0.796	752	434	318	0.577	905	910	-5	1.006
454	72S	1310	1420	81	81	61	87	26	556	556	0	1.000	572	572	0	1.000	838	838	0	1.000
416	727	1405	1500	80	97	61	87	26	566	510	56	0.901	582	350	232	0.601	881	1825	-944	2.072
418	727	1505	1600	78	83	61	87	26	746	509	237	0.682	752	349	403	0.464	905	862	43	0.952

420	727	1605	1700	80	78	61	87	26	566	746	-180	1.318	582	752	-170	1.292	881	905	-24	1.027
460	725	1645	1745	76	85	61	87	26	926	594	332	0.641	932	434	498	0.466	960	910	50	0.949
422	727	1705	1800	78	79	61	87	26	746	739	7	0.991	752	755	-3	1.004	905	855	50	0.945
729	725	1750	1855	93	91	61	87	26	598	299	299	0.500	492	193	299	0.392	1417	1237	180	0.873
424	727	1810	1900	80	80	61	87	26	566	566	0	1.000	582	582	0	1.000	881	881	0	1.000
797	725	1835	1925	97	99	61	87	26	828	828	0	1.000	668	568	100	0.850	1825	2059	-234	1.134
426	727	1910	2000	78	80	61	87	26	746	566	180	0.759	752	582	170	0.774	905	881	24	0.973
428	727	2035	2100	80	80	61	87	26	566	566	0	1.000	582	582	0	1.000	881	881	0	1.000
464	725	2010	2110	79	97	61	87	26	739	510	229	0.690	755	350	405	0.464	855	1825	-970	2.135
427	727	2105	2155	78	81	61	87	26	746	556	190	0.745	752	572	180	0.761	905	838	67	0.926
162	725	2115	2155	81	80	61	87	26	556	566	-10	1.018	572	582	-10	1.017	838	881	-43	1.051
241	725	2130	2240	79	97	61	87	26	739	510	229	0.690	755	350	405	0.464	855	1825	-970	2.135
429	727	2205	2300	80	81	61	87	26	566	556	10	0.982	582	572	10	0.983	881	838	43	0.951
711	D9S	0	700	87	89	50	71	21	807	553	254	0.685	701	447	254	0.638	994	1100	-106	1.107
721	D9S	0	730	93	99	50	71	21	598	418	180	0.699	492	312	180	0.634	1417	2069	-652	1.460
341	D9S	0	755	73	87	50	71	21	1285	855	430	0.665	1301	695	606	0.534	1221	994	227	0.814
982	D9S	0	900	89	93	50	71	21	601	1158	-553	1.920	329	882	-553	2.681	1100	1417	-317	1.288
795	D9S	445	1050	89	95	50	71	21	553	781	-228	1.412	447	675	-228	1.510	1100	1610	-510	1.464
612	D9S	710	800	76	83	50	71	21	926	509	417	0.550	932	349	583	0.374	960	862	98	0.898
107	D9S	740	930	72	79	50	71	21	1269	739	530	0.582	1285	755	530	0.588	1175	855	320	0.728
774	D9S	800	855	93	99	50	71	21	598	418	180	0.699	492	312	180	0.634	1417	2059	-652	1.460
308	D9S	815	845	73	78	50	71	21	1285	746	539	0.581	1301	752	549	0.578	1221	905	316	0.741
444	D9S	815	900	76	77	50	71	21	926	919	7	0.992	932	935	-3	1.003	960	905	55	0.943
365	D9S	855	945	75	78	50	71	21	1102	746	356	0.677	1118	752	366	0.673	1006	905	101	0.900
642	D9S	930	1035	74	87	50	71	21	1106	855	251	0.773	1112	695	417	0.625	1075	994	81	0.925
600	D9S	930	1050	76	77	50	71	21	926	919	7	0.992	932	935	-3	1.003	960	905	55	0.943
654	D9S	1025	1115	83	97	50	71	21	509	510	-1	1.002	349	350	-1	1.003	862	1825	-963	2.117
778	D9S	1030	1230	93	89	50	71	21	598	553	45	0.925	492	447	45	0.909	1417	1100	317	0.776
385	D9S	1140	1545	76	99	50	71	21	2013	418	1595	0.208	1907	312	1595	0.164	960	2069	-1109	2.155
315	D9S	1230	1320	79	80	50	71	21	739	566	173	0.766	755	582	173	0.771	855	881	-26	1.030
102	D9S	1240	1310	83	97	50	71	21	509	510	-1	1.002	349	350	-1	1.003	862	1825	-963	2.117
349	D9S	1305	1405	74	79	50	71	21	1106	739	367	0.668	1112	755	357	0.679	1075	855	220	0.795
780	D9S	1305	1500	93	89	50	71	21	598	553	45	0.925	492	447	45	0.909	1417	1100	317	0.776

709	D9S	1320	1410	89	93	50	71	21	553	598	-45	1.081	447	492	-45	1.101	1100	1417	-317	1.288
650	D9S	1450	1550	71	79	50	71	21	1287	739	548	0.574	1303	755	548	0.579	1171	855	316	0.730
646	D9S	1530	1605	74	97	50	71	21	1106	510	596	0.461	1112	350	762	0.315	1075	1825	-750	1.698
351	D9S	1555	1625	76	77	50	71	21	926	919	7	0.992	932	935	-3	1.003	960	905	55	0.943
659	D9S	1555	1650	79	76	50	71	21	739	926	-187	1.253	755	932	-177	1.234	855	960	-105	1.123
713	D9S	1600	1725	93	93	50	71	21	598	598	0	1.000	492	492	0	1.000	1417	1417	0	1.000
446	D9S	1640	1750	72	77	50	71	21	1269	919	350	0.724	1285	935	350	0.728	1175	905	270	0.770
727	D9S	1645	1745	95	95	50	71	21	781	781	0	1.000	675	675	0	1.000	1610	1610	0	1.000
784	D9S	1655	1805	97	76	50	71	21	828	1373	-545	1.658	668	1363	-695	2.040	1325	960	355	0.526
387	D9S	1655	1815	71	75	50	71	21	1287	1102	185	0.856	1303	1118	185	0.858	1171	1006	165	0.859
655	D9S	1730	1800	75	81	50	71	21	1102	556	546	0.505	1118	572	546	0.512	1006	838	168	0.833
604	D9S	1810	1555	77	79	50	71	21	919	739	180	0.804	935	755	180	0.807	905	855	50	0.945
233	D9S	1820	1925	72	78	50	71	21	1269	746	523	0.588	1285	752	533	0.585	1175	905	270	0.770
383	D9S	1830	1920	73	77	50	71	21	1285	919	366	0.715	1301	935	365	0.719	1221	905	316	0.741
353	D9S	1840	1915	74	76	50	71	21	1106	926	180	0.837	1112	932	180	0.838	1075	960	115	0.893
153	D9S	1845	1920	71	75	50	71	21	1287	1102	185	0.856	1303	1118	185	0.858	1171	1006	165	0.859
786	D9S	1905	1955	95	91	50	71	21	781	299	482	0.383	675	193	482	0.286	1610	1237	373	0.768
329	D9S	1955	2055	72	79	50	71	21	1269	739	530	0.582	1285	755	530	0.588	1175	855	320	0.728
719	D9S	2000	2120	93	93	50	71	21	598	598	0	1.000	492	492	0	1.000	1417	1417	0	1.000
357	D9S	2015	2120	71	78	50	71	21	1287	746	541	0.580	1303	752	551	0.577	1171	905	266	0.773
311	D9S	2015	2120	73	77	50	71	21	1285	919	366	0.715	1301	935	366	0.719	1221	905	316	0.741
355	D9S	2035	2130	76	76	50	71	21	926	926	0	1.000	932	932	0	1.000	960	960	0	1.000
333	D9S	2120	2225	74	79	50	71	21	1106	739	367	0.668	1112	755	357	0.679	1075	855	220	0.795
309	D9S	2135	2315	85	91	50	71	21	1068	299	769	0.280	962	193	769	0.201	910	1237	-327	1.359
356	D9S	2155	2240	76	78	50	71	21	926	746	180	0.806	932	752	180	0.807	960	905	55	0.943
788	D9S	2155	2255	93	89	50	71	21	598	553	45	0.925	492	447	45	0.909	1417	1100	317	0.776
297	D9S	2205	2255	73	80	50	71	21	1285	566	719	0.440	1301	582	719	0.447	1221	881	340	0.722
334	D9S	2305	2350	79	81	50	71	21	739	556	183	0.752	755	572	183	0.758	855	838	17	0.980
332	D9S	2310	2400	74	80	50	71	21	1106	566	540	0.512	1112	582	530	0.523	1075	881	194	0.820
467	D9S	2320	2400	75	97	50	71	21	1102	510	592	0.463	1118	350	768	0.313	1006	1825	-819	1.814
789	D9S	2345	2400	89	91	50	71	21	553	299	254	0.541	447	193	254	0.432	1100	1237	-137	1.125
361	DC9	0	700	74	77	40	58	17	1106	919	187	0.831	1112	935	177	0.841	1075	905	170	0.842
442	DC9	0	740	85	76	40	58	17	594	926	-332	1.559	434	932	-498	2.147	910	960	-50	1.055

303	DC9	0	800	71	75	40	58	.17	1287	1102	185	0.856	1303	1118	185	0.858	1171	1006	165	0.859
450	DC9	830	935	79	76	40	58	17	739	926	-187	1.253	755	932	-177	1.234	855	960	-105	1.123
363	DC9	900	950	73	99	40	58	17	1285	957	328	0.745	1301	797	504	0.613	1221	2069	-848	1.695
373	DC9	950	1215	71	79	40	58	17	1287	739	548	0.574	1303	755	548	0.579	1171	855	316	0.730
371	DC9	1010	1115	72	78	40	58	17	1269	746	523	0.588	1285	752	533	0.585	1175	905	270	0.770
347	DC9	1235	1315	75	78	40	58	17	1102	746	356	0.677	1118	752	365	0.673	1036	905	131	0.900
317	DC9	1330	1530	72	78	40	58	17	1269	746	523	0.588	1285	752	533	0.585	1175	905	270	0.770
522	DC9	1340	1450	83	87	40	58	17	509	855	-346	1.680	349	695	-335	1.991	862	998	-132	1.153
383	DC9	1430	1530	79	77	40	58	17	739	919	-180	1.244	755	935	-180	1.238	855	905	-50	1.058
526	DC9	1440	1605	81	75	40	58	17	556	1102	-546	1.982	572	1118	-545	1.955	838	1006	-168	1.200
327	DC9	1700	1745	73	74	40	58	17	1285	1106	179	0.861	1301	1112	189	0.855	1221	1075	146	0.880
542	DC9	1715	1800	74	78	40	58	17	1106	746	360	0.675	1112	752	360	0.676	1075	905	170	0.842
406	DC9	1805	1940	76	74	40	58	17	926	1106	-180	1.194	932	1112	-180	1.193	960	1075	-115	1.120
324	DC9	1720	1950	75	57	40	58	17	1102	510	592	0.463	1118	350	769	0.313	1036	1825	-819	1.814
391	DC9	1930	2100	74	75	40	58	17	1106	1102	4	0.996	1112	1118	-6	1.005	1075	1006	69	0.936
535	DC9	2120	2215	72	99	40	58	17	1269	957	312	0.754	1285	797	488	0.620	1175	2059	-334	1.761
469	DC9	2200	2245	83	77	40	58	17	509	919	-410	1.806	349	935	-586	2.679	862	905	-43	1.050
398	DC9	2300	2400	76	79	40	58	17	926	739	187	0.798	932	755	177	0.810	960	855	105	0.891

APPENDIX C

PREPROCESSOR OR MODEL

GENERATING PROGRAM

(Written in DATAMAT)

```
NAME          PLANES
*
* TABLES:
*
* G:PLANES
*          NAME OF FLIGHT AS 'AAAA:B', 'B' A CODE FOR TYPE
*               (ONE COLUMN IN TABLE FOR EACH FLIGHT)
*    ARRIVAL   TIME AS HH.MM
*    DEPARTUR  TIME AS HH.MM
*    CAPACITY  NUMBER ON PLANE
*
* M:TYPENAME
*          CODE FOR TYPE - 'B' FROM FLIGHT NAME
*               (ONE COLUMN FOR EACH GATE TYPE)
*    TABLNAME  'TABLE_NAME' FOR (ONE OF) FOLLOWING TABLE(S)
*    DISTNAME  'DISTANCE_NAME' FOR A ROW IN NAMED GATE TABLE
*
* G:'TABLE_NAME'
*          NAME FOR GATE AS 'ZZ'
*               (ONE COLUMN FOR EACH GATE IN THE TYPE)
*    ROW(S)    WALKING DISTANCE TO GATE
*    'DISTANCE_NAME'
*
* M:GATETABL
*          (STUB TABLE)
*    'TABLE_NAME' FOR GATE TABLE(S)
*          TABLE(S) MUST PARTITIION GATES
*
* TABLES TO KEEP MAXIMAL CONFLICT SETS
*
     FORM M:MINDEPRT = 'HEAD',M:GATETABL(STUB)
     TABLE M:SETSTUB
          ORDER
          MAXORDER
...
     FORM G:SETCOUNT = M:SETSTUB(STUB),M:GATETABL(STUB)
     FORM M:MINCHAIN = 'NEXT',G:PLANES(HEAD)
     M:MINDEPRT(HEAD,!1) = 'VOID'
     G:SETCOUNT(!2,!1) = 0
     M:MINCHAIN(NEXT,!1) = 'NOTCHAIN'
*
* PROCESS FLIGHTS IN ORDER OF ARRIVAL
*
     NEWMODEL
//NXTLUP
     N:NEXT = DUMMY
     E:NEXT = 1E20
     LOOP M:MINCHAIN(0,!1) <NE> DUMMY
          IF M:MINCHAIN(NEXT,!1) <EQ> 'NOTCHAIN',1
GOTO ENDNXT
               IF G:PLANES(ARRIVAL,!1) <LT> E:NEXT,1
GOTO ENDNXT
                    E:NEXT = G:PLANES(ARRIVAL,!1)
                    N:NEXT = G:PLANES(0,!1)
//ENDNXT
```

```
          CONTINUE
     IF N:NEXT <NE> DUMMY,1
GOTO ENDLUP
*
* NAME OF FLIGHT, GATE TYPE CODE, TABLE OF GATES FOR TYPE
*
     N:PLANE = MASK (G:PLANES(0,N:NEXT),'******00')
     N:TYPE = SHIFT (MASK (N:PLANE,'00000*00'),5)
     N:GATETABL = M:TYPENAME(TABLNAME,N:TYPE)
     N:DISTNAME = M:TYPENAME(DISTNAME,N:TYPE)
*
* SELECTION CONSTRAINT FOR FLIGHT, WALKING DISTANCES
*
     ROW N:PLANE <EQTYPE>, N:PLANE &  G:N:GATETABL(0,!1) = 1.
     RHS UNITY, N:PLANE = 1
     ROW WALKDIST, N:PLANE & G:N:GATETABL(0,!1) =
               G:PLANES(CAPACITY,N:NEXT) * G:N:GATETABL(N:DISTNAME,!1)
*
* DETERMINE MEMBERSHIP OF NEXT FLIGHT IN CURRENT CONFLICT SET
*
     N:MIN = M:MINDEPRT(HEAD,N:GATETABL)
     IF N:MIN <NE> 'VOID',1
GOTO ADDNXT
          IF G:PLANES(ARRIVAL,N:NEXT) <GT> G:PLANES(DEPARTUR,N:MIN),1
GOTO ADDNXT
          IF G:SETCOUNT(ORDER,N:GATETABL) <GT> 1,1
GOTO DELETE
*
* MUST WRITE CONSTRAINT FOR CURRENT SET
*         THEN DELETE FLIGHTS NOT CONFLICTING WITH NEXT
*
     N:CONFLICT = MASK (G:PLANES(0,N:MIN),'****0000') & '::'
     ROW N:CONFLICT & G:N:GATETABL(0,!1) <LETYPE>
     RHS UNITY, N:CONFLICT & G:N:GATETABL(0,!1) = 1.
     N:INDEX = N:MIN
//DOCNST
     IF N:INDEX <NE> 'VOID',1
GOTO NDCNST
       COL MASK (G:PLANES(0,N:INDEX),'******00') & G:N:GATETABL(0,!1),
            N:CONFLICT & G:N:GATETABL(0,!1) = 1.
       N:INDEX = M:MINCHAIN(NEXT,N:INDEX)
GOTO DOCNST
//NDCNST
*
* DELETE NON-CONFLICTING FLIGHTS FROM CHAIN
*
//DELETE
     N:INDEX = N:MIN
//DODEL
     IF G:PLANES(DEPARTUR,N:INDEX) <LT> G:PLANES(ARRIVAL,N:NEXT),1
GOTO ENDEL
          N:INDEX = M:MINCHAIN(NEXT,N:INDEX)
G:SETCOUNT(ORDER,N:GATETABL) = G:SETCOUNT(ORDER,N:GATETABL) - 1
          IF N:INDEX <EQ> 'VOID',1
GOTO DODEL
```

```
               N:MIN = 'VOID'
GOTO ADDNXT
//ENDEL
     N:MIN = N:INDEX
     M:MINDEPRT(HEAD,N:GATETABL) = N:MIN
*
* ADD NEXT TO CHAIN FOR ITS TYPE, DEPARTURE-ORDERED
*
//ADDNXT
     N:INDEX = N:MIN
//DOCHAN
     IF N:INDEX <NE> 'VOID',1
GOTO RCHAN
       IF G:PLANES(DEPARTUR,N:INDEX) <LT> G:PLANES(DEPARTUR,N:NEXT),1
GOTO RCHAN
             N:LAST = N:INDEX
             N:INDEX = M:MINCHAIN(NEXT,N:INDEX)
GOTO DOCHAN
//RCHAN
     IF N:INDEX <NE> N:MIN,2
          M:MINDEPRT(HEAD,N:GATETABL) = N:NEXT
GOTO RCHAND
          M:MINCHAIN(NEXT,N:LAST) = N:NEXT
//RCHAND
     M:MINCHAIN(NEXT,N:NEXT) = N:INDEX
     G:SETCOUNT(ORDER,N:GATETABL) = G:SETCOUNT(ORDER,N:GATETABL) + 1
IF G:SETCOUNT(MAXORDER,N:GATETABL) <GT> G:SETCOUNT(ORDER,N:GATETABL),1
     G:SETCOUNT(MAXORDER,N:GATETABL) = G:SETCOUNT(ORDER,N:GATETABL)
GOTO NXTLUP
//ENDLUP
*
* WRITE CONSTRAINTS FOR FINAL CONFLICT SETS
*
     LOOP M:MINDEPRT(0,!1) <NE> DUMMY
          N:GATETABL = M:MINDEPRT(0,!1)
          I:MAXORDER = G:SETCOUNT(MAXORDER,!1)
          DISPLAY N:GATETABL,I:MAXORDER
          N:MIN = M:MINDEPRT(HEAD,!1)
          IF G:SETCOUNT(ORDER,N:GATETABL) <GT> 1,1
GOTO ENDCLR
     N:CONFLICT = MASK (G:PLANES(0,N:MIN),'****0000') & '::'
     ROW N:CONFLICT & G:N:GATETABL(0,!2) <LETYPE>
     RHS UNITY, N:CONFLICT & G:N:GATETABL(0,!2) = 1.
     N:INDEX = N:MIN
//DCONST
     IF N:INDEX <NE> 'VOID',1
GOTO NDCON
       COL MASK (G:PLANES(0,N:INDEX),'******00') & G:N:GATETABL(0,!2),
            N:CONFLICT & G:N:GATETABL(0,!2) = 1.
       N:INDEX = M:MINCHAIN(NEXT,N:INDEX)
GOTO DCONST
//NDCON
//ENDCLR
          CONTINUE
QUIT
```

ENDATA

APPENDIX D

THE POSTPROCESSOR PROGRAM

(Written in DATAMAT)

Listing of functions in the Postprocessor:

Name	Purpose
FINAL	Constructs a condensed table containing all flights and their LP assigned gate
MEAN	Constructs a table containing, for each flight, the gate assignment and corresponding passenger mean walking distance under each of the three policies: 1) the original airport assingment 2) the heuristic algorithm and 3) the LP
ARRIVALS	Same as MEAN, but instead of listing the overall mean walking distance, it lists the expected walking distance for the arriving passengers.
DEPARTUR	Same as ARRIVALS, but for the departing passengers.
TRANSFER	Same as ARRIVALS, but for the transfer passengers.
HISTO	Produces a statistical distribution for the distances listed in the table produced by MEAN. In other words, it lists a histogram of the overall mean walking distance.

Name	Purpose
ARRHISTO	Same as HISTO, but the histogram is for distances of arriving passengers only.
DEPHISTO	Same as HISTO, but for the departing passenger.
TRFHISTO	Same as HISTO, but for the transfer passenger .

```
*TABLES NEEDED FOR MACROS IN THIS FILE:
*G:PLANES, A LIST OF FLIGHTS, THEIR ARRIVAL AND DEPARTURE TIME,...ETC.
*G:ALGOTES,WHICH CONTAINS RESULTS OF THE ALGORITHM AS WELL AS DATA
*           CONCERNED WITH THE ORIGINAL ASSIGNMENT GIVEN BY THE AIRPORT,
*G:GATEDIST,WHICH CONTAINS THE MEAN WALKING DISTANCE FROM EACH GATE AND FOR
*             EACH TYPE OF FLIGHT:DOMESTIC,TRANSBORDER,INTERNATIONAL,
*G:TRANSDIS,THE TRANSPOSE OF G:GATEDIST
*G:GATES,WHICH CONTAINS THE WALKING DISTANCE FROM EACH GATE FOR EACH TYPE
*             OF FLIGHT(DOM,TRAB,INT'L)AND FOR EACH TYPE OF PASSENGER(ARRIVING,
*             DEPARTING,TRANSFER)
*AND M:TYPENAME.
*FINALLY M:GATEASSGN,WHICH IS CONSTRUCTED IN THE FIRST MACRO IN
*THIS FILE, IS NEEDED FOR THE REMAINING MACROS.
NAME         FINAL
*THIS MACRO  CONSTRUCTS A TABLE CONTAINING A LIST OF FLIGHTS AND THEIR GATE
*ASSIGNMENT ACCORDING TO THE LINEAR PROGRAM.
*
* PRINT REPORTS FOR SOLUTION FROM 'PLANES' MODEL
*
* $MODEL, $DDMODEL SET FOR GENERATED MODEL.
* $DDRESLT, N:CASENAME SET FOR OPTIMAL SOLUTION
*
REFORM M:PAIRINGS = COLS
*   FORM LIST OF ACTIVE PAIRINGS
I:ASSIGNED = 0
LOOP M:PAIRINGS(!1,0) <NE> DUMMY
     IF X:(M:PAIRINGS(!1,0),N:CASENAME) <EQ> 0., 2
        I:ASSIGNED = I:ASSIGNED + 1
        STUB M:ASSIGNED(I:ASSIGNED) = M:PAIRINGS(!1,0)
     CONTINUE
*
STUB M:FLIGHTS = MASK(M:ASSIGNED(!1,0),'******00')
FORM M:GATEASGN = M:FLIGHTS(STUB),GATE
M:GATEASGN(!1,GATE) = MASK(M:ASSIGNED(!1,0),'000000**')
DISPLAY M:GATEASGN
ENDATA
NAME         MEAN
TABLE M:SPEK=ORGATE,ORWD,ALGOGATE,ALGOWD,LPGATE,LPWD,PAX
...
STUB M:LO=MASK (M:GATEASGN(!1,0),'****0000')
FORM G:COMPARE=M:LO(STUB),M:SPEK(HEAD)
G:COMPARE(!1,LPWD)=G:TRANSDIS(M:GATEASGN(!1,GATE),
                 M:TYPENAME(DISTNAME,MASK(M:GATEASGN(!1,0),'00000*00')))
G:COMPARE("1,ORGATE)=G:ALGOTES("1,GTE)
G:COMPARE("1,ALGOGATE)=G:ALGOTES("1,KTE)
FORM M:ALGOTES=G:ALGOTES(STUB),G:ALGOTES(HEAD)
M:ALGOTES(!1,!2)=G:ALGOTES(!1,!2)
FORM M:COMPARE=G:COMPARE(STUB),G:COMPARE(HEAD)
M:COMPARE(!1,!2)=G:COMPARE(!1,!2)
G:COMPARE("1,PAX)=G:ALGOTES("1,ARR)+G:ALGOTES("1,DEP)+G:ALGOTES("1,TRA)
M:COMPARE(!1,PAX)=G:COMPARE(!1,PAX)
M:COMPARE(!1,PAX)=MASK(M:COMPARE(!1,PAX),'00000***')
M:COMPARE(!1,LPGATE)=M:GATEASGN(!1,GATE)
M:COMPARE(!1,ALGOGATE)=MASK(M:COMPARE(!1,ALGOGATE),'000000**')
M:COMPARE(!1,ORGATE)=MASK(M:COMPARE(!1,ORGATE),'000000**')
```

```
M:COMPARE(!1,LPWD)=MASK(M:COMPARE(!1,LPWD),'0000****')
G:COMPARE(!1,ALGOWD)=G:TRANSDIS(M:COMPARE(!1,ALGOGATE),
          M:TYPENAME(DISTNAME,MASK(M:GATEASGN(!1,0),'00000*00')))
G:COMPARE(!1,ORWD)=G:TRANSDIS(M:COMPARE(!1,ORGATE),
        M:TYPENAME(DISTNAME,MASK(M:GATEASGN(!1,0),'00000*00')))
M:COMPARE(!1,ORWD)=G:COMPARE(!1,ORWD)
M:COMPARE(!1,ORWD)=MASK(M:COMPARE(!1,ORWD),'0000****')
M:COMPARE(!1,ALGOWD)=G:COMPARE(!1,ALGOWD)
M:COMPARE(!1,ALGOWD)=MASK(M:COMPARE(!1,ALGOWD),'0000****')
DISPLAY M:COMPARE
ENDATA
NAME           ARRIVALS
TABLE M:A=DOM,TRAB,INT
...
FORM G:ARRI=G:GATES(STUB),M:A(HEAD)
G:ARRI(!1,TRAB)=G:GATES(!1,ARR1)
G:ARRI(!1,INT)=G:GATES(!1,ARR2)
G:ARRI(!1,DOM)=G:GATES(!1,ARR0)
TABLE M:SPEK=ORGATE,ORWD,ALGOGATE,ALGOWD,LPGATE,LPWD,PAX
...
STUB M:LO=MASK (M:GATEASGN(!1,0),'****0000')
FORM M:ARRIVALS=M:LO(STUB),M:SPEK(HEAD)
M:ARRIVALS(!1,LPWD)=G:ARRI(M:GATEASGN(!1,GATE),
            M:TYPENAME(DISTNAME,MASK(M:GATEASGN(!1,0),'00000*00')))
M:ARRIVALS(!1,LPWD)=MASK(M:ARRIVALS(!1,LPWD),'0000****')
M:ARRIVALS("1,ORGATE)=G:ALGOTES("1,GTE)
M:ARRIVALS("1,ALGOGATE)=G:ALGOTES("1,KTE)
M:ARRIVALS("1,PAX)=G:ALGOTES("1,ARR)
M:ARRIVALS(!1,PAX)=MASK(M:ARRIVALS(!1,PAX),'00000***')
M:ARRIVALS(!1,LPGATE)=M:GATEASGN(!1,GATE)
M:ARRIVALS(!1,ALGOGATE)=MASK(M:ARRIVALS(!1,ALGOGATE),'000000**')
M:ARRIVALS(!1,ORGATE)=MASK(M:ARRIVALS(!1,ORGATE),'000000**')
M:ARRIVALS(!1,ALGOWD)=G:ARRI(M:ARRIVALS(!1,ALGOGATE),
            M:TYPENAME(DISTNAME,MASK(M:GATEASGN(!1,0),'00000*00')))
M:ARRIVALS(!1,ALGOWD)=MASK(M:ARRIVALS(!1,ALGOWD),'0000****')
M:ARRIVALS(!1,ORWD)=G:ARRI(M:ARRIVALS(!1,ORGATE),
            M:TYPENAME(DISTNAME,MASK(M:GATEASGN(!1,0),'00000*00')))
M:ARRIVALS(!1,ORWD)=MASK(M:ARRIVALS(!1,ORWD),'0000****')
DISPLAY M:ARRIVALS
ENDATA
NAME           DEPARTUR
TABLE M:SPEK=ORGATE,ORWD,ALGOGATE,ALGOWD,LPGATE,LPWD,PAX
...
STUB M:LO=MASK (M:GATEASGN(!1,0),'****0000')
FORM G:DEPARTUR=M:LO(STUB),M:SPEK(HEAD)
TABLE M:D=DOM,TRAB,INT
...
FORM G:DEPI=G:GATES(STUB),M:D(HEAD)
G:DEPI(!1,DOM)=G:GATES(!1,DEP0)
G:DEPI(!1,TRAB)=G:GATES(!1,DEP1)
G:DEPI(!1,INT)=G:GATES(!1,DEP2)
G:DEPARTUR(!1,LPWD)=G:DEPI(M:GATEASGN(!1,GATE),
            M:TYPENAME(DISTNAME,MASK(M:GATEASGN(!1,0),'00000*00')))
G:DEPARTUR("1,ORGATE)=G:ALGOTES("1,GTE)
G:DEPARTUR("1,ALGOGATE)=G:ALGOTES("1,KTE)
```

```
FORM M:ALGOTES=G:ALGOTES(STUB),G:ALGOTES(HEAD)
M:ALGOTES(!1,!2)=G:ALGOTES(!1,!2)
FORM M:DEPARTUR=G:DEPARTUR(STUB),G:DEPARTUR(HEAD)
M:DEPARTUR(!1,!2)=G:DEPARTUR(!1,!2)
G:DEPARTUR("1,PAX)=G:ALGOTES("1,DEP)
M:DEPARTUR(!1,PAX)=G:DEPARTUR(!1,PAX)
M:DEPARTUR(!1,PAX)=MASK(M:DEPARTUR(!1,PAX),'00000***')
M:DEPARTUR(!1,LPGATE)=M:GATEASGN(!1,GATE)
M:DEPARTUR(!1,ALGOGATE)=MASK(M:DEPARTUR(!1,ALGOGATE),'000000**')
M:DEPARTUR(!1,ORGATE)=MASK(M:DEPARTUR(!1,ORGATE),'000000**')
M:DEPARTUR(!1,LPWD)=MASK(M:DEPARTUR(!1,LPWD),'0000****')
G:DEPARTUR(!1,ALGOWD)=G:DEPI(M:DEPARTUR(!1,ALGOGATE),
             M:TYPENAME(DISTNAME,MASK(M:GATEASGN(!1,0),'00000*00')))
M:DEPARTUR(!1,ALGOWD)=G:DEPARTUR(!1,ALGOWD)
M:DEPARTUR(!1,ALGOWD)=MASK(M:DEPARTUR(!1,ALGOWD),'0000****')
G:DEPARTUR(!1,ORWD)=G:DEPI(M:DEPARTUR(!1,ORGATE),
             M:TYPENAME(DISTNAME,MASK(M:GATEASGN(!1,0),'00000*00')))
M:DEPARTUR(!1,ORWD)=G:DEPARTUR(!1,ORWD)
M:DEPARTUR(!1,ORWD)=MASK(M:DEPARTUR(!1,ORWD),'0000****')
DISPLAY M:DEPARTUR
ENDATA
NAME          TRANSFER
TABLE M:SPEK=ORGATE,ORWD,ALGOGATE,ALGOWD,LPGATE,LPWD,PAX
...
STUB M:LO=MASK (M:GATEASGN(!1,0),'****0000')
FORM M:TRANSFER=M:LO(STUB),M:SPEK(HEAD)
M:TRANSFER(!1,LPWD)=G:GATES(M:GATEASGN(!1,GATE),TRANS)
M:TRANSFER("1,ALGOGATE)=G:ALGOTES("1,KTE)
M:TRANSFER("1,ORGATE)=G:ALGOTES("1,GTE)
M:TRANSFER ("1,PAX)=G:ALGOTES("1,TRA)
M:TRANSFER(!1,PAX)=MASK(M:TRANSFER(!1,PAX),'00000***')
M:TRANSFER(!1,LPGATE)=M:GATEASGN(!1,GATE)
M:TRANSFER(!1,ALGOGATE)=MASK(M:TRANSFER(!1,ALGOGATE),'000000**')
M:TRANSFER(!1,ORGATE)=MASK(M:TRANSFER(!1,ORGATE),'000000**')
M:TRANSFER(!1,ALGOWD)=G:GATES(M:TRANSFER(!1,ALGOGATE),TRANS)
M:TRANSFER(!1,ORWD)=G:GATES(M:TRANSFER(!1,ORGATE),TRANS)
M:TRANSFER(!1,ALGOWD)=MASK(M:TRANSFER(!1,ALGOWD),'0000****')
M:TRANSFER(!1,ORWD)=MASK(M:TRANSFER(!1,ORWD),'0000****')
M:TRANSFER(!1,LPWD)=MASK(M:TRANSFER(!1,LPWD),'0000****')
DISPLAY M:TRANSFER
ENDATA
NAME          HISTO
TABLE M:SO=OR,ALGO,LP
...
FORM G:HISTO=G:PR(HEAD),M:SO(HEAD)
LOOP G:PR(0,!1) <NE> RU
  E:LL=G:PR(NUM,!1)
   LOOP G:COMPARE(!2,ORWD)<LT> (100*(E:LL+1))
          IF G:COMPARE(!2,ORWD)<LT> (100*E:LL),2
               G:HISTO(!1,OR)=G:HISTO(!1,OR)+G:COMPARE(!2,PAX)
        E:ORTOT=E:ORTOT+(G:COMPARE(!2,ORWD)*G:COMPARE(!2,PAX))
   CONTINUE
   LOOP G:COMPARE(!2,ALGOWD) <LT> (100*(E:LL+1))
          IF G:COMPARE(!2,ALGOWD)<LT> (100*E:LL),2
               G:HISTO(!1,ALGO)=G:HISTO(!1,ALGO)+G:COMPARE(!2,PAX)
```

```
          E:ALGOTOT=E:ALGOTOT+(G:COMPARE(!2,ALGOWD)*G:COMPARE(!2,PAX))
   CONTINUE
   LOOP G:COMPARE(!2,LPWD) <LT> (100*(E:LL+1))
         IF G:COMPARE(!2,LPWD)<LT> (100*E:LL),2
              G:HISTO(!1,LP)=G:HISTO(!1,LP)+G:COMPARE(!2,PAX)
       E:LPTOT=E:LPTOT+(G:COMPARE(!2,LPWD)*G:COMPARE(!2,PAX))
   CONTINUE
E:SUM=E:SUM+G:HISTO(!1,OR)
CONTINUE
DISPLAY G:HISTO
DISPLAY E:SUM
E:ORAVG=E:ORTOT/E:SUM
E:ALGOAVG=E:ALGOTOT/E:SUM
E:LPAVG=E:LPTOT/E:SUM
DISPLAY E:ORAVG
DISPLAY E:ALGOAVG
DISPLAY E:LPAVG
FORM G:PERCENT=G:HISTO(STUB),G:HISTO(HEAD)
G:PERCENT(!1,!2)=G:HISTO(!1,!2)/E:SUM
G:PERCENT(!1,!2)=G:PERCENT(!1,!2)*100
DISPLAY G:PERCENT
TABLE G:SUMRY=ORAVG,ALGOAVG,LPAVG
NMBRS=E:ORAVG,E:ALGOAVG,E:LPAVG
...
ENDATA
NAME          DEPHISTO
FORM G:DEPARTUR=M:DEPARTUR (STUB),M:DEPARTUR(HEAD)
G:DEPARTUR(!1,ORWD)=G:DEPI(M:DEPARTUR(!1,ORGATE),
                        M:TYPENAME(DISTNAME,MASK(M:GATEASGN(!1,0),'00000*00')))
G:DEPARTUR(!1,ALGOWD)=G:DEPI(M:DEPARTUR(!1,ALGOGATE),
                        M:TYPENAME(DISTNAME,MASK(M:GATEASGN(!1,0),'00000*00')))
G:DEPARTUR(!1,LPWD)=G:DEPI(M:DEPARTUR(!1,LPGATE),
                        M:TYPENAME(DISTNAME,MASK(M:GATEASGN(!1,0),'00000*00')))
G:DEPARTUR("1,PAX)=G:ALGOTES("1,DEP)
TABLE M:SO=OR,ALGO,LP
...
FORM G:DEPHISTO=G:PR(HEAD),M:SO(HEAD)
LOOP G:PR(0,!1) <NE> EU
  E:LL=G:PR(NUM,!1)
   LOOP G:DEPARTUR(!2,ORWD)<LT> (100*(E:LL+1))
         IF G:DEPARTUR(!2,ORWD)<LT> (100*E:LL),2
              G:DEPHISTO(!1,OR)=G:DEPHISTO(!1,OR)+G:DEPARTUR(!2,PAX)
       E:ORTOTD=E:ORTOTD+(G:DEPARTUR(!2,ORWD)*G:DEPARTUR(!2,PAX))
   CONTINUE
   LOOP G:DEPARTUR(!2,ALGOWD) <LT> (100*(E:LL+1))
         IF G:DEPARTUR(!2,ALGOWD)<LT> (100*E:LL),2
              G:DEPHISTO(!1,ALGO)=G:DEPHISTO(!1,ALGO)+G:DEPARTUR(!2,PAX)
          E:ALGOTOTD=E:ALGOTOTD+(G:DEPARTUR(!2,ALGOWD)*G:DEPARTUR(!2,PAX))
   CONTINUE
   LOOP G:DEPARTUR(!2,LPWD) <LT> (100*(E:LL+1))
         IF G:DEPARTUR(!2,LPWD)<LT> (100*E:LL),2
              G:DEPHISTO(!1,LP)=G:DEPHISTO(!1,LP)+G:DEPARTUR(!2,PAX)
       E:LPTOTD=E:LPTOTD+(G:DEPARTUR(!2,LPWD)*G:DEPARTUR(!2,PAX))
   CONTINUE
E:SUMD=E:SUMD+G:DEPHISTO(!1,OR)
```

```
CONTINUE
DISPLAY G:DEPHISTO
DISPLAY E:SUMD
E:ORAVGD=E:ORTOTD/E:SUMD
E:ALGOAVGD=E:ALGOTOTD/E:SUMD
E:LPAVGD=E:LPTOTD/E:SUMD
DISPLAY E:ORAVGD
DISPLAY E:ALGOAVGD
DISPLAY E:LPAVGD
FORM G:PERCENT=G:DEPHISTO(STUB),G:DEPHISTO(HEAD)
G:PERCENT(!1,!2)=G:DEPHISTO(!1,!2)/E:SUMD
G:PERCENT(!1,!2)=G:PERCENT(!1,!2)*100
DISPLAY G:PERCENT
TABLE G:SUMRYD=ORAVGD,ALGOAVGD,LPAVGD
NMBRS=E:ORAVGD,E:ALGOAVGD,E:LPAVGD
...
ENDATA
NAME          ARRHISTO
FORM G:ARRIVALS=M:ARRIVALS (STUB),M:ARRIVALS(HEAD)
G:ARRIVALS(!1,ORWD)=G:ARRI(M:ARRIVALS(!1,ORGATE),
                    M:TYPENAME(DISTNAME,MASK(M:GATEASGN(!1,0),'00000*00')))
G:ARRIVALS(!1,ALGOWD)=G:ARRI(M:ARRIVALS(!1,ALGOGATE),
                    M:TYPENAME(DISTNAME,MASK(M:GATEASGN(!1,0),'00000*00')))
G:ARRIVALS(!1,LPWD)=G:ARRI(M:ARRIVALS(!1,LPGATE),
                    M:TYPENAME(DISTNAME,MASK(M:GATEASGN(!1,0),'00000*00')))
G:ARRIVALS("1,PAX)=G:ALGOTES("1,ARR)
TABLE M:SO=OR,ALGO,LP
...
FORM G:ARRHISTO=G:PR(HEAD),M:SO(HEAD)
LOOP G:PR(0,!1) <NE> RU
  E:LL=G:PR(NUM,!1)
   LOOP G:ARRIVALS(!2,ORWD)<LT> (100*(E:LL+1))
        IF G:ARRIVALS(!2,ORWD)<LT> (100*E:LL),2
             G:ARRHISTO(!1,OR)=G:ARRHISTO(!1,OR)+G:ARRIVALS(!2,PAX)
      E:ORTOTA=E:ORTOTA+(G:ARRIVALS(!2,ORWD)*G:ARRIVALS(!2,PAX))
   CONTINUE
   LOOP G:ARRIVALS(!2,ALGOWD) <LT> (100*(E:LL+1))
        IF G:ARRIVALS(!2,ALGOWD)<LT> (100*E:LL),2
             G:ARRHISTO(!1,ALGO)=G:ARRHISTO(!1,ALGO)+G:ARRIVALS(!2,PAX)
        E:ALGOTOTA=E:ALGOTOTA+(G:ARRIVALS(!2,ALGOWD)*G:ARRIVALS(!2,PAX))
   CONTINUE
   LOOP G:ARRIVALS(!2,LPWD) <LT> (100*(E:LL+1))
        IF G:ARRIVALS(!2,LPWD)<LT> (100*E:LL),2
             G:ARRHISTO(!1,LP)=G:ARRHISTO(!1,LP)+G:ARRIVALS(!2,PAX)
      E:LPTOTA=E:LPTOTA+(G:ARRIVALS(!2,LPWD)*G:ARRIVALS(!2,PAX))
   CONTINUE
E:SUMA=E:SUMA+G:ARRHISTO(!1,OR)
CONTINUE
DISPLAY G:ARRHISTO
DISPLAY E:SUMA
E:ORAVGA=E:ORTOTA/E:SUMA
E:ALGOAVGA=E:ALGOTOTA/E:SUMA
E:LPAVGA=E:LPTOTA/E:SUMA
DISPLAY E:ORAVGA
DISPLAY E:ALGOAVGA
```

```
DISPLAY E:LPAVGA
FORM G:PERCENT=G:ARRHISTO(STUB),G:ARRHISTO(HEAD)
G:PERCENT(!1,!2)=G:ARRHISTO(!1,!2)/E:SUMA
G:PERCENT(!1,!2)=G;PERCENT(!1,!2)*100
DISPLAY G:PERCENT
TABLE G:SUMRYA=ORAVGA,ALGOAVGA,LPAVGA
NMBRS=E:ORAVGA,E:ALGOAVGA,E:LPAVGA
...
ENDATA
NAME          TRFHISTO
FORM G:TRANSFER=M:TRANSFER(STUB),M:TRANSFER(HEAD)
G:TRANSFER(!1,LPWD)=G:GATES(M:GATEASGN(!1,GATE),TRANS)
G:TRANSFER  ("1,PAX)=G:ALGOTES("1,TRA)
G:TRANSFER(!1,ALGOWD)=G:GATES(M:TRANSFER(!1,ALGOGATE),TRANS)
G:TRANSFER(!1,ORWD)=G:GATES(M:TRANSFER(!1,ORGATE),TRANS)
TABLE M:SO=OR,ALGO,LP
...
FORM G:TRFHISTO=G:PR(HEAD),M:SO(HEAD)
LOOP G:PR(0,!1) <NE> RU
  E:LL=G:PR(NUM,!1)
    LOOP G:TRANSFER(!2,ORWD)<LT> (100*(E:LL+1))
          IF G:TRANSFER(!2,ORWD)<LT> (100*E:LL),2
               G:TRFHISTO(!1,OR)=G:TRFHISTO(!1,OR)+G:TRANSFER(!2,PAX)
       E:ORTOTT=E:ORTOTT+(G:TRANSFER(!2,ORWD)*G:TRANSFER(!2,PAX))
    CONTINUE
    LOOP G:TRANSFER(!2,ALGOWD) <LT> (100*(E:LL+1))
          IF G:TRANSFER(!2,ALGOWD)<LT> (100*E:LL),2
               G:TRFHISTO(!1,ALGO)=G:TRFHISTO(!1,ALGO)+G:TRANSFER(!2,PAX)
           E:ALGOTOTT=E:ALGOTOTT+(G:TRANSFER(!2,ALGOWD)*G:TRANSFER(!2,PAX))
    CONTINUE
    LOOP G:TRANSFER(!2,LPWD) <LT> (100*(E:LL+1))
          IF G:TRANSFER(!2,LPWD)<LT> (100*E:LL),2
               G:TRFHISTO(!1,LP)=G:TRFHISTO(!1,LP)+G:TRANSFER(!2,PAX)
       E:LPTOTT=E:LPTOTT+(G:TRANSFER(!2,LPWD)*G:TRANSFER(!2,PAX))
    CONTINUE
E:SUMT=E:SUMT+G:TRFHISTO(!1,OR)
CONTINUE
DISPLAY G:TRFHISTO
DISPLAY E:SUMT
E:ORAVGT=E:ORTOTT/E:SUMT
E:ALGOAVGT=E:ALGOTOTT/E:SUMT
E:LPAVGT=E:LPTOTT/E:SUMT
DISPLAY E:ORAVGT
DISPLAY E:ALGOAVGT
DISPLAY E:LPAVGT
FORM G:PERCENT=G:TRFHISTO(STUB),G:TRFHISTO(HEAD)
G:PERCENT(!1,!2)=G:TRFHISTO(!1,!2)/E:SUMT
G:PERCENT(!1,!2)=G:PERCENT(!1,!2)*100
DISPLAY G:PERCENT
TABLE G:SUMRYT=ORAVGT,ALGOAVGT,LPAVGT
NMBRS=E:ORAVGT,E:ALGOAVGT,E:LPAVGT
...
ENDATA
```

APPENDIX E

OUTPUT OF THE POSTPROCESSOR PROGRAM

Note: In Table 2.1 - 2.4, the following column headings refer to:

ORGATE	Gate originally assigned to the flight by Air Canada.
ORWD	Expected walking distance for a passenger in the flight accord- to the original assignment.
ALGOGATE	Gate assigned to the flight by the heuristic algorithm.
ALGOWD	Expected walking distance for a passenger in the flight according to the heuristic algorithm's assignment.
LGATE	Gate assigned to the flight by the linear program
LPWD	Expected walking distance for a passenger in the flight according to the linear program's assign- ment.

$M:COMPARE	=ORGATE	,ORWD	,ALGOGATE	,ALGOWD	,LPGATE	,LPWD	,PAX
F608	=77	,0924	,83	,0481	,85	,0561	,340
F243	=75	,1095	,85	,0561	,81	,0606	,340
F105	=81	,0606	,81	,0606	,78	,0772	,340
F310	=76	,0934	,80	,0621	,83	,0481	,271
F920	=91	,0386	,91	,0386	,91	,0386	,271
F440	=79	,0764	,97	,0627	,80	,0621	,180
F400	=80	,0621	,79	,0764	,79	,0764	,174
F402	=78	,0772	,78	,0772	,97	,0627	,174
F701	=87	,0782	,89	,0582	,93	,0667	,142
F721	=93	,0667	,99	,0612	,89	,0582	,142
F341	=73	,1283	,87	,0795	,87	,0795	,142
F982	=89	,0539	,93	,1057	,95	,1241	,142
F361	=74	,1104	,77	,0924	,77	,0924	,115
F442	=85	,0561	,76	,0934	,75	,1095	,115
F303	=71	,1277	,75	,1095	,76	,0934	,115
F727	=00	,0000	,00	,0000	,99	,0612	,000
F705	=89	,0582	,95	,0852	,99	,0612	,142
F164	=79	,0764	,80	,0621	,83	,0481	,340
F612	=76	,0934	,83	,0481	,85	,0561	,142
F107	=72	,1262	,79	,0764	,77	,0924	,142
F796	=95	,1241	,89	,0539	,89	,0539	,174
F774	=93	,0667	,99	,0612	,93	,0667	,142
F960	=87	,0359	,87	,0359	,85	,0505	,271
F404	=80	,0621	,97	,0627	,97	,0627	,174
F308	=73	,1283	,78	,0772	,80	,0621	,142
F444	=76	,0934	,77	,0924	,76	,0934	,142
F791	=91	,0386	,91	,0386	,91	,0386	,340
F117	=77	,0924	,83	,0481	,83	,0481	,340
F103	=71	,1277	,80	,0621	,79	,0764	,174
F450	=79	,0764	,76	,0934	,75	,1095	,115
F992	=85	,0505	,85	,0505	,87	,0359	,271
F365	=75	,1095	,78	,0772	,81	,0606	,142
F363	=73	,1283	,99	,1043	,80	,0621	,115
F406	=78	,0772	,97	,0627	,85	,0561	,174
F902	=91	,0386	,91	,0386	,91	,0386	,180
F642	=74	,1104	,87	,0795	,97	,0627	,142
F600	=76	,0934	,77	,0924	,79	,0764	,142
F123	=77	,0924	,83	,0481	,83	,0481	,340
F373	=71	,1277	,79	,0764	,77	,0924	,115
F246	=81	,0606	,81	,0606	,85	,0561	,174
F408	=80	,0621	,80	,0621	,81	,0606	,174
F371	=72	,1262	,78	,0772	,78	,0772	,115
F654	=83	,0481	,97	,0627	,80	,0621	,142
F778	=93	,0667	,89	,0582	,89	,0582	,142
F410	=78	,0772	,83	,0481	,83	,0481	,174
F385	=76	,1802	,99	,0612	,99	,0612	,142
F412	=80	,0621	,85	,0561	,83	,0481	,174
F347	=75	,1095	,78	,0772	,78	,0772	,115
F315	=79	,0764	,80	,0621	,85	,0561	,142
F249	=77	,0924	,83	,0481	,80	,0621	,271
F102	=83	,0481	,97	,0627	,97	,0627	,142
F725	=91	,0386	,91	,0386	,91	,0386	,174
F414	=78	,0772	,85	,0561	,83	,0481	,174
F349	=74	,1104	,79	,0764	,81	,0606	,142

Table E.1 A Partial List of the Flights, Their Gate Assignment and the Per Passenger Walking Distance under Each of the Three Assignment Policies.

$M:ARRIVALS	=ORGATE	,ORWD	,ALGOGATE	,ALGOWD	,LPGATE	,LPWD	,PAX
F608	=77	,0919	,83	,0509	,85	,0594	,119
F243	=75	,1102	,85	,0594	,81	,0556	,119
F105	=81	,0556	,81	,0556	,78	,0746	,119
F310	=76	,0926	,80	,0566	,83	,0509	,095
F920	=91	,0299	,91	,0299	,91	,0299	,095
F440	=79	,0739	,97	,0510	,80	,0566	,063
F400	=80	,0566	,79	,0739	,79	,0739	,061
F402	=78	,0746	,78	,0746	,97	,0510	,061
F701	=87	,0807	,89	,0553	,93	,0598	,050
F721	=93	,0598	,99	,0418	,89	,0553	,050
F341	=73	,1285	,87	,0855	,87	,0855	,050
F982	=89	,0601	,93	,1154	,95	,1337	,050
F361	=74	,1106	,77	,0919	,77	,0919	,040
F442	=85	,0594	,76	,0926	,75	,1102	,040
F303	=71	,1287	,75	,1102	,76	,0926	,040
F727	=00	,0000	,00	,0000	,99	,0418	,000
F705	=89	,0553	,95	,0781	,99	,0418	,050
F164	=79	,0739	,80	,0566	,83	,0509	,119
F612	=76	,0926	,83	,0509	,85	,0594	,050
F107	=72	,1269	,79	,0739	,77	,0919	,050
F796	=95	,1337	,89	,0601	,89	,0601	,061
F774	=93	,0598	,99	,0418	,93	,0598	,050
F960	=87	,0347	,87	,0347	,85	,0428	,095
F404	=80	,0566	,97	,0510	,97	,0510	,061
F308	=73	,1285	,78	,0746	,80	,0566	,050
F444	=76	,0926	,77	,0919	,76	,0926	,050
F791	=91	,0299	,91	,0299	,91	,0299	,119
F117	=77	,0919	,83	,0509	,83	,0509	,119
F103	=71	,1287	,80	,0566	,79	,0739	,061
F450	=79	,0739	,76	,0926	,75	,1102	,040
F992	=85	,0428	,85	,0428	,87	,0347	,095
F365	=75	,1102	,78	,0746	,81	,0556	,050
F363	=73	,1285	,99	,0957	,80	,0566	,040
F406	=78	,0746	,97	,0510	,85	,0594	,061
F902	=91	,0299	,91	,0299	,91	,0299	,063
F642	=74	,1106	,87	,0855	,97	,0510	,050
F600	=76	,0926	,77	,0919	,79	,0739	,050
F123	=77	,0919	,83	,0509	,83	,0509	,119
F373	=71	,1287	,79	,0739	,77	,0919	,040
F246	=81	,0556	,81	,0556	,85	,0594	,061
F408	=80	,0566	,80	,0566	,81	,0556	,061
F371	=72	,1269	,78	,0746	,78	,0746	,040
F654	=83	,0509	,97	,0510	,80	,0566	,050
F778	=93	,0598	,89	,0553	,89	,0553	,050
F410	=78	,0746	,83	,0509	,83	,0509	,061
F385	=76	,2013	,99	,0418	,99	,0418	,050
F412	=80	,0566	,85	,0594	,83	,0509	,061
F347	=75	,1102	,78	,0746	,78	,0746	,040
F315	=79	,0739	,80	,0566	,85	,0594	,050
F249	=77	,0919	,83	,0509	,80	,0566	,095
F102	=83	,0509	,97	,0510	,97	,0510	,050
F725	=91	,0299	,91	,0299	,91	,0299	,061
F414	=78	,0746	,85	,0594	,83	,0509	,061
F349	=74	,1106	,79	,0739	,81	,0556	,050

Table E.2 A Partial List of the Flights, Their Gate Assignment and the Expected Walking Distance for Arriving Passengers under Each of the Three Assignment Policies.

$M:DEPARTUR	=ORGATE	,ORWD	,ALGOGATE	,ALGOWD	,LPGATE	,LPWD	,PAX
F608	=77	,0935	,83	,0349	,85	,0434	,170
F243	=75	,1118	,85	,0434	,81	,0572	,170
F105	=81	,0572	,81	,0572	,78	,0752	,170
F310	=76	,0932	,80	,0582	,83	,0349	,136
F920	=91	,0193	,91	,0193	,91	,0193	,136
F440	=79	,0755	,97	,0350	,80	,0582	,090
F400	=80	,0582	,79	,0755	,79	,0755	,087
F402	=78	,0752	,78	,0752	,97	,0350	,087
F701	=87	,0701	,89	,0447	,93	,0492	,071
F721	=93	,0492	,99	,0312	,89	,0447	,071
F341	=73	,1301	,87	,0695	,87	,0695	,071
F982	=89	,0329	,93	,0882	,95	,1065	,071
F361	=74	,1112	,77	,0935	,77	,0935	,058
F442	=85	,0434	,76	,0932	,75	,1118	,058
F303	=71	,1303	,75	,1118	,76	,0932	,058
F727	=00	,0000	,00	,0000	,99	,0312	,000
F705	=89	,0447	,95	,0675	,99	,0312	,071
F164	=79	,0755	,80	,0582	,83	,0349	,170
F612	=76	,0932	,83	,0349	,85	,0434	,071
F107	=72	,1285	,79	,0755	,77	,0935	,071
F796	=95	,1065	,89	,0329	,89	,0329	,087
F774	=93	,0492	,99	,0312	,93	,0492	,071
F960	=87	,0177	,87	,0177	,85	,0438	,136
F404	=80	,0582	,97	,0350	,97	,0350	,087
F308	=73	,1301	,78	,0752	,80	,0582	,071
F444	=76	,0932	,77	,0935	,76	,0932	,071
F791	=91	,0193	,91	,0193	,91	,0193	,170
F117	=77	,0935	,83	,0349	,83	,0349	,170
F103	=71	,1303	,80	,0582	,79	,0755	,087
F450	=79	,0755	,76	,0932	,75	,1118	,058
F992	=85	,0438	,85	,0438	,87	,0177	,136
F365	=75	,1118	,78	,0752	,81	,0572	,071
F363	=73	,1301	,99	,0797	,80	,0582	,058
F406	=78	,0752	,97	,0350	,85	,0434	,087
F902	=91	,0193	,91	,0193	,91	,0193	,090
F642	=74	,1112	,87	,0695	,97	,0350	,071
F600	=76	,0932	,77	,0935	,79	,0755	,071
F123	=77	,0935	,83	,0349	,83	,0349	,170
F373	=71	,1303	,79	,0755	,77	,0935	,058
F246	=81	,0572	,81	,0572	,85	,0434	,087
F408	=80	,0582	,80	,0582	,81	,0572	,087
F371	=72	,1285	,78	,0752	,78	,0752	,058
F654	=83	,0349	,97	,0350	,80	,0582	,071
F778	=93	,0492	,89	,0447	,89	,0447	,071
F410	=78	,0752	,83	,0349	,83	,0349	,087
F385	=76	,1907	,99	,0312	,99	,0312	,071
F412	=80	,0582	,85	,0434	,83	,0349	,087
F347	=75	,1118	,78	,0752	,78	,0752	,058
F315	=79	,0755	,80	,0582	,85	,0434	,071
F249	=77	,0935	,83	,0349	,80	,0582	,136
F102	=83	,0349	,97	,0350	,97	,0350	,071
F725	=91	,0193	,91	,0193	,91	,0193	,087
F414	=78	,0752	,85	,0434	,83	,0349	,087
F349	=74	,1112	,79	,0755	,81	,0572	,071

Table E.3 A Partial List of the Flights, Their Gate Assignment and the Expected Walking Distance for Departing Passengers Under Each of the Three Assignment Policies.

$M:TRANSFER	=ORGATE	,ORWD	,ALGOGATE	,ALGOWD	,LPGATE	,LPWD	,PAX
F608	=77	,0910	,83	,0866	,85	,0908	,051
F243	=75	,1008	,85	,0908	,81	,0842	,051
F105	=81	,0842	,81	,0842	,78	,0909	,051
F310	=76	,0966	,80	,0887	,83	,0866	,040
F920	=91	,1241	,91	,1241	,91	,1241	,040
F440	=79	,0861	,97	,1830	,80	,0887	,027
F400	=80	,0887	,79	,0861	,79	,0861	,026
F402	=78	,0909	,78	,0909	,97	,1830	,026
F701	=87	,0999	,89	,1106	,93	,1423	,021
F721	=93	,1423	,99	,2074	,89	,1106	,021
F341	=73	,1223	,87	,0999	,87	,0999	,021
F982	=89	,1106	,93	,1423	,95	,1615	,021
F361	=74	,1081	,77	,0910	,77	,0910	,017
F442	=85	,0908	,76	,0966	,75	,1008	,017
F303	=71	,1175	,75	,1008	,76	,0966	,017
F727	=00	,0000	,00	,0000	,99	,2074	,000
F705	=89	,1106	,95	,1615	,99	,2074	,021
F164	=79	,0861	,80	,0887	,83	,0866	,051
F612	=76	,0966	,83	,0866	,85	,0908	,021
F107	=72	,1181	,79	,0861	,77	,0910	,021
F796	=95	,1615	,89	,1106	,89	,1106	,026
F774	=93	,1423	,99	,2074	,93	,1423	,021
F960	=87	,0999	,87	,0999	,85	,0908	,040
F404	=80	,0887	,97	,1830	,97	,1830	,026
F308	=73	,1223	,78	,0909	,80	,0887	,021
F444	=76	,0966	,77	,0910	,76	,0966	,021
F791	=91	,1241	,91	,1241	,91	,1241	,051
F117	=77	,0910	,83	,0866	,83	,0866	,051
F103	=71	,1175	,80	,0887	,79	,0861	,026
F450	=79	,0861	,76	,0966	,75	,1008	,017
F992	=85	,0908	,85	,0908	,87	,0999	,040
F365	=75	,1008	,78	,0909	,81	,0842	,021
F363	=73	,1223	,99	,2074	,80	,0887	,017
F406	=78	,0909	,97	,1830	,85	,0908	,026
F902	=91	,1241	,91	,1241	,91	,1241	,027
F642	=74	,1081	,87	,0999	,97	,1830	,021
F600	=76	,0966	,77	,0910	,79	,0861	,021
F123	=77	,0910	,83	,0866	,83	,0866	,051
F373	=71	,1175	,79	,0861	,77	,0910	,017
F246	=81	,0842	,81	,0842	,85	,0908	,026
F408	=80	,0887	,80	,0887	,81	,0842	,026
F371	=72	,1181	,78	,0909	,78	,0909	,017
F654	=83	,0866	,97	,1830	,80	,0887	,021
F778	=93	,1423	,89	,1106	,89	,1106	,021
F410	=78	,0909	,83	,0866	,83	,0866	,026
F385	=76	,0966	,99	,2074	,99	,2074	,021
F412	=80	,0887	,85	,0908	,83	,0866	,026
F347	=75	,1008	,78	,0909	,78	,0909	,017
F315	=79	,0861	,80	,0887	,85	,0908	,021
F249	=77	,0910	,83	,0866	,80	,0887	,040
F102	=83	,0866	,97	,1830	,97	,1830	,021
F725	=91	,1241	,91	,1241	,91	,1241	,026
F414	=78	,0909	,85	,0908	,83	,0866	,026
F349	=74	,1081	,79	,0861	,81	,0842	,021

Table E.4 A Partial List of the Flights, Their Gate Assignment and the Expected Walking Distance for Transferring Passengers Under Each of the Three Assignment Policies.

:PERMEAN	=	OR	ALGO	LP
100	=	.	.	.
200	=	.	.	.
300	=	10.288305	15.376247	16.522212
400	=	4.8921124	13.747960	15.194923
500	=	8.0108794	16.177697	17.106074
600	=	18.701723	24.315503	26.480508
700	=	15.318223	13.773345	11.963735
800	=	3.6264733	4.2792384	1.9655485
900	=	14.531278	6.7198549	7.0208522
1000	=	6.5457842	4.2611061	2.8141432
1100	=	4.4786945	.83408885	.41704442
1200	=	13.091568	.	.51495920
1300	=	.	.51495920	.
1400	=	.	.	.
1500	=	.	.	.
1600	=	.	.	.
1700	=	.	.	.
1800	=	.51495920	.	.
1900	=	.	.	.
2000	=	.	.	.
2100	=	.	.	.
2200	=	.	.	.
2300	=	.	.	.
2400	=	.	.	.

Table E.5 Statistical Distribution of the Overall Mean Walking Distance (used to draw Fig. 4.1)

:PERARR	=	OR	,	ALGO	,	LP
100	=	.	,	.	,	.
200	=	5.7119205	,	6.2293046	,	7.3778974
300	=	4.5633278	,	9.1266556	,	9.1266556
400	=	4.4081126	,	4.0355960	,	6.1879139
500	=	25.693295	,	46.637003	,	48.520281
600	=	1.5004139	,	3.5802980	,	4.0976821
700	=	13.348510	,	13.017384	,	12.013659
800	=	5.5980960	,	5.0496689	,	1.9143212
900	=	14.021109	,	7.5538079	,	7.0260762
1000	=	.51738411	,	.	,	.
1100	=	11.020281	,	4.2528974	,	3.2181291
1200	=	11.485927	,	.	,	.
1300	=	1.6142384	,	.51738411	,	.51738411
1400	=	.	,	.	,	.
1500	=	.	,	.	,	.
1600	=	.	,	.	,	.
1700	=	.	,	.	,	.
1800	=	.	,	.	,	.
1900	=	.	,	.	,	.
2000	=	.51738411	,	.	,	.
2100	=	.	,	.	,	.
2200	=	.	,	.	,	.
2300	=	.	,	.	,	.
2400	=	.	,	.	,	.

Table E.6 Statistical Distribution of the Mean Walking Distance for an Arriving Passenger (used to draw Fig. 4.2)

G:PERTRANS	=	OR	,	ALGO	,	LP
100	=	.	,	.	,	.
200	=	.	,	.	,	.
300	=	.	,	.	,	.
400	=	.	,	.	,	.
500	=	.	,	.	,	.
600	=	.	,	.	,	.
700	=	.	,	.	,	.
800	=	28.198879	,	35.413112	,	37.728491
900	=	31.026078	,	32.366561	,	32.561540
1000	=	10.041433	,	2.5834755	,	2.5834755
1100	=	9.9683159	,	6.1174750	,	6.1174750
1200	=	12.308067	,	7.2142335	,	7.3848404
1300	=	.	,	.	,	.
1400	=	4.2164270	,	2.8271996	,	2.3153790
1500	=	.	,	.	,	.
1600	=	1.1455033	,	1.6573239	,	1.4867170
1700	=	.	,	.	,	.
1800	=	1.1455033	,	6.8486473	,	6.7999025
1900	=	.	,	.	,	.
2000	=	1.9497928	,	4.9719717	,	3.0221789
2100	=	.	,	.	,	.
2200	=	.	,	.	,	.
2300	=	.	,	.	,	.
2400	=	.	,	.	,	.

.

Fig. E.7 Statistical Distribution of the Expected Walking Distance for a Departing Passenger (used to draw Fig. 4.3)

G:PERDEP	=	OR	ALGO	LP
100	=	10.291136	15.382387	16.526651
200	=	.	.	.
300	=	6.3948436	25.709733	28.128621
400	=	10.747393	12.767961	14.172943
500	=	17.410197	19.329374	17.431924
600	=	1.6584589	3.1068946	1.9191773
700	=	14.332271	12.188586	11.037080
800	=	.	1.1442642	.63006952
900	=	14.520568	7.2421784	7.0249131
1000	=	.63006952	.	.51419467
1100	=	11.029838	2.6144264	2.6144264
1200	=	4.3018540	.	.
1300	=	8.1691773	.51419467	.
1400	=	.	.	.
1500	=	.	.	.
1600	=	.	.	.
1700	=	.	.	.
1800	=	.	.	.
1900	=	.51419467	.	.
2000	=	.	.	.
2100	=	.	.	.
2200	=	.	.	.
2300	=	.	.	.
2400	=	.	.	.

Table E.8 Statistical Distribution of the Expected Walking Distance for a Transfer Passenger (used to draw Fig. 4.4)

REFERENCES

1. Braaksma, John, P., [1976],"Time Stamping: A New Way to Survey Pedestrian Traffic in Airport Terminals", Transportation Research Record 588, pp. 27-34.

2. Braaksma, John, P. [1977], "Reducing Walking Distances at Existing Airports", Airport Forum, No.4, August 1977, pp. 135-142.

3. Bustinduy, Javier, [1977], "Gate Assignment Algorithms at Airport Terminals", unpublished M.S. Thesis, Department of Civil Engineering, Massachusetts Institute of Technology

4. Hoffman, A. T. and Kruskal, J. B., Integral Boundary Points of Convex Polyhedra, Linear Inequalities and Related Systems, Princeton University Press, Princeton, N.J., 1956, pp. 223-246.

5. Mangoubi, Rami [1978], "Testing Gate Assignment Algorithms at Airport Terminals", unpublished B.S. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology.

6. National Bureau of Economic Research [1975], SESAME Reference Manual, Cambridge, Massachusetts.

7. Ibid. [1975], DATAMAT Reference Manual, Cambridge, Mass.