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.
<table>
<thead>
<tr>
<th>CONTENTS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>2</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>3</td>
</tr>
<tr>
<td>List of Figures</td>
<td>6</td>
</tr>
<tr>
<td>List of Tables</td>
<td>7</td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>10</td>
</tr>
<tr>
<td>1.1 The Problem</td>
<td>10</td>
</tr>
<tr>
<td>1.2 A Brief Review of Past Research</td>
<td>11</td>
</tr>
<tr>
<td>1.3 Outline of Research and Contributions</td>
<td>15</td>
</tr>
<tr>
<td>2. THE &quot;CROWDEST-COME-BEST-SERVE&quot; ALGORITHM</td>
<td>17</td>
</tr>
<tr>
<td>2.1 Description of the Heuristic Algorithms</td>
<td>17</td>
</tr>
<tr>
<td>2.2 Proof of the Algorithm's Suboptimality</td>
<td>21</td>
</tr>
<tr>
<td>2.3 Data Used to Solve the Problem</td>
<td>24</td>
</tr>
<tr>
<td>2.3.1 Flight and Passenger Information</td>
<td>25</td>
</tr>
<tr>
<td>2.3.2 Walking Distance</td>
<td>27</td>
</tr>
<tr>
<td>3. SOLVING THE PROBLEM AS A LINEAR PROGRAM</td>
<td>30</td>
</tr>
<tr>
<td>3.1 Formulation of the Linear Program</td>
<td>34</td>
</tr>
<tr>
<td>3.2 Solving an Example Problem for a Small Airport</td>
<td>45</td>
</tr>
<tr>
<td>3.3 Implementation of the Model on the Computer</td>
<td>52</td>
</tr>
<tr>
<td>4. RESULTS</td>
<td>54</td>
</tr>
<tr>
<td>4.1 Comparison of the Two Methods of Solution</td>
<td>54</td>
</tr>
<tr>
<td>4.2 Computational Costs</td>
<td>66</td>
</tr>
<tr>
<td>5. CONCLUSION</td>
<td>70</td>
</tr>
<tr>
<td>Appendix</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>6.</td>
<td><strong>APPENDIX A:</strong> COMPUTER PROGRAM IMPLEMENTING THE &quot;CROWDEST-COME-BEST-SERVE&quot; ALGORITHM</td>
</tr>
<tr>
<td>7.</td>
<td><strong>APPENDIX B:</strong> RESULTS OF THE &quot;CROWDEST-COME-BEST-SERVE&quot; ALGORITHM</td>
</tr>
<tr>
<td>8.</td>
<td><strong>APPENDIX C:</strong> PREPROCESSOR OR MODEL GENERATING PROGRAM</td>
</tr>
<tr>
<td>9.</td>
<td><strong>APPENDIX D:</strong> THE POSTPROCESSOR PROGRAM</td>
</tr>
<tr>
<td>10.</td>
<td><strong>APPENDIX E:</strong> OUTPUT OF THE POSTPROCESSOR PROGRAM</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Flow Chart for the &quot;Crowdest-Come-Best-Serve&quot; Algorithm</td>
<td>19</td>
</tr>
<tr>
<td>2.2</td>
<td>Plan of Terminal 2 at Toronto International Airport</td>
<td>29</td>
</tr>
<tr>
<td>3.1</td>
<td>Diagram Showing Conflict Sets L(i), i=1 to 5 for Example Problem</td>
<td>47</td>
</tr>
<tr>
<td>4.1</td>
<td>Cumulative Distributions of the Overall Mean Walking Distance for All Passengers under each of the Three Different Assignment Policies</td>
<td>57</td>
</tr>
<tr>
<td>4.2</td>
<td>Cumulative Distributions of the Expected Walking Distance for Arriving Passengers under Each of the Three Assignment Policies</td>
<td>59</td>
</tr>
<tr>
<td>4.3</td>
<td>Cumulative Distribution of the Expected Walking Distance for Departing Passengers under Each of the Three Assignment Policies</td>
<td>61</td>
</tr>
<tr>
<td>4.4</td>
<td>Cumulative Distributions of the Expected Walking Distance for Transfer Passengers under Each of the Three Assignment Policies</td>
<td>64</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Various Statistics on Passengers' Walking Distance at Toronto Terminal No. 2</td>
<td>14</td>
</tr>
<tr>
<td>2.1</td>
<td>Scheduled Flights Information for Example Given in Section 2.2</td>
<td>22</td>
</tr>
<tr>
<td>2.2</td>
<td>Average Walking Distances for Gates A and B</td>
<td>22</td>
</tr>
<tr>
<td>2.3</td>
<td>Gates and Walking Distances for Both the &quot;Crowdest-Come-Best-Serve&quot; and the Optimal Assignment Policies for the Example Problem</td>
<td>23a</td>
</tr>
<tr>
<td>2.4</td>
<td>Average Walking Distances for all Passengers for the Two Assignment Policies</td>
<td>23a</td>
</tr>
<tr>
<td>2.5</td>
<td>Summary of Aircraft Data for Toronto Terminal No. 2</td>
<td>26</td>
</tr>
<tr>
<td>2.6</td>
<td>Walking Distances for Non-Transfer Passengers</td>
<td>30</td>
</tr>
<tr>
<td>2.7</td>
<td>Matrix of Inter-gate Distances</td>
<td>32</td>
</tr>
<tr>
<td>3.1</td>
<td>Average Gate Walking Distance per Passenger (in feet) for Hypothetical Airport</td>
<td>46</td>
</tr>
<tr>
<td>3.2</td>
<td>Flight Information for Example Problem</td>
<td>46</td>
</tr>
<tr>
<td>3.3</td>
<td>Optimal Gate Assignment and Walking Distances for Each Flight</td>
<td>50</td>
</tr>
<tr>
<td>4.1a</td>
<td>Mean and Mean Saving in the Expected Distance for All Passengers (in ft.) under the Three Assignment Policies</td>
<td>58</td>
</tr>
<tr>
<td>4.1b</td>
<td>Percentiles of Expected Walking Distances for All Passengers under the Three Assignment Policies</td>
<td>58</td>
</tr>
<tr>
<td>Table</td>
<td>Name</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.2a</td>
<td>Mean and Mean Saving in Expected Distance for Arriving Passengers</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>(in feet) under the Three Assignment Policies</td>
<td></td>
</tr>
<tr>
<td>4.2b</td>
<td>Percentiles of Expected Walking Distances for Arriving Passengers</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>under the Three Different Policies</td>
<td></td>
</tr>
<tr>
<td>4.3a</td>
<td>Mean and Mean Saving in Expected Walking Distance for Departing</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>Passengers under Each of the Three Assignment Policies</td>
<td></td>
</tr>
<tr>
<td>4.3b</td>
<td>Percentile of Expected Walking Distance for Departing Passengers</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>under Each of the Three Policies</td>
<td></td>
</tr>
<tr>
<td>4.4a</td>
<td>Mean and Mean Difference in Walking Distance for Transfer Passengers</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>under Each of the Three Assignment Policies</td>
<td></td>
</tr>
<tr>
<td>4.4b</td>
<td>Percentiles of Expected Walking Distances for Transfer Passengers</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>under the Three Assignment Policies</td>
<td></td>
</tr>
<tr>
<td>4.5a</td>
<td>Resource Utilization and Their Costs for the &quot;Crowdest-Come-</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>Best-Serve&quot; Algorithm</td>
<td></td>
</tr>
<tr>
<td>4.5b</td>
<td>Very Approximate Costs for Running the Linear Program</td>
<td>67</td>
</tr>
</tbody>
</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          102

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     103

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  104
<table>
<thead>
<tr>
<th>Table</th>
<th>Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>E.4</td>
<td>A Partial List of the Flights, Their Gate Assignment and the</td>
<td>105</td>
</tr>
<tr>
<td></td>
<td>Expected Walking Distance for Transferring Passengers under Each of</td>
<td></td>
</tr>
<tr>
<td></td>
<td>the Three Assignment Policies</td>
<td></td>
</tr>
<tr>
<td>E.5</td>
<td>Statistical Distribution of the Overall Mean Walking Distance</td>
<td>106</td>
</tr>
<tr>
<td>E.6</td>
<td>Statistical Distribution of the Mean Walking Distance for an</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td>Arriving Passenger</td>
<td></td>
</tr>
<tr>
<td>E.7</td>
<td>Statistical Distribution of the Expected Walking Distance for a</td>
<td>108</td>
</tr>
<tr>
<td></td>
<td>Departing Passenger</td>
<td></td>
</tr>
<tr>
<td>E.8</td>
<td>Statistical Distribution of the Expected Walking Distance for a</td>
<td>109</td>
</tr>
<tr>
<td></td>
<td>Transfer Passenger</td>
<td></td>
</tr>
</tbody>
</table>
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.
<table>
<thead>
<tr>
<th></th>
<th>1973</th>
<th>1974</th>
<th>1975</th>
</tr>
</thead>
<tbody>
<tr>
<td>85th Percentile</td>
<td>1,300</td>
<td>1,100</td>
<td>1,165</td>
</tr>
<tr>
<td>Mean Distance</td>
<td>923</td>
<td>744</td>
<td>800</td>
</tr>
<tr>
<td>50th Percentile</td>
<td>890</td>
<td>660</td>
<td>765</td>
</tr>
<tr>
<td>15th Percentile</td>
<td>480</td>
<td>380</td>
<td>430</td>
</tr>
</tbody>
</table>

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 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 assignment 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

```
start

d = 0

d = d + 1

STOP

Yes

f = m + 1?

No

S(Available gates, walk.distance)

g = 0

g = g + 1

Yes

g = n + 1?

No

Gate suitable to AC?

Yes

g = s

No

Compute WD(g)

g = n + 1

Yes

Select STAR GATE

Clean G and store unoccupied gates in G

WD(g) = walking distance for gate g.
```
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.
<table>
<thead>
<tr>
<th>Flight</th>
<th>AC</th>
<th>Arrival</th>
<th>Departure</th>
<th>Pax</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B727</td>
<td>10:00</td>
<td>10:40</td>
<td>240</td>
</tr>
<tr>
<td>2</td>
<td>B747</td>
<td>10:00</td>
<td>13:00</td>
<td>400</td>
</tr>
<tr>
<td>3</td>
<td>B727</td>
<td>10:45</td>
<td>11:30</td>
<td>240</td>
</tr>
<tr>
<td>4</td>
<td>B727</td>
<td>12:00</td>
<td>13:20</td>
<td>240</td>
</tr>
</tbody>
</table>

Table 2.1 Scheduled Flights Information for Example Given in Section 2.2

<table>
<thead>
<tr>
<th>Gates</th>
<th>Walking Distance (ft)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>650</td>
</tr>
<tr>
<td>B</td>
<td>800</td>
</tr>
</tbody>
</table>

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
<table>
<thead>
<tr>
<th>Flight</th>
<th>AC</th>
<th>PAX</th>
<th>Algorithm's Assignment</th>
<th>Optimal Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Gate</td>
<td>Walking Distance</td>
</tr>
<tr>
<td>1</td>
<td>B727</td>
<td>240</td>
<td>B</td>
<td>800</td>
</tr>
<tr>
<td>2</td>
<td>B747</td>
<td>400</td>
<td>A</td>
<td>650</td>
</tr>
<tr>
<td>3</td>
<td>B727</td>
<td>240</td>
<td>B</td>
<td>800</td>
</tr>
<tr>
<td>4</td>
<td>B727</td>
<td>240</td>
<td>B</td>
<td>800</td>
</tr>
</tbody>
</table>

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 heuristics 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).
### Summary of Aircraft Data for Toronto Terminal No. 2

<table>
<thead>
<tr>
<th>AIRCRAFT</th>
<th>CAPACITY</th>
<th>OCCUPATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>B747</td>
<td>382</td>
<td>248</td>
</tr>
<tr>
<td>L10</td>
<td>262</td>
<td>170</td>
</tr>
<tr>
<td>D8S</td>
<td>210</td>
<td>137</td>
</tr>
<tr>
<td>DC8</td>
<td>140</td>
<td>91</td>
</tr>
<tr>
<td>72S</td>
<td>135</td>
<td>88</td>
</tr>
<tr>
<td>727</td>
<td>135</td>
<td>88</td>
</tr>
<tr>
<td>D9S</td>
<td>110</td>
<td>72</td>
</tr>
<tr>
<td>DC9</td>
<td>90</td>
<td>59</td>
</tr>
</tbody>
</table>

Table 2.5
Of the arriving passengers, it is assumed that 30% or 74 intend to board another flight at Toronto Terminal No. 2. These connecting 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])
<table>
<thead>
<tr>
<th>GATE</th>
<th>ARRIVALS</th>
<th>DEPARTURES</th>
</tr>
</thead>
<tbody>
<tr>
<td>71</td>
<td>1287</td>
<td>1285</td>
</tr>
<tr>
<td>72</td>
<td>1269</td>
<td>1285</td>
</tr>
<tr>
<td>73</td>
<td>1285</td>
<td>1285</td>
</tr>
<tr>
<td>74</td>
<td>1106</td>
<td>1112</td>
</tr>
<tr>
<td>75</td>
<td>1102</td>
<td>1118</td>
</tr>
<tr>
<td>76</td>
<td>926</td>
<td>932</td>
</tr>
<tr>
<td>77</td>
<td>919</td>
<td>935</td>
</tr>
<tr>
<td>78</td>
<td>746</td>
<td>752</td>
</tr>
<tr>
<td>79</td>
<td>739</td>
<td>755</td>
</tr>
<tr>
<td>80</td>
<td>566</td>
<td>582</td>
</tr>
<tr>
<td>81</td>
<td>556</td>
<td>572</td>
</tr>
<tr>
<td>82</td>
<td>509</td>
<td>349</td>
</tr>
<tr>
<td>83</td>
<td>594</td>
<td>434</td>
</tr>
<tr>
<td>84</td>
<td>855</td>
<td>701</td>
</tr>
<tr>
<td>85</td>
<td>1109</td>
<td>949</td>
</tr>
<tr>
<td>86</td>
<td>1363</td>
<td>1203</td>
</tr>
<tr>
<td>87</td>
<td>1662</td>
<td>1502</td>
</tr>
<tr>
<td>88</td>
<td>1845</td>
<td>492</td>
</tr>
<tr>
<td>89</td>
<td>510</td>
<td>350</td>
</tr>
<tr>
<td>90</td>
<td>957</td>
<td>797</td>
</tr>
</tbody>
</table>

Table 2.6
Walking Distances for Non-Transfer Passengers
[in feet]

-30-
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,

$$p_{kj} = p = \frac{1}{N} \quad \forall k, j = 1, \ldots, N$$  

(Eq. 2.1)

$N$ being the number of gates at the airport.
Table 2.7
Matrix of Inter-gate distances
[in feet]

| GATE | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 |
|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|      | 0  | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100| 110| 120| 130| 140| 150| 160| 170| 180| 190| 200| 210| 220| 230| 240| 250| 260| 270|
| 71   | 0  | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100| 110| 120| 130| 140| 150| 160| 170| 180| 190| 200| 210| 220| 230| 240| 250| 260| 270|
| 72   | 0  | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100| 110| 120| 130| 140| 150| 160| 170| 180| 190| 200| 210| 220| 230| 240| 250| 260| 270| 280| 290|
| 73   | 0  | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100| 110| 120| 130| 140| 150| 160| 170| 180| 190| 200| 210| 220| 230| 240| 250| 260| 270| 280| 290|
| 74   | 0  | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100| 110| 120| 130| 140| 150| 160| 170| 180| 190| 200| 210| 220| 230| 240| 250| 260| 270|
| 75   | 0  | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100| 110| 120| 130| 140| 150| 160| 170| 180| 190| 200| 210| 220| 230| 240| 250| 260| 270|
| 76   | 0  | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100| 110| 120| 130| 140| 150| 160| 170| 180| 190| 200| 210| 220| 230| 240| 250| 260| 270|
| 77   | 0  | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100| 110| 120| 130| 140| 150| 160| 170| 180| 190| 200| 210| 220| 230| 240| 250| 260| 270|
| 78   | 0  | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100| 110| 120| 130| 140| 150| 160| 170| 180| 190| 200| 210| 220| 230| 240| 250| 260| 270|
| 79   | 0  | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100| 110| 120| 130| 140| 150| 160| 170| 180| 190| 200| 210| 220| 230| 240| 250| 260| 270|

This Side is symmetric to the other one.
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

$$d_{k}^{T} = \frac{1}{N} \sum_{j=1}^{N} P_{kj} W_{kj} \quad \text{for } k = 1, \ldots, N$$

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,

\[ \text{Min } Z = \sum_{j=1}^{N} \sum_{i=1}^{M} \{P_i d_j x_{ij}\} \]  \hspace{1cm} (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

\[ x_{ij} = \begin{cases} 
1 & \text{if flight } i \text{ is assigned to gate } j \\
0 & \text{otherwise}
\end{cases} \]
Here, $X_{ij}$ is a binary variable. If, for instance, flight $i$ is not assigned to gate $j$, $x_{i3} = 0$ and the product term $P_i d_3$ 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,

$$d_j = .35d_{ja} + .5d_{jd} + .15d_{jt}$$

(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 businessmen 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

\[
\text{Min } Z = \sum_{i=1}^{N} \sum_{j=1}^{N} \gamma_{i} p_{i} d_{j} x_{i} \text{ (3.3)}
\]

where \( \gamma_{i} \) 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) The Constraints

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.

Constraints inherent to the assignment problem:

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

\[
\sum_{j=1}^{N} x_{ij} = 1 \quad \forall \ i = 1, \ldots, M \tag{3.4}
\]

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:

\[
L(i) = \{ \ell | t_{\ell}^a + t_{\ell}^g \geq t_{i}^a, \ell = 1, \ldots, i-1 \}
\]

\[
= \{ \ell | t_{\ell}^a + t_{\ell}^g \geq t_{i}^a, \ell \in L(i-1) \} \tag{3.5}
\]

where \(t_{\ell}^a\) = Arrival time for flight \(\ell\)
and
\(t_{\ell}^g\) = ground stay time of flight \(\ell\).

Note that \(t_{\ell}^a + t_{\ell}^g\) is actually the departing time for flight \(\ell\). 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 preceding 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:

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

Equation 3.6 says that if any flight 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 $([M-1] \times N)$ 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 $([M-1] \times N)$ total constraints. For the case of Toronto Terminal No. 2, the total number of constraints is

$$
([M-1] \times N) + M = (138 \times 20) + 138 = 2,878
$$

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

\[
\sum_{\ell \in L(p)} x_{\ell,j} + x_{p,j} = x_{p-3,j} + x_{p-2,j} + x_{p-1,j} + x_{p,j} \leq 1 \quad (3.7a)
\]

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

\[
\sum_{\ell \in L(p+1)} x_{\ell,j} + x_{p+1,j} = x_{p-3,j} + x_{p-2,j} + x_{p-1,j} + x_{p,j} + x_{p+1,j} \leq 1 \quad (3.7b)
\]

Here, \( L(p) \subseteq L(p+1) \) 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 \( L(i) \subseteq L(i+k) \), for any \( k=2,\ldots,M-i+1 \),
then \( L(i) \subseteq L(i+1) \subseteq \cdots \subseteq L(i+k) \)

Proof: Assume that \( L(i+r) \subseteq L(i+r+1) \) for some \( r = 0, \ldots, k-1 \). Then \( \exists f \) 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, \( t_{i+k}^a > t_{i+r+1}^a \) 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, \( L(3) \subseteq L(7) \), 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 $t$ are serving a large number of the same passengers, then the
If these flights are assigned to gates too distant, then the following can be done. Fix one of the flights, say flight $\ell$, to the gate assigned to it by the linear program, say gate $z$. Thus, fix $X_{\ell z} = 1$ and add the following constraint:

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

(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 $\ell$'s, or gate $z$. If, as a result of introducing this constraint, the value of the optimal solution is greatly increased (which also means 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 flight $r$ to its gate and attempt to bring flight $\ell$ 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 $t$ 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.

$$\sum_{j=1}^{N} \sum_{i=1}^{N} x_{ij} w_{ij} x_{rs} \leq D \quad (3.10)$$
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:

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

and

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

Each pair of subsets $I_s$ of $I$ and $J_s$ of $J$ can then be 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 $x_{73} = 0$, for all flights 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 $L(i), i = 1,\ldots,5$. 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 succinctly

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

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.
<table>
<thead>
<tr>
<th>GATE</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(d_j) (in feet)</td>
<td>1000</td>
<td>2400</td>
<td>3000</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>FLIGHT</th>
<th>ARRIVAL TIME</th>
<th>DEPARTURE TIME</th>
<th>PASSENGERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>00:00</td>
<td>00:25</td>
<td>400</td>
</tr>
<tr>
<td>2</td>
<td>00:10</td>
<td>00:40</td>
<td>200</td>
</tr>
<tr>
<td>3</td>
<td>00:20</td>
<td>00:50</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>00:30</td>
<td>00:44</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>00:45</td>
<td>00:100</td>
<td>250</td>
</tr>
</tbody>
</table>

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 \( L(3) \not\subseteq L(4) \).
The constraints generated by the third flight are not, therefore, redundant. Similarly, the fourth flight leaves before the fifth flight arrives and \( L(4) \not\subseteq L(5) \).

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

\[
\begin{align*}
\text{Min } Z &= \sum_{i=1}^{5} \sum_{j=1}^{3} d_{ij} p_i x_{ij} \\
\text{S.T.} \\
1\text{st Type of Constraints: } &\sum_{j} x_{ij} = 1 \quad \forall i \\
X_{11} + X_{12} + X_{13} &= 1 \\
X_{21} + X_{22} + X_{23} &= 1 \\
X_{31} + X_{32} + X_{33} &= 1 \\
X_{41} + X_{42} + X_{43} &= 1 \\
X_{51} + X_{52} + X_{53} &= 1
\end{align*}
\]
2nd Type of Constraints: \[ \sum_{\ell \in L(i)} X_{\ell j} + X_{ij} \leq 1 \forall i, j \]

- \( L(1) = \emptyset \)
- \( L(2) = \{1\} \)
- \( X_{11} + X_{21} \leq 1 \)
- \( X_{12} + X_{22} \leq 1 \)
- \( X_{13} + X_{23} \leq 1 \)

- \( L(3) = \{1, 2\} \)
- \( X_{11} + X_{21} + X_{31} \leq 1 \)
- \( +X_{12} + X_{22} + X_{32} \leq 1 \)
- \( +X_{13} + X_{23} + X_{33} \leq 1 \)

- \( L(4) = \{2, 3\} \)
- \( +X_{21} + X_{31} + X_{41} \leq 1 \)
- \( +X_{22} + X_{32} + X_{42} \leq 1 \)
- \( +X_{23} + X_{33} + X_{43} \leq 1 \)

- \( L(5) = \{3\} \)
- \( +X_{31} + X_{51} \leq 1 \)
- \( +X_{32} + X_{52} \leq 1 \)
- \( X_{33} + X_{53} \leq 1 \)

\( X_{ij} = 0, 1 \quad \forall i = 1, \ldots, 5 \quad j = 1, \ldots, 3 \)
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 is used in conjunction with the VM/CMS Operating System of the IBM 370 computer.
<table>
<thead>
<tr>
<th>FLIGHT</th>
<th>GATE</th>
<th>AVERAGE WALKING DISTANCE</th>
<th>OBJECTIVE FUNCTION TERM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1,000</td>
<td>400,000</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2,400</td>
<td>480,000</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3,000</td>
<td>300,000</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1,000</td>
<td>100,000</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1,000</td>
<td>250,000</td>
</tr>
</tbody>
</table>

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 $A$. 

-51-
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 $59 \times 20 = 1,180$ 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 Data-mat 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
| 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

<table>
<thead>
<tr>
<th>Percentile</th>
<th>25th</th>
<th>50th</th>
<th>75th</th>
<th>99th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>617</td>
<td>750</td>
<td>1,000</td>
<td>1,300</td>
</tr>
<tr>
<td>Algorithm</td>
<td>460</td>
<td>617</td>
<td>735</td>
<td>1,100</td>
</tr>
<tr>
<td>Linear Program</td>
<td>450</td>
<td>600</td>
<td>700</td>
<td>1,083</td>
</tr>
</tbody>
</table>

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
<table>
<thead>
<tr>
<th>Original</th>
<th>784</th>
<th>—</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>608</td>
<td>176</td>
<td>22%</td>
</tr>
<tr>
<td>Linear Program</td>
<td>582</td>
<td>202</td>
<td>26%</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Percentile</th>
<th>25th</th>
<th>50th</th>
<th>75th</th>
<th>99th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>540</td>
<td>765</td>
<td>1,000</td>
<td>1,300</td>
</tr>
<tr>
<td>Algorithm</td>
<td>517</td>
<td>567</td>
<td>743</td>
<td>1,200</td>
</tr>
<tr>
<td>Linear Program</td>
<td>507</td>
<td>540</td>
<td>700</td>
<td>1,200</td>
</tr>
</tbody>
</table>

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)

\[ d_j = .35d_j^a + .5d_j^d + .15d_j^t \]  \hspace{1cm} (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
<table>
<thead>
<tr>
<th>Mean Percentage Saving (Compared to Original)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>Original</td>
</tr>
<tr>
<td>Algorithm</td>
</tr>
<tr>
<td>Linear Program</td>
</tr>
</tbody>
</table>

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

![](image)

<table>
<thead>
<tr>
<th>Percentile</th>
<th>25th</th>
<th>50th</th>
<th>75th</th>
<th>99th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>483</td>
<td>720</td>
<td>1,000</td>
<td>1,400</td>
</tr>
<tr>
<td>Algorithm</td>
<td>335</td>
<td>467</td>
<td>636</td>
<td>1,173</td>
</tr>
<tr>
<td>Linear Program</td>
<td>220</td>
<td>433</td>
<td>583</td>
<td>1,167</td>
</tr>
</tbody>
</table>

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
<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Mean Difference</th>
<th>Percentage Difference (Compared to original)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>1045</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>Algorithm</td>
<td>1091</td>
<td>-46</td>
<td>-4%</td>
</tr>
<tr>
<td>Linear Program</td>
<td>1062</td>
<td>-17</td>
<td>-1%</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>25th</th>
<th>50th</th>
<th>75th</th>
<th>99th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>900</td>
<td>930</td>
<td>1,120</td>
<td>1,900</td>
</tr>
<tr>
<td>Algorithm</td>
<td>900</td>
<td>920</td>
<td>1,150</td>
<td>2,100</td>
</tr>
<tr>
<td>Linear Program</td>
<td>900</td>
<td>920</td>
<td>1,100</td>
<td>2,100</td>
</tr>
</tbody>
</table>

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 VSL 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 on purpose. Different costs can be obtained during different computer runs. The only certain conclusion that the reader
VS1 Resource

CPU Time  3.40 seconds @ $1.667/sec.  
Virtual Core  4.231 knet sec. @ $.00014/KNS  
Subtotal  
802 printer lines @ $1.55 per 1,000 lines  
Subtotal  
Adjustment for day shift and standard priority  

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

<table>
<thead>
<tr>
<th></th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Development</td>
<td>$150</td>
</tr>
<tr>
<td>Model Permanent File</td>
<td>$120</td>
</tr>
<tr>
<td>rewriting</td>
<td></td>
</tr>
<tr>
<td>Simplex Method</td>
<td>$210</td>
</tr>
<tr>
<td>TOTAL</td>
<td>$480</td>
</tr>
</tbody>
</table>

Table 4.5b Very Approximate Costs for Running the Linear Program

Cost
(No initial feasible basis)

Cost
(Algorithm's basis Used)

$150

$120

$210

$480

$150

$120

$40

$310
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 permanent 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 LID,
// PROFILE='DEFER', MEMORY=150K,
// TIME=(0,10)
// PASSWORD DJEBEL
// EXEC FTG1C1G,PRINT='PRINT'
//FORT.SYSIN DD *

C DECLARATIONS

C DIMENSION AC(10),IPLTNO(150),IAC(150),ILP(150),
1 IAABT(150),IDEPHT(150),ITRANS(150),ICAT(150),IGATE(25),
2 IGT(25),IFPA(25),IFT(25),IPWA(25),
3 IAGAT(150),ISGAT.E(150),IWLTA(150),ICGATE[150),
4 IFA(25),IFDA(25),IFTA(25),IFNAA(25)
C CONTINUE

C INPUT AIRCRAFT DATA
NAC=1
100 READ(5,110) AC(NAC),IPLTNO(NAC)
110 FORMAT(A4,I4)
IF(IPLTNO(NAC).NE.777) GO TO 120
NAC=NAC-1
GO TO 200
120 NAC=NAC+1
GO TO 100

C INPUT FLIGHT DATA
200 NFLT=1
C FORMAT & READ REP
205 READ(5,210) ISEQN,IPLTNO(NFLT),IAC(NFLT),ACTYPE,
1 IAABT(NFLT),IDEPHT(NFLT),ICAT(NFLT),IGATE(NFLT)
210 FORMAT(I4,I4,I3,A4,I5,I5I2,I3)
C FLD ADD
IFACT(NFLT)=ICAT(NFLT)+1
IF(IPLTNO(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(IAC(J).EQ.ACTYPE) IAC(NFLT)=J
220 CONTINUE
IF(IAC(NFLT).NE.0) GO TO 240
WRITE(6,230) IAC(NFLT),IPLTNO(NFLT)
230 FORMAT(' INCORRECT AIRCRAFT TYPE ON FLIGHT NUMBER',I4,I3,
1 ' FLIGHT IGNORED')
GO TO 205
240 ILP(NFLT)=65
ITRANS(NFLT)=30
FILE: ALGO  VS1JOB  A

MFLT=MFLT+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(13,6I5)
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
350 READ(5,350) (ITWALK(I,J), J=1,NGATE)
370 CONTINUE
380 DO 390 I=1,NGATE
DO 360 J=1,NGATE
ITWALK(I,J) = ITWALK(J,I)
390 CONTINUE
WRITE(6,394) (ITWALK(I,J), J=1,NGATE)
394 FORMAT(1X,2016)
C WRITE (6,457)
457 FORMAT(///,11X,'LARGEST COMM BEST SERVE')
C INITIALIZE GATE AVAILABILITY
400 DO 410 I=1,NGATE
410 IGTIME(1,I) = 0
C C
C C
DO 420 I=1,MFLT
JG=0
DO 213 K=1,NGATE
213 IF(IGATE(I).EQ.IGATE(K)) JG=K
IF(JG.EQ.0) WRITE(6,272) I
272 POEMAT(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

-74-
FILE: ALGPS VS1JOB A

IPT=ISEAT (IAC(I)) * 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.IAERT(I).LE.IDEPT(L)) GO TO 420
IF (IARRT(I).LE.IAERT(I).AND.IDEPT(I).GE.IDEPT(L)) GO TO 420
IF (IARRT(I).LE.IAERT(I).AND.IDEPT(I).GE.IDEPT(L)) GO TO 420
411 CONTINUE

C COMPUTE AVERAGE WALKING DISTANCE FOR GATE J
416 IDA=IWALK(J,IAC(I))
IDD=IWALK(J,(IAC(I)+3))
IDT=0
DO 412 K=1,NGATE
412 IDT=IDT+ITWALK(J,K)/NGATE
IPA=IDA*IPA
IPD=IDD*IPD
IPT=IDT*IPT
IDIS=(IPA+IPD+IPT)/(IPA+IPD+IPT)

C SELECT MINIMUM WALKING DISTANCE
IF (IDIST.GT.MINDIS) GO TO 420
NEARBY=J
ISGAT(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) IFLTNOCI)
430 FORMAT(' FLIGHT I, I4,' COULD NOT BE ASSIGNED TO ANY AVAILABLE GATE. ','
1 '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)
IGATE(I)=NEARBY
IDA=IWALK(NEARBY,IAC(I))
IDB=IWALK(JG,IAC(I))
IDU=IWALK(NEARBY,(IAC(I)+3))
IDAA=IWALK(JG,IAC(I)+3)
IDT=0
IDTA=0
DO 455 K=1,NGATE
IDTA=IDTA+ITWALK(JG,K)/NGATE
455 IDT=IDT+ITWALK(NEARBY,K)/NGATE
FILE: ALGO VS1JOB A

JDEPT=IDEPT(I)
IF (IDEPT(I).GT.2400) JDEPT=IDEPT(I)-2400
IDIFA=IDAA-IDA
IDIFD=IDDA-IDD
IDIFT=IDTA-IDT
RIFA=FLOAT(IDA)/FLOAT(IDAA)
RIFD=FLOAT(IDD)/FLOAT(IDDA)
RIFT=FLOAT(IDT)/FLOAT(IDTA)
WRITE(6,460) IFLTNO(I),AC(IAC(I)),IABRT(I),JDEPT,
1 IDIFA,IDAIDIFA,RIFA,
2 IDIFFIDIFD,EIFTD,
3 IDIFT,IDT,IDIFT,RIFT,
4 WRITE(6,460)
      1X,14,1X,14,2X,14,1X,14,2X,215,16,1F8.3,3X,215,16,1F8.3
      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)
      IFWA(IWA)=IWA/100
      IFWAA(IWA)=IFWAA(IWA)+IPA+IPD+IPT
      K4=IWA/100
      NWK=IFWAA(IWA)+IPA+IPD+IPT
      WRITE(6,510)
      510 FORMAT(/,1X,8I10)
STOP
END

//GO.SYSIN DD *
DC9 90
DS9 110
DC8 140
DS8 210
727 135
725 135
L10 262
747 302
777 777
67 857857 747 1545 1645 2 87

-76-
<table>
<thead>
<tr>
<th>File: ALGO VS1JOB A</th>
</tr>
</thead>
<tbody>
<tr>
<td>51 724725 72S 1300 1410 1.91</td>
</tr>
<tr>
<td>53 411414 72S 1305 1400 0.78</td>
</tr>
<tr>
<td>55 465454 72S 1310 1420 0.81</td>
</tr>
<tr>
<td>59 413616 72S 1405 1500 0.80</td>
</tr>
<tr>
<td>66 415418 72S 1505 1600 0.78</td>
</tr>
<tr>
<td>73 417420 72S 1605 1700 0.80</td>
</tr>
<tr>
<td>79 455460 72S 1645 1745 0.76</td>
</tr>
<tr>
<td>86 419422 72S 1705 1800 0.78</td>
</tr>
<tr>
<td>91 726729 72S 1750 1855 1.93</td>
</tr>
<tr>
<td>95 421424 72S 1810 1900 0.80</td>
</tr>
<tr>
<td>100 797797 72S 1835 1925 2.97</td>
</tr>
<tr>
<td>105 423426 72S 1910 2000 0.78</td>
</tr>
<tr>
<td>115 425428 72S 2020 2100 0.80</td>
</tr>
<tr>
<td>117 461469 72S 2040 2110 0.79</td>
</tr>
<tr>
<td>124 427427 72S 2105 2155 0.78</td>
</tr>
<tr>
<td>127 162162 72S 2115 2155 0.81</td>
</tr>
<tr>
<td>138 241241 72S 2130 2240 0.79</td>
</tr>
<tr>
<td>136 429429 72S 2205 2300 0.80</td>
</tr>
<tr>
<td>5 000701 D9S 0000 0700 1.67</td>
</tr>
<tr>
<td>9 000721 D9S 0000 0730 1.93</td>
</tr>
<tr>
<td>13 000741 D9S 0000 0755 0.73</td>
</tr>
<tr>
<td>29 000982 D9S 0000 0900 2.89</td>
</tr>
<tr>
<td>36 720705 D9S 0445 1050 1.69</td>
</tr>
<tr>
<td>8 612612 D9S 0730 0800 0.76</td>
</tr>
<tr>
<td>10 238107 D9S 0740 0930 0.72</td>
</tr>
<tr>
<td>14 700774 D9S 0800 0855 1.93</td>
</tr>
<tr>
<td>20 308308 D9S 0815 0845 0.73</td>
</tr>
<tr>
<td>21 362444 D9S 0815 0900 0.76</td>
</tr>
<tr>
<td>27 346365 D9S 0855 0945 0.75</td>
</tr>
<tr>
<td>32 342642 D9S 0930 1035 0.74</td>
</tr>
<tr>
<td>33 605600 D9S 0930 1050 0.76</td>
</tr>
<tr>
<td>41 625654 D9S 1025 1115 0.83</td>
</tr>
<tr>
<td>42 773770 D9S 1030 1230 1.93</td>
</tr>
<tr>
<td>45 704385 D9S 1140 1545 1.76</td>
</tr>
<tr>
<td>48 368315 D9S 1230 1320 0.79</td>
</tr>
<tr>
<td>49 102102 D9S 1240 1310 0.83</td>
</tr>
<tr>
<td>52 344349 D9S 1305 1405 0.74</td>
</tr>
<tr>
<td>54 777780 D9S 1305 1500 1.93</td>
</tr>
<tr>
<td>56 706709 D9S 1320 1410 1.89</td>
</tr>
<tr>
<td>65 647650 D9S 1450 1550 0.71</td>
</tr>
<tr>
<td>68 646666 D9S 1530 1605 0.74</td>
</tr>
<tr>
<td>70 351351 D9S 1555 1625 0.76</td>
</tr>
<tr>
<td>71 601658 D9S 1555 1650 0.79</td>
</tr>
<tr>
<td>72 779713 D9S 1600 1725 1.93</td>
</tr>
<tr>
<td>78 609446 D9S 1640 1750 0.72</td>
</tr>
<tr>
<td>80 710727 D9S 1645 1745 1.96</td>
</tr>
<tr>
<td>83 983794 D9S 1655 1805 2.97</td>
</tr>
<tr>
<td>81 649387 D9S 1655 1815 0.71</td>
</tr>
<tr>
<td>88 655655 D9S 1730 1800 0.75</td>
</tr>
<tr>
<td>96 604604 D9S 1810 1855 0.77</td>
</tr>
<tr>
<td>98 489233 D9S 1820 1925 0.72</td>
</tr>
<tr>
<td>99 382389 D9S 1830 1920 0.73</td>
</tr>
<tr>
<td>101 353353 D9S 1840 1915 0.74</td>
</tr>
<tr>
<td>103 163163 D9S 1845 1920 0.71</td>
</tr>
<tr>
<td>104 781786 D9S 1905 1955 1.95</td>
</tr>
<tr>
<td>FILE: ALGO</td>
</tr>
<tr>
<td>------------</td>
</tr>
<tr>
<td>112 653329 D9S 1955 2055 0 72</td>
</tr>
<tr>
<td>114 716719 D9S 2000 2120 1 93</td>
</tr>
<tr>
<td>119 330357 D9S 2015 2120 0 71</td>
</tr>
<tr>
<td>120 354331 D9S 2035 2130 0 76</td>
</tr>
<tr>
<td>122 394357 D9S 2015 2120 0 71</td>
</tr>
<tr>
<td>125 728309 D9S 2135 2315 1 85</td>
</tr>
<tr>
<td>128 748388 D9S 2140 2320 0 76</td>
</tr>
<tr>
<td>130 398397 D9S 2205 2255 0 73</td>
</tr>
<tr>
<td>134 334334 D9S 2305 2350 0 79</td>
</tr>
<tr>
<td>135 332332 D9S 2310 2400 0 74</td>
</tr>
<tr>
<td>139 396397 D9S 2205 2255 0 73</td>
</tr>
<tr>
<td>140 334334 D9S 2305 2350 0 79</td>
</tr>
<tr>
<td>142 332332 D9S 2310 2400 0 74</td>
</tr>
<tr>
<td>144 467467 D9S 2320 2400 0 75</td>
</tr>
<tr>
<td>145 789789 D9S 2345 2400 1 89</td>
</tr>
<tr>
<td>71 1287 2367 1727 1303 2261 1737</td>
</tr>
<tr>
<td>72 1269 2350 1710 1265 2244 1720</td>
</tr>
<tr>
<td>73 1285 2365 1725 1301 2259 1735</td>
</tr>
<tr>
<td>74 1106 2193 1553 1112 2087 1543</td>
</tr>
<tr>
<td>75 1102 2182 1542 1118 2076 1552</td>
</tr>
<tr>
<td>76 926 2013 1373 932 1407 1363</td>
</tr>
<tr>
<td>77 919 1929 1289 935 1623 1299</td>
</tr>
<tr>
<td>78 746 1833 1193 752 1727 1183</td>
</tr>
<tr>
<td>79 739 1749 1109 755 1643 1119</td>
</tr>
<tr>
<td>79 739 1749 1109 755 1643 1119</td>
</tr>
<tr>
<td>80 566 1670 1030 502 1564 1020</td>
</tr>
<tr>
<td>81 556 1566 926 572 1460 936</td>
</tr>
<tr>
<td>82 509 1343 703 349 1237 713</td>
</tr>
<tr>
<td>83 594 1068 428 434 962 438</td>
</tr>
<tr>
<td>84 855 807 347 695 701 177</td>
</tr>
<tr>
<td>85 1109 553 601 949 447 329</td>
</tr>
<tr>
<td>86 1363 299 855 1203 193 583</td>
</tr>
<tr>
<td>87 1662 598 1154 1502 492 882</td>
</tr>
<tr>
<td>88 1845 781 1337 1605 675 1065</td>
</tr>
<tr>
<td>89 510 418 828 350 312 668</td>
</tr>
<tr>
<td>90 957 418 828 797 312 568</td>
</tr>
</tbody>
</table>
FILE: ALGO  VS1JOB  A

10  20  310  270  530  420  720  610  910
30  300  230  540  430  730  620  920  810105012901570189021102380265029203190
310  200  510  400  700  590  890  780102012601540181020802350307033403610
110  200  330  220  500  330  690  93011701450172019902660253028003070
110  200  500  390  690  560  82010601340161018902150242026902960
110  190  300  220  490  730  9731250152017902060233026002870
110  190  490  380  620  8601140141016801950222024902760
110  190  300  560  7801060133016001870214024102680
300  190  430  670  950122014901760203023002570
110  350  590  870114014101680195022202490
240  480  760103013001570184021102380
240  520  79010601330160018702140
1280  550  8201090136016301900
270  540  810108013501620
270  540  81010801350
270  540  8101080
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:

<table>
<thead>
<tr>
<th>First Column</th>
<th>Heading</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Column</td>
<td>FLT</td>
<td>Flight number</td>
</tr>
<tr>
<td>Second &quot;</td>
<td>AC</td>
<td>Aircraft type</td>
</tr>
<tr>
<td>Third &quot;</td>
<td>ARR</td>
<td>Flight's arrival time</td>
</tr>
<tr>
<td>Fourth &quot;</td>
<td>DEP</td>
<td>Flight's departure time</td>
</tr>
<tr>
<td>Fifth &quot;</td>
<td>GTE</td>
<td>Original gate assignment</td>
</tr>
<tr>
<td>Sixth &quot;</td>
<td>KTE</td>
<td>Algorithm's gate assignment</td>
</tr>
<tr>
<td>Seventh &quot;</td>
<td>ACT</td>
<td>Walking distance under original assignment for arriving passengers.</td>
</tr>
<tr>
<td>Eighth &quot;</td>
<td>CAL</td>
<td>Walking distance under algorithm's assignment for departing passengers</td>
</tr>
<tr>
<td>Ninth &quot;</td>
<td>PIF</td>
<td>Difference in the walking distances listed in the two previous columns</td>
</tr>
<tr>
<td>Tenth &quot;</td>
<td>RAT</td>
<td>Ratio of the algorithm's walking distance to the original walking distance for the arriving passengers.</td>
</tr>
<tr>
<td>Content</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eleventh through fourteenth columns</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Same as 7th through 10th columns, but for departing passengers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fifteenth through nineteenth column</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Same as 7th through 10th columns, but for transfer passengers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>303</td>
<td>DC9</td>
<td>0</td>
</tr>
<tr>
<td>450</td>
<td>DC9</td>
<td>430</td>
</tr>
<tr>
<td>363</td>
<td>DC9</td>
<td>600</td>
</tr>
<tr>
<td>373</td>
<td>DC9</td>
<td>552</td>
</tr>
<tr>
<td>417</td>
<td>DC9</td>
<td>1110</td>
</tr>
<tr>
<td>317</td>
<td>DC9</td>
<td>1330</td>
</tr>
<tr>
<td>322</td>
<td>DC9</td>
<td>1535</td>
</tr>
<tr>
<td>383</td>
<td>DC9</td>
<td>1510</td>
</tr>
<tr>
<td>570</td>
<td>DC9</td>
<td>1441</td>
</tr>
<tr>
<td>527</td>
<td>DC9</td>
<td>1700</td>
</tr>
<tr>
<td>542</td>
<td>DC9</td>
<td>1710</td>
</tr>
<tr>
<td>456</td>
<td>DC9</td>
<td>1915</td>
</tr>
<tr>
<td>129</td>
<td>DC9</td>
<td>7123</td>
</tr>
<tr>
<td>351</td>
<td>DC9</td>
<td>1100</td>
</tr>
<tr>
<td>745</td>
<td>DC9</td>
<td>2120</td>
</tr>
<tr>
<td>469</td>
<td>DC9</td>
<td>2200</td>
</tr>
<tr>
<td>398</td>
<td>DC9</td>
<td>2300</td>
</tr>
</tbody>
</table>
APPENDIX C

PREPROCESSOR OR MODEL

GENERATING PROGRAM

(Written in DATAMAT)
FILE: FLIGHT DATA
RUN FILE

CONVERSATIONAL MONITOR SYSTEM

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 PARTITION GATES

* TABLES TO KEEP MAXIMAL CONFLICT SETS

* FORM M:MINDLPRT = '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:MINDLPRT(HEAD,11) = 'VOID'
  G:SETCOUNT(12,11) = 0
  M:MINCHAIN(Next,11) = 'NOTCHAIN'

* PROCESS FLIGHTS IN ORDER OF ARRIVAL

* NEWMODEL
  //NXTLUP
  M:NEXT = DUMMY
  E:NEXT = 1E20
  LOOP M:MINCHAIN(0,11) <NE> DUMMY
    IF M:MINCHAIN(Next,11) <EQ> 'NOTCHAIN'
      GOTO ENDMNX
  GOTO ENDMNX
  E:NEXT = G:PLANES[ARRIVAL,11]
  M:NEXT = G:PLANES(0,11)
  //ENDNX

-89-
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), '0000000')
N: TYPE = SHIFT (MASK (N: PLANE, '0000000'), 5)
N: GATETABL = M: TYPENAME (TBLNAME, 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, 11) = 1.
RHS UNITY, N: PLANE = 1
ROW WALKDIST, N: PLANE & G: N: GATETABL (0, 11) =
G: PLANES (CAPACITY, N: NEXT) & G: N: GATETABL (N: DISTNAME, 11)

* DETERMINE MEMBERSHIP OF NEXT FLIGHT IN CURRENT CONFLICT SET
* N: MIN = M: MIN DEPRT (HEAD, N: GATETABL)
IF N: MIN <NE> 'VOID', 1
GOTO ADDNXT

GOTO ADDNXT
IF G: PLANES (ARRIVAL, N: NEXT) <GT> G: PLANES (DEPART, 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), '0000000') & '0000000'
ROW N: CONFLICT & G: N: GATETABL (0, 11) <LETYPE>
RHS UNITY, N: CONFLICT & G: N: GATETABL (0, 11) = 1.
N: INDEX = N: MIN
//DOCNST
IF N: INDEX <NE> 'VOID', 1
GOTO NDCNST

COL MASK (G: PLANES (0, N: INDEX), '0000000') & G: N: GATETABL (0, 11),
N: CONFLICT & G: N: GATETABL (0, 11) = 1.
N: INDEX = M: MINCHAIN (NEXT, N: INDEX)
GOTO DOCNST

//NDCNST
* DELET NON-CONFLICTING FLIGHTS FROM CHAIN
//DELETE
N: INDEX = N: MIN
//DODEL
IF G: PLANES (DEPART, 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
FILE: FLIGHT DATARUN F

CONVERSATIONAL MONITOR SYSTEM

N:MIN = 'VOID'
GOTO ADDNXT
//ENDEL
N:MIN = N:INDEX
* ADD NEXT TO CHAIN FOR ITS TYPE, DEPARTURE-ORDERED *
//ADDNXT
N:INDEX = N:MIN
//DOCHAN
IF N:INDEX < #H> 'VOID',1
GOTO RCHAN
GOTO RCHAN
N:LAST = N:INDEX
N:INDEX = M:MINCHAIN(NEXT,N:INDEX)
GOTO WCHAN
//RCHAN
IF N:INDEX < NE> N:MIN,2
M:MINDEPRT(HEAD,N:GATETABL) = N:NEXT
GOTO RCHAN
M:MINCHAIN(NEXT,N:LAST) = N:NEXT
//RCHAN
M:MINCHAIN(NEXT,N:INDEX) = N:INDEX
IF G:SETCOUNT[MAXORDER,N:GATETABL] < GT> G:SETCOUNT[ORDER,N:GATETABL],1
GOTO NWLUP
//ENDLUP
*
* WRITE CONSTRAINTS FOR FINAL CONFLICT SETS *
*
LOOP M:MINDEPRT(0,11) < NE> DUMMY
M:GATETABL = M:MINDEPRT(0,11)
I:MAXORDER = G:SETCOUNT[MAXORDER,11]
DISPLAY N:GATETABL,1:MAXORDER
N:MIN = M:MINDEPRT(HEAD,11)
IF G:SETCOUNT[ORDER,N:GATETABL] < GT> 1,1
GOTO ENDCLR
N:CONFLICT = MASK [G:PLANES(0,N:MIN),'**********00') & '*' :
EOW N:CONFLICT & G:N:GATETABL(0,12) < LETYPE>
EWS UNITY, N:CONFLICT & G:N:GATETABL(0,12) = 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(0,12),
N:CONFLICT & G:N:GATETABL(0,12) = 1.
N:INDEX = M:MINCHAIN(NEXT,N:INDEX)
GOTO DCCNST
//NDCON
//ENDCLR
QUIT
FILE: FLIGHT DATARUN P

CONVERSATIONAL MONITOR SYSTEM

ENDATA
## APPENDIX D

**THE POSTPROCESSOR PROGRAM**

*(Written in DATAMAT)*

Listing of functions in the Postprocessor:

<table>
<thead>
<tr>
<th>Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>FINAL</td>
<td>Constructs a condensed table containing all flights and their LP assigned gate</td>
</tr>
<tr>
<td>MEAN</td>
<td>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 assignment 2) the heuristic algorithm and 3) the LP</td>
</tr>
<tr>
<td>ARRIVALS</td>
<td>Same as MEAN, but instead of listing the overall mean walking distance, it lists the expected walking distance for the arriving passengers.</td>
</tr>
<tr>
<td>DEPARTUR</td>
<td>Same as ARRIVALS, but for the departing passengers.</td>
</tr>
<tr>
<td>TRANSFER</td>
<td>Same as ARRIVALS, but for the transfer passengers.</td>
</tr>
<tr>
<td>HISTO</td>
<td>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.</td>
</tr>
<tr>
<td>Name</td>
<td>Purpose</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ARRHISTO</td>
<td>Same as HISTO, but the histogram is for distances of arriving passengers only.</td>
</tr>
<tr>
<td>DEPHISTO</td>
<td>Same as HISTO, but for the departing passenger.</td>
</tr>
<tr>
<td>TRPHISTO</td>
<td>Same as HISTO, but for the transfer passenger.</td>
</tr>
</tbody>
</table>
FILE: RESULTS DATAMAC A

*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, TRANSFER, INTERNATIONAL.
*G:TRANSDIS, THE TRANSPOSE OF G:GATEDIST
*G:GATES, WHICH CONTAINS THE WALKING DISTANCE FROM EACH GATE FOR EACH TYPE OF FLIGHT (DOM, TRANS, INT'L) AND FOR EACH TYPE OF PASSENGER (ARRIVING, DEPARTING, TRANSFER)
*AND M:TYPENAME.
*FINALLY, M:GATASSGN, 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, $DMODEL SET FOR GENERATED MODEL
* $DDRESULT, N:CASENAME SET FOR OPTIMAL SOLUTION
* $DDRESULT, N:CASENAME SET FOR OPTIMAL SOLUTION
* REFORM M:PAIRINGS = COLS
* FORM LIST OF ACTIVE PAIRINGS
I:ASSIGNED = 0
LOOP M:PAIRINGS(11,0) <NE> DUMMY
   IF X:(M:PAIRINGS(11,0),N:CASENAME) <EQ> 0., 2
      I:ASSIGNED = I:ASSIGNED + 1
      STUB M:ASSIGNED(I:ASSIGNED) = M:PAIRINGS(11,0)
      CONTINUE
   STUB M:FLIGHTS = MASK(M:ASSIGNED(11,0),'*****00')
   FORM M:GATEASGN = M:FLIGHTS(STUB),GATE
   M:GATASSGN(11,GATE) = MASK(M:ASSIGNED(11,0),'000000**')
   DISPLAY M:GATASGN
ENDATA
NAME MEAN TABLE M:SPLK=ORGATE,ORWD,ALGOGATE,ALGOWD,LPGAATE,LPWD,PAX
   ... STUB M:LO=MASK (M:GATASSGN(11,0),'*****00')
   FORM G:COMPARE=M:LO(STUB),M:SPLK
   G:COMPARE(11,LPWD)=G:TRANSDIS(M:GATASSGN(11,GATE),'M:TYPENAME(DISTNAME)
   M:GATASSGN(11,0),'000000**'))
   G:COMPARE("1",GATE)=G:ALGOTES("1",GATE)
   G:COMPARE("1",ALGOGATE)=G:ALGOTES("1",ALGOGATE)
   FORM M:ALGOTES=G:ALGOTES(STUB),G:ALGOTES(HEAD)
   M:ALGOTES(11,12)=G:ALGOTES(11,12)
   FORM G:COMPARE=G:COMPARE(STUB),G:COMPARE(HEAD)
   M:COMPARE(11,12)=G:COMPARE(11,12)
   G:COMPARE("1",PAX)=G:ALGOTES("1",PAX)
   M:COMPARE("1",PAX)=G:COMPARE("1",PAX)
   M:COMPARE(11,PAX)=M:CASRENPE[11,PAX],"000000**")
   M:COMPARE(11,LPWD)=M:GATASSGN(11,GATE)
   M:COMPARE(11,ALGOGATE)=M:COMPARE(11,ALGOGATE),"000000**")
   M:COMPARE(11,ORWD)=M:COMPARE(11,ORWD),"000000**")

-95-
FILE: RESULTS DATAMAC A  CONVERSATIONAL MONITOR SYSTEM

NAME: ARRIVALS
TABLE M:A=DOM(1,TRAB,INT)
***
FORM G:ARRIVALS=M:LO(STUB), M:SPEC(HEAD)
M:TYPENAME(DISTNAME,MASK(M:GATESGN[11,0], "00000*00"))
M:TYPENAME(DISTNAME,MASK(M:GATESGN[11,0], "00000*00"))
DISPLAY M:ARRIVALS
ENDATA
NAME: DEPARTURE
TABLE M:SPEK=OGATE,OPWD,ALGOGATE,ALOGATE,LPGATE,LPWD,PAX
***
STUB M:LO=MASK (M:GATESGN[11,0], "00000***")
FORM M:DEPARTURES=M:LO(STUB), M:SPEC(HEAD)
DISPLAY M:DEPARTURES
ENDATA
NAME: DEPARTURE
TABLE M:SPEK=OGATE,OPWD,ALGOGATE,ALOGATE,LPGATE,LPWD,PAX
***
STUB M:LO=MASK (M:GATESGN[11,0], "00000***")
FORM M:DEPARTURES=M:LO(STUB), M:SPEC(HEAD)
TABLE M:D=DOM(1,TRAB,INT)
***
FORM M:DEPARTURES=M:LO(STUB), M:SPEC(HEAD)
FILE: RESULTS DATAMAC A

CONVERSATIONAL MONITOR SYSTEM

FORM M:ALGOTES=G:ALGOTES [STUB], G:ALGOTES [HEAD]
M:ALGOTES[1,12] = G:ALGOTES(1,12)
FORM M:DEPARTURE=G:DEPARTURE [STUB], G:DEPARTURE [HEAD]
M:DEPARTURE[1,12] = G:DEPARTURE(1,12)
M:DEPARTURE[11,PAK] = MASK (M:DEPARTURE(11,PAK), '0000000000')
M:DEPARTURE[11,LPGATE] = MASK (M:DEPARTURE(11,ALGATE), '0000000000')
M:DEPARTURE[11,LOCATE] = MASK (M:DEPARTURE(11,ORGATE), '0000000000')
M:DEPARTURE[11,ALGOWD] = MASK (M:DEPARTURE(11,ALGOWD), '0000000000')
M:DEPARTURE[11,OLGW] = MASK (M:DEPARTURE(11,OLGW), '0000000000')
DISPLAY M:DEPARTURE
ENDATA
NAME TRANSFER
TABLE M:SPEK=OGEATE,OF,PE,ALGATE,ALGOWD,LPGATE,LPWD,PAK
***
SUB M:LO=MASK (M:GATEASGN(11,0), '0000000000')
FORM M:TRANSFER=M:LO [STUB], M:SPEK [HEAD]
M:TRANSFER["1,ALGATE]=G:ALGOTES["1,ALGATE]
M:TRANSFER["1,ORGATE]=G:ALGOTES["1,ORGATE]
M:TRANSFER["1,PAK]=G:ALGOTES["1,PAK]
M:TRANSFER[11,PAK] = MASK (M:TRANSFER(11,PAK), '0000000000')
M:TRANSFER[11,LPGATE] = MASK (M:TRANSFER(11,LPGATE), '0000000000')
M:TRANSFER[11,LOCATE] = MASK (M:TRANSFER(11,LOCATE), '0000000000')
M:TRANSFER[11,OLGW] = MASK (M:TRANSFER(11,OLGW), '0000000000')
M:TRANSFER[11,LPWD] = MASK (M:TRANSFER(11,LPWD), '0000000000')
DISPLAY M:TRANSFER
ENDATA
NAME HISTO
TABLE M:SO=06,ALGO,LP
***
FORM M:HISTO=G:PR [HEAD], M:SO [HEAD]
LOOP G:PR[0,1] < NL > EU
E:LL=G:FE [NUM, 11]
LOOP G:COMPARE[12,OLGW] < LT > (100*E:LL+1)
IF G:COMPARE[12,OLGW] < LT > (100*E:LL), G:HISTO(11,OR) = G:COMPARE[12,PAK]
CONTINUE
LOOP G:COMPARE[12,ALGOWD] < LT > (100*E:LL+1)
FILE: RESULTS DATAMAC A
CONVERSATIONAL MONITOR SYSTEM

E:ALGOTOT=E:ALGOTOT+[G:COMPARE(12,ALGOWD)*G:COMPARE(12,PAX)]
CONTINUE
LOOP G:COMPARE(12,LPWD) <LT> (100*(E:LL+1))
IF G:COMPARE(12,LPWD) <LT> (100*E:LL), 2
G:ALGOAVG=G:ALGOAVG+[G:COMPARE(12,PAX)]
E:LPTOT=E:LPTOT+[G:COMPARE(12,LPWD)*G:COMPARE(12,PAX)]
CONTINUE
E:SUM=E:SUM+G:HISTO(11,OB)
CONTINUE
DISPLAY G:HISTO
DISPLAY E:SUM
E:ORAVG=E:ORAVG/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(11,12)=G:HISTO(11,12)/E:SUM
G:PERCENT(11,12)=G:PERCENT(11,12) * 100
DISPLAY G:PERCENT
TABLE G:SUNRY=ORAVG,ALGOAVG,LPAVG
NAME
DEPHISTO
FORM G:DEPARTUR=M:DEPARTUR (STUB),M:DEPARTUR (HEAD)
G:DEPARTUR(11,OWD)=G:DEP (M:DEPARTUR(11,ORWD),
M:TYPE NAME (DISTNAME, MASK(M:DATEASGN(11,0),*00000*00*))
G:DEPARTUR(11,ALGOWD)=G:DEP (M:DEPARTUR(11,ALGOWD),
M:TYPE NAME (DISTNAME, MASK(M:DATEASGN(11,0),*00000*00*))
G:DEPARTUR(11,LPWD)=G:DEP (M:DEPARTUR(11,LPWD),
M:TYPE NAME (DISTNAME, MASK(M:DATEASGN(11,0),*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(0,11) <NE> EU
E:LL=G:PR (NUM,1)
LOOP G:DEPARTUR(12,OWD) <LT> (100*(E:LL+1))
IF G:DEPARTUR(12,OWD) <LT> (100*E:LL), 2
G:DEPHISTO(11,OR)=G:DEPHISTO(11,OR)+G:DEPARTUR(12,PAX)
E:ORAVG=E:ORAVG+[G:DEPARTUR(12,OWD)*G:DEPARTUR(12,PAX)]
CONTINUE
LOOP G:DEPARTUR(12,ALGOWD) <LT> (100*(E:LL+1))
IF G:DEPARTUR(12,ALGOWD) <LT> (100*E:LL), 2
G:DEPHISTO(11,ALGO)=G:DEPHISTO(11,ALGO)+G:DEPARTUR(12,PAX)
E:ALGOAVG=E:ALGOAVG+[G:DEPARTUR(12,ALGOWD)*G:DEPARTUR(12,PAX)]
CONTINUE
LOOP G:DEPARTUR(12,LPWD) <LT> (100*(E:LL+1))
IF G:DEPARTUR(12,LPWD) <LT> (100*E:LL), 2
G:DEPHISTO(11,LP)=G:DEPHISTO(11,LP)+G:DEPARTUR(12,PAX)
E:LPTOT=E:LPTOT+[G:DEPARTUR(12,LPWD)*G:DEPARTUR(12,PAX)]
CONTINUE
E:SUNRY=E:SUNRY+G:DEPHISTO(11,OR)
FILE: RESULTS DATAMAC A

CONVERSATIONAL MONITOR SYSTEM

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(11,1,4)=G:DEPHISTO(11,12)/E:SUMD
G:PERCENT(11,12)=G:PERCENT(11,12)*100
DISPLAY G:PERCENT
TABLE G:SUMRYD=ORAVGD,ALGOAVGD,LPAVGD
NAME=E:ORAVGD,E:ALGOAVGD,E:LPAVGD
***
ENDATA
NAME
ARRIVALS
FORM G:ARRIVALS=M:ARRIVALS(STUB),M:ARRIVALS(HEAD)
G:ARRIVALS(11,0,OM)=G:ARRIVALS(11,0,GATE)
M:TYPE=NAME(DISTNAME,M:GATEASGN(11,0),'-0000000*'))
G:ARRIVALS(11,1,ALGOWD)=G:ARRIVALS(11,1,GATE)
M:TYPE=NAME(DISTNAME,M:GATEASGN(11,0),'-0000000*'))
G:ARRIVALS(11,1,LPWD)=G:ARRIVALS(11,1,PAT)
M:TYPE=NAME(DISTNAME,M:GATEASGN(11,0),'-0000000*'))
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(0,11)<NE>RU
E:LL=G:PR[NUN,11]
LOOP G:ARRIVALS(12,0,OM)<LT> (100*(E:LL+1))
IF G:ARRIVALS(12,0,OM)<LT> (100*(E:LL+1),2
G:ARRIVALS(11,0,OM)=G:ARRIVALS(11,0,OM)+G:ARRIVALS(12,PAX)
E:ORTOTA=E:ORTOTA+G:ARRIVALS(12,0,OM)*G:ARRIVALS(12,PAX)
CONTINUE
LOOP G:ARRIVALS(12,1,ALGOWD)<LT> (100*(E:LL+1))
IF G:ARRIVALS(12,1,ALGOWD)<LT> (100*(E:LL+1),2
G:ARRIVALS(11,1,ALGO)=G:ARRIVALS(11,1,ALGO)+G:ARRIVALS(12,PAX)
E:ALGOTOTA=E:ALGOTOTA+G:ARRIVALS(12,1,ALGOWD)*G:ARRIVALS(12,PAX)
CONTINUE
LOOP G:ARRIVALS(12,1,LPWD)<LT> (100*(E:LL+1))
IF G:ARRIVALS(12,1,LPWD)<LT> (100*(E:LL+1),2
G:ARRIVALS(11,1,LP)=G:ARRIVALS(11,1,LP)+G:ARRIVALS(12,PAX)
E:LPTOTA=E:LPTOTA+G:ARRIVALS(12,1,LPWD)*G:ARRIVALS(12,PAX)
CONTINUE
E:SUMA=E:SUMA+G:ARRIVALS(11,OM)
CONTINUE
DISPLAY G:ARRHISTO
DISPLAY L:SUMA
E:ORAVGD=E:ORIVD/E:SUMA
E:ALGOAVG=E:ALGOV/LSUMA
E:LPAVGD=E:LPTOTD/E:SUMA
DISPLAY E:ORAVGD
DISPLAY E:ALGOAVGD
FILE: RESULTS DATAC A CONVERSATIONAL MONITOR SYSTEM

DISPLAY E:LPAVGA
FORM G:PERCENT=G:TFPHISTO(STUB), G:TFPHISTO(HEAD)
G:PERCENT(11,12)=G:ARRHISTO(11,12)/E:SUMT
G:PERCENT(11,12)=G:PERCENT{(11,12)*100
DISPLAY G:PERCENT
TABL G:SUMRY=E:LAVGA, ALGOAVGA, LPAVGA
MEMES=E:OAVGA, E:ALGOAVGA, E:LPAVGA

ENDATA
NAME TEPHISTO
FORM G:TRANSFER=M:TRANSFLR (STUB) , M:TRANSFER (HEAD)
G:TRANSFER{11,LPWD}=G:GATES{11,GATE}{11,GATE,TRANS}
G:TRANSFER{11,LPWD}=G:ALGOTE{11,TRA}
G:TRANSFER{11,LPWD}=G:GATES{11,ALGATE}{11,ALGO,TRANS}
G:TRANSFER{11,LPWD}=G:GATES{11,TRANS}{11,TRANS}
TABLE M:SO=O:ALGO, LP

FORM G:TEPHISTO=G:TF (HEAD), M:SO (HEAD)
LOOP G:PE(0,11) <NE> RU
  E:LL=G:PF (NUM,11)
  LOOP G:TRANSFER{12,LPWD} <LT> (100*E:LL+1)
    IF G:TRANSFER{12,LPWD} <LT> (100*E:LL), 2
      E:ORAVGT=E:ORAVGT + G:TRANSFER{12,PAZ}
    CONTINUE
    LOOP G:TRANSFER{12,LPWD} <LT> (100*E:LL+1)
      IF G:TRANSFER{12,LPWD} <LT> (100*E:LL), 2
        E:ALGOAVGT=E:ALGOAVGT + G:TRANSFER{12,PAZ}
      CONTINUE
    LOOP G:TRANSFER{12,LPWD} <LT> (100*E:LL+1)
      IF G:TRANSFER{12,LPWD} <LT> (100*E:LL), 2
        E:LPAVGT=E:LPAVGT + G:TRANSFER{12,PAZ}
      CONTINUE
    E:SUMT=E:SUMT + G:TRANSFER{11,OR}
    CONTINUE
    DISPLAY G:TFPHISTO
    DISPLAY E:SUMT
    E:ORAVGT=E:ORAVGT/E:SUMT
    E:ALGOAVGT=E:ALGOAVGT/E:SUMT
    E:LPAVGT=E:LPAVGT/E:SUMT
    DISPLAY E:AVG
    DISPLAY E:ALGOAVGT
    DISPLAY E:LPAVGT
FORM G:PERCENT=G:TFPHISTO(STUB), G:TFPHISTO(HEAD)
G:PERCENT(11,12)=G:TFPHISTO(11,12)/E:SUMT
G:PERCENT(11,12)=G:PERCENT{(11,12)*100
DISPLAY G:PERCENT
TABL G:SUMRY=E:AVGA, ALGOAVGA, LPAVGA
MEMES=E:OAVGA, E:ALGOAVGA, E:LPAVGA

ENDATA

-100-
APPENDIX E
OUTPUT OF THE POSTPROCESSOR PROGRAM

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

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORGATE</td>
<td>Gate originally assigned to the flight by Air Canada.</td>
</tr>
<tr>
<td>ORWD</td>
<td>Expected walking distance for a passenger in the flight according to the original assignment.</td>
</tr>
<tr>
<td>ALGOGATE</td>
<td>Gate assigned to the flight by the heuristic algorithm.</td>
</tr>
<tr>
<td>ALGOWD</td>
<td>Expected walking distance for a passenger in the flight according to the heuristic algorithm's assignment.</td>
</tr>
<tr>
<td>LGATE</td>
<td>Gate assigned to the flight by the linear program</td>
</tr>
<tr>
<td>LPWD</td>
<td>Expected walking distance for a passenger in the flight according to the linear program's assignment.</td>
</tr>
</tbody>
</table>
$H: COMPARE$  
or gate  
or wd  
alg gate  
alg wd  
lfgate  
lfwd  
fax  
F608  = 71  
F243  = 75  
F305  = 81  
F210  = 76  
F310  = 76  
F920  = 91  
F440  = 79  
F400  = 80  
F402  = 78  
F701  = 87  
F721  = 93  
F341  = 73  
F982  = 89  
F361  = 74  
F442  = 85  
F303  = 71  
F727  = 00  
F705  = 89  
F164  = 79  
F612  = 76  
F107  = 72  
F796  = 95  
F774  = 93  
F960  = 87  
F404  = 80  
F308  = 73  
F444  = 76  
F791  = 91  
F117  = 77  
F103  = 71  
F450  = 79  
F992  = 85  
F365  = 75  
F363  = 73  
F406  = 78  
F292  = 91  
F642  = 74  
F600  = 76  
F123  = 71  
F373  = 71  
F246  = 81  
F408  = 80  
F371  = 72  
F654  = 83  
F778  = 93  
F410  = 78  
F385  = 76  
F412  = 80  
F347  = 75  
F315  = 79  
F249  = 77  
F102  = 83  
F725  = 91  
F414  = 78  
F349  = 74  

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.
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.
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.
| 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 | 0867 | 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 | 024 |
| 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.

-105-
<table>
<thead>
<tr>
<th>:PERMEAN</th>
<th>OR</th>
<th>ALGO</th>
<th>LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>10.288305</td>
<td>15.376247</td>
<td>16.522212</td>
</tr>
<tr>
<td>400</td>
<td>4.8921124</td>
<td>13.747960</td>
<td>15.194923</td>
</tr>
<tr>
<td>500</td>
<td>8.0108794</td>
<td>16.177697</td>
<td>17.106074</td>
</tr>
<tr>
<td>600</td>
<td>18.701723</td>
<td>24.315503</td>
<td>26.480508</td>
</tr>
<tr>
<td>700</td>
<td>15.318223</td>
<td>13.773345</td>
<td>11.963735</td>
</tr>
<tr>
<td>800</td>
<td>3.6264733</td>
<td>4.2792384</td>
<td>1.9655485</td>
</tr>
<tr>
<td>900</td>
<td>14.531278</td>
<td>6.7198549</td>
<td>7.0208522</td>
</tr>
<tr>
<td>1000</td>
<td>6.5457842</td>
<td>4.2611061</td>
<td>2.8141432</td>
</tr>
<tr>
<td>1100</td>
<td>4.4786945</td>
<td>.83408885</td>
<td>.41704442</td>
</tr>
<tr>
<td>1200</td>
<td>13.091568</td>
<td></td>
<td>.51495920</td>
</tr>
<tr>
<td>1300</td>
<td></td>
<td>.51495920</td>
<td></td>
</tr>
<tr>
<td>1400</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1500</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1700</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1800</td>
<td>.51495920</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1900</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2200</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2300</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2400</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table E.5 Statistical Distribution of the Overall Mean Walking Distance (used to draw Fig. 4.1)
Table E.6  Statistical Distribution of the Mean Walking Distance for an Arriving Passenger (used to draw Fig. 4.2)
<table>
<thead>
<tr>
<th>G:PERTRANS</th>
<th>OR</th>
<th>ALGO</th>
<th>LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>200</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>300</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>400</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>500</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>600</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>700</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>800</td>
<td>28.198879, 35.413112, 37.728491</td>
<td></td>
<td></td>
</tr>
<tr>
<td>900</td>
<td>31.026078, 32.366561, 32.561540</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>10.041433, 2.5834755, 2.5834755</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1100</td>
<td>9.9683159, 6.1174750, 6.1174750</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1200</td>
<td>12.308067, 7.2142335, 7.3848404</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1300</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>1400</td>
<td>4.2164270, 2.8271996, 2.3153790</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1500</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>1600</td>
<td>1.1455033, 1.6573239, 1.4867170</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1700</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>1800</td>
<td>1.1455033, 6.8486473, 6.7999025</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1900</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>2000</td>
<td>1.9497928, 4.9719717, 3.0221789</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2100</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>2200</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>2300</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>2400</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

Fig. E.7 Statistical Distribution of the Expected Walking Distance for a Departing Passenger (used to draw Fig. 4.3)
Table E.8 Statistical Distribution of the Expected Walking Distance for a Transfer Passenger (used to draw Fig. 4.4)
REFERENCES


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


