SCHEDULING OF AIRCRAFT CREWS
OF A COMMERCIAL AIRLINE

HANEEF IQBAL MALIK
B.Sc.(Hons.),M.A.

A thesis submitted to the
UNIVERSITY OF ASTON IN BIRMINGHAM
DEPARTMENT OF MATHEMATICS

for the degree of
MASTER OF SCIENCE

JUNE 1970

ACKNOWLEDGEMENT

I wish to acknowledge my thanks to the following persons who assisted me in various ways during my studies regarding this Project.

Dr.A.M.Rajput, my Supervisor, who originally introduced me to this problem which he came across a year earlier when he visited P.I.A. Headquarters in Karachi. I am greatly indebted to him for his keen interest, guidance and help throughout the period of my studies.

Pakistan International Airline for providing me with all the relevant data for this Project, and the interest they have shown during the last one and a half years. I am especially grateful to the Director of Planning and the Chief Scheduling Officer of P.I.A. for their keen interest in the problem.

The Government of Pakistan, who awarded me a Scholarship for an M.Sc.,Course.

The programming and operating staff of the University of Aston Computer Centre, who have helped me from time to time in computer programming.

Finally, I wish to dedicate this Dissertation to my parents.

CONTENTS

Page No.

CHAPTER VII

Integer Programming

CHAPTER VIII

Solution of Layover time Matrices and Selection of Bases.

CHAPTER IX

OR Time Table

CHAPTER X

Monthly Assignment

Page No.

CHAPTER XI

Estimation of Reserve Crews.

APPENDIX

S U M M A R Y

This dissertation consists of a theoretical background of scheduling of aircraft crews of a commercial airline. Linear programming technique is applied to complete this dissertation.

This dissertation consists of three sections.

SECTION I.

i) Creating a time matrix (daily and weekly) by the use of an electronic computer considering all possible flight combinations of arrivals and departures of the crews at each slip station.

ii) Then all the time matrices are modified in the light of rules and regulations. These rules are framed by the Airline Authority and the crews' association.

SECTION II.

iii) Solving these matrices by the use of an electronic computer using the Linear Programming techniques, to minimize the layover time of the crews at a place other than the home base.

SECTION III.

In this Section various related topics are discussed such as

iv) Monthly assignment of the crew by a mathematical technique and number of crews required for a given schedule.

v) OR time table and its schedule.

vi) Estimation of reserve utilization of the crews.

The Appendix is devoted to the programmes which were used in completing this dissertation and which are written in I.C.L. 1900 Algol. and Fortran.

SECTION I

CHAPTER I.

1.1) Introduction.

Most of the commercial airlines of the World are investigating an adequate method of optimizing the utility of their crews. I have approached several airlines and have discussed this problem with them. As far as I know the scheduling of the crews in most of the airlines is done manually using no scientific method. Unfortunately, little has been published on this topic, thus no clue was available to use as to how the problem should be tackled.

Pakistan International Airline (P.I.A) wished that someone should investigate the possibility of improving the utilization of their crews. To understand the problem it is necessary to look into the present system of aircrews' scheduling. At present it is done manually by an experienced scheduler. He frames the schedule based on his past experience considering certain legal requirements, working knowledge of the routes etc., Then this schedule is submitted for approval to a committee consisting of the officials of the company and the representative of the crews' association. Such a schedule is usually expensive to the company in the following ways:-

1) At the slipstation (other than the home base) where the company is responsible for the crew's meals and accommodation until next assignment.
2) Wasteful of the crew's time.

In addition to these points, the human problem is also involved that the crews may be unhappy in spending

1.1) contd.

and wasting their time away from their families and home.

1.2) Limitations.

For both men and machine there are certain limits beyond which it is unwise to pursue continuing application to a particular task without a break for relaxation or overhaul, because both men and machine have a capacity limited by safety requirements. More specifically in the case of aircrew it is laid down in the regulation that they must not fly more than eight hours in 24 consecutive hours. They must also take an adequate rest before they take up their assignment. Thus the crews are given rest periods at or before the termination of eight scheduled hours of duty aloft, 30 hours in a week, 70 hours in a month and 700 hours in a year. These limitations vary from airline to airline.

In the shorter route where the flight time of the return trip is less than or equal to 8 hours, the same crew can go and come back, but in the case of longer routes, for the sake of safety, the Airline Authority have to replace the crew before exceeding the limit of 8 hours somewhere along the route. At these slipstations, the company is responsible for the crews accommodation and meals until their next assignment. The stay at the slipstation is determined by the flying time from the previous station or base to the slipstation. To reduce this expenditure or layover time of the crews, most of the airlines are investigating a scientific method of utilizing their crews and thereby reducing the layover time at the slip stations. Minimizing this time has the dual advantage of

1.2) contd.

making more crews available at the home base and of reducing the expenditure of the airline for meals and lodgings of the crews away from the home base. Significant reduction in the layover time at slipstations should correspond to some reduction in the number of required crews.

1.2.1) Technique used. I have used a Linear Programming technique to automate the scheduling of aircrews of P.I.A. and a net saving of 570 hours per week is shown in layover time of the crews at slipstations other than the home bases. Consequently, this would mean that the number of crews can be reduced from 54 to 43 (Chapter X).

A system is also developed to eliminate the possibilities of biased allocation of crews to flights. This is presented in Chapter X. In this method OR schedule is circulated among the crews and their preference bids are taken. Then a mathematical technique is applied to allocate duties to the crews, which eliminate the human bias.

In Chapter IX, OR time table is developed by deviating 25 minutes in Karachi-Dacca sector to certain flights. A further reduction of 119-30 hours to the layover time of the crews is shown.

In Chapter XI, regression analysis is used to estimate the reserve crews for regular and emergency flights.

1.3) Crew Scheduling Problem.

The crew scheduling problem is an operational research classic. It is mathematically identical to"travelling salesman problem in certain variation.

"Definition of Crew Scheduling problem".

1.3) contd.

The scheduled flight path of a commercial aircraft, often referred to as an "aircraft routing", is the single most important element of air transport industry planning and operations. This flight path is determined primarly by marketing considerations and is limited by practical considerations of maintenance, airport facilities etc.,

The aircraft routing is considered as fixed, as a basis of computation of flight crew schedules. At this point an entirely different set of considerations applies. Flight crew schedules are carefully defined (which are not identical with any other airline) and severely limited by the Government, the Company, consideration of safety, operability, overall profit and the necessity of good working conditions at the same time.

In certain circumstances it is possible to express the outcome of the crew scheduling in the form of the sequences A - a - b - c - B, where the end terms signify two special events and the inner terms must be carried out in due sequence. The two end events signify departure from and arrival at the crew base, the intermediate events signify successive sectors flown. It is not possible for the same crew to finish the sequence describe, because of the safety requirements. Some rules and regulations have to be obeyed. Every airline of the World has a problem of optimum crew utilization because the crews are highly paid and the Company has to spend a large amount on their accommodation at the slipstations. This gives the impression that the optimum crew utilization means simply,

1.3) contd.

maximum number of hours aloft.

An ideal crew assignment or 'trip pairing' may be defined as follows:-

"The crew arrives at City A at 9 o'clock in the morning and spends a busy number of hours preparing a flight plan, checking equipment and other formalities. At 10 o' clock their plane departs for City B arriving on schedule at 13-45. At exactly 14-45 the crew takes a return flight out for City A, arriving on schedule 18-30 o' clock. Debriefing requires half an hour, at 19-00 o'clock, the crew leaves for their home." They have been on duty 10 hours which is maximum duty time for a crew.

Unfortunately, very few actual trip pairing approach this ideal situation. The aircraft schedules are designed under the major influence of marketing consideration, ideal crew pairing are often simply coincidental.

In practice, the aircraft routing is usually broken down to individual flights. These are re-assembled by trial and error methods into a set of crew pairing, governed by many factors which are often conflicting. The 'ideal" trip described above might even disappear.

If a crew could board an aircraft at a point of origin and stay with it until it returns to base, then there is no problem. But this ideal situation is not possible or practical for numerous reasons, and thus the crew scheduling problem arises. The nature of the problem is not always the same for every airline, as the flying rules vary from airline to airline.

1.4) "<u>Nature of the Schedule</u>".

The Government and Crew Association of all the World airlines are in unanimous agreement on one point - "safety". Any trip pairing which gives excessive crew fatigue without satisfactory rest is automatically not considered.

Immediately beyond this area which is not easy to define, things become hazy. The first complication is the difference between scheduled and actual flying. Generally speaking, the legality of a scheduled pairing is re-examined at the conclusion of each flight segment. For many reasons the flights are late or irregular. A crew flying perfectly valid trip sequence may find themselves circling the airport for an hour or two and upon landing it is illegal to take out their scheduled flight the next day. Such possibilities must be considered while making the schedule.

In this general era of safety, some factors are hard to define. If a liberal schedule is framed, then a huge amount of foreign exchange is used on their accommodation and meals. If, on the other hand, a tight schedule is framed, then operating difficulties will occur.

I have used Linear Programming techniques for framing an optimum P.I.A. Schedule taking all possible factors into account.

1.5) <u>Highlights</u>.

The crew scheduling problem is essentially the same for all the airlines, by virtue of the fact that they operate similar equipment, over similar routes, by similarly trained

1.5) contd.

pilots, who have similar relationships with their respective airlines. But the technique of assigning the duty is different because the flying rules are different from airline to airline. In some airlines the schedule is released to all crews and their preference bid is taken. Normally the crew members with highest sneiority have first opportunity to select a desired flight schedule. The airline having a big fleet have solved their problem by making more than one base in order to minimize the layover time at slip stations.

1.6) <u>Computer Programs</u>.

The following computer programs have been used:-

1) Daily layover time matrix - rows departure and columns arrival .
2) Daily layover time matrix - columns departure and rows arrival
3) Weekly layover time matrix - rows departure and columns arrival
4) Weekly time matrix - columnsdeparture and rowsarrival
5) Transportation program
6) Integer linear programming program
7) Assignment program
8) Sum of squares and product program
9) Inverse matrix program
10) Mean program

1.7) Advantages of Computer Programs.

The advantages of using a computer in order to frame the schedule, are given below:-

1) To avoid the loss of time devoted to the manual elaboration of the Summer and Winter scheduling of aircrews.

2) To print a good reply to every modification of flight, that is, any addition or cancellation of flight and at the same time optimal solution can be obtained.

3) To modify the schedule in the light of any addition or cancellation of flights, the new schedule can be framed in a few minutes only.

4) Using the same principle, the scheduling of air hostesses and stewards can be prepared.

CHAPTER II

GENERAL THEORY OF TIME MATRIX

2.1) Calculation of Time Matrix.

A time matrix consists of rectangular array of time; unlike the determinant it does not have quantitative value. If the number of rows and columns are equal then it is a square matrix. The time matrix is obtained by taking all possible combinations of arrivals and departures of all flights at changeover City keeping in view the legal limitations which are laid down by the Crew Association and Airline Authority.

There is only one variable - time on the ground away from the location at which the crew is based. Minimizing of this time has the dual advantage of making more crews available at the home base and reducing the expenditure of the crews on boarding and lodging at changeover places.

2.2) General Theory.

A simple illustration will serve to demonstrate the fundamental nature of the problem. Consider a flight system that involves only three locations on a line and two flights in each direction as shown in the diagram.

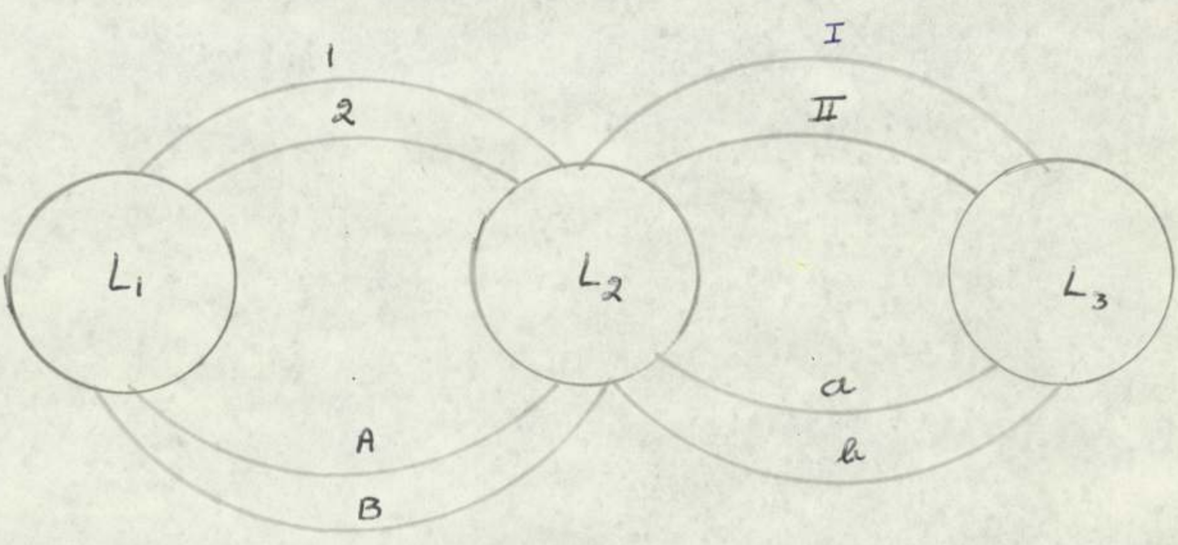


Diagram - 1

2.2) contd.

If the turn around in either direction at L_2 is excluded, the possible combinations of eight flights is represented in the following table.

Flight Segment		I		II	
		1	2	1	2
A	a		x		
	b				
B	a				
	b				

Table 1 - Matrix of possible flight combinations.

Let in this layover time matrix, the cell marked "x" represent the combination of four flights (I,2,A,a) and this is the time away from the home base on the ground. Similarly all cells are filled by taking different possible combinations.

Consequently, in order to obtain a mathematical model in the light of legal limitations, the problem of flight pairings and allocations is reduced to a series of "Two City Problem" and "One City Problem", which are as follows.

2.3) Mathematical Models.

2.3.1) Two City Problem.

If the flight time between the two cities A and B is less than or equal to eight hours for

2.3) contd.

2.3.1) contd.

both directions i.e.

$$\vec{AB} + \vec{BA} \leqslant 8 \text{ hours.}$$

Then the above problem is reduced to two city problem and layover time for each city is calculated. Suppose there are n flights from A to B and m flights from B to A (n odd flights and m even flights).

Diagramatically it can be represented as follows:-

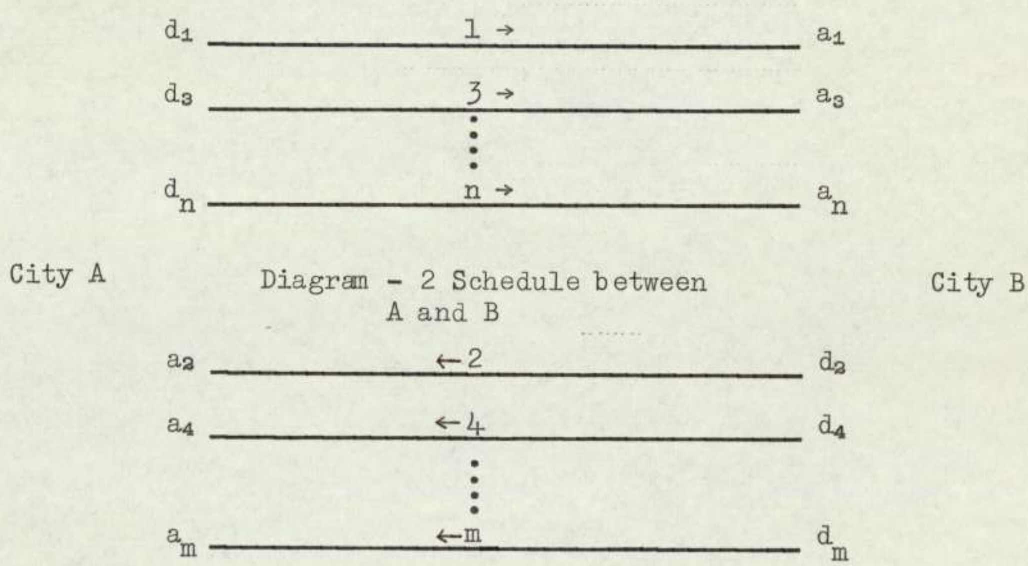


Diagram - 2 Schedule between A and B

where all a's and d's are the arrival and departure times of flights. First the layover time matrix for City A is calculated by taking the City B as home base and similarly time matrix for the City B.

2.3) contd.

2.3.1) contd.

Arrive On

Leave on	a_2	a_4		a_m
d_1	ℓ_{21}	ℓ_{41}	. . . . .	ℓ_{m1}
d_3	ℓ_{23}	ℓ_{43}	. . . . .	ℓ_{m2}
⋮				
d_n	ℓ_{2n}	ℓ_{4n}	. . . . .	ℓ_{mn}

Table 2-A { Layover time at A crew based at B

Leave On

Arrive on	d_2	d_4		d_m
a_1	ℓ_{12}	ℓ_{14}	. . . . .	ℓ_{1m}
a_3	ℓ_{32}	ℓ_{34}	. . . . .	ℓ_{3m}
⋮				
a_n	ℓ_{n2}	ℓ_{n4}	. . . . .	a_{nm}

Table 2-B { Layover time at B and crew based at A.

where ℓ_{21} is the layover time at City A $(a_2 - d_1)$ and ℓ_{12} is the layover time at B $(a_1 - d_2)$ and similarly for all cells.

The above two matrices are combined into one composite matrix by comparing step by step analysis of the respective corresponding elements of the two matrices in such a manner that the minimum value for both set is selected for the third matrix. Each cell of the third matrix is marked with A or B subject to the selected element, where the crew is based. The third matrix is as follows:-

2.3) contd.

2.3.1) contd.

Flight Nos.

	2	4	. . . . .	m
1	L_{12}	L_{14}		L_{1m}
3	L_{32}	L_{34}		L_{2m}
Flight Nos.				
n	L_{n1}	L_{n2}		L_{nm}

Table - 3 Composite matrix

where L_{12} is the minimum value of ℓ_{12} and ℓ_{21} similarly for all other cells.

It is necessary that the third matrix must be square. It is only when n = m, if this is not the case, then other legal flights can be considered at that city having less flights.

2.3.2) Example.

Suppose there are two cities A and B, joined by direct route i.e. A to B and B to A and four planes leave A for B and they are denoted by odd numbers 1,3,5,7 and the flight from B to A be denoted by even numbers 2,4,6,8. The arrival and departure time for each flight is known. The flight time between cities (A and B) and (B and A) is uniform and constant. The flying time of the round trip is less than eight hours.

2.3) contd.

2.3.2) contd.

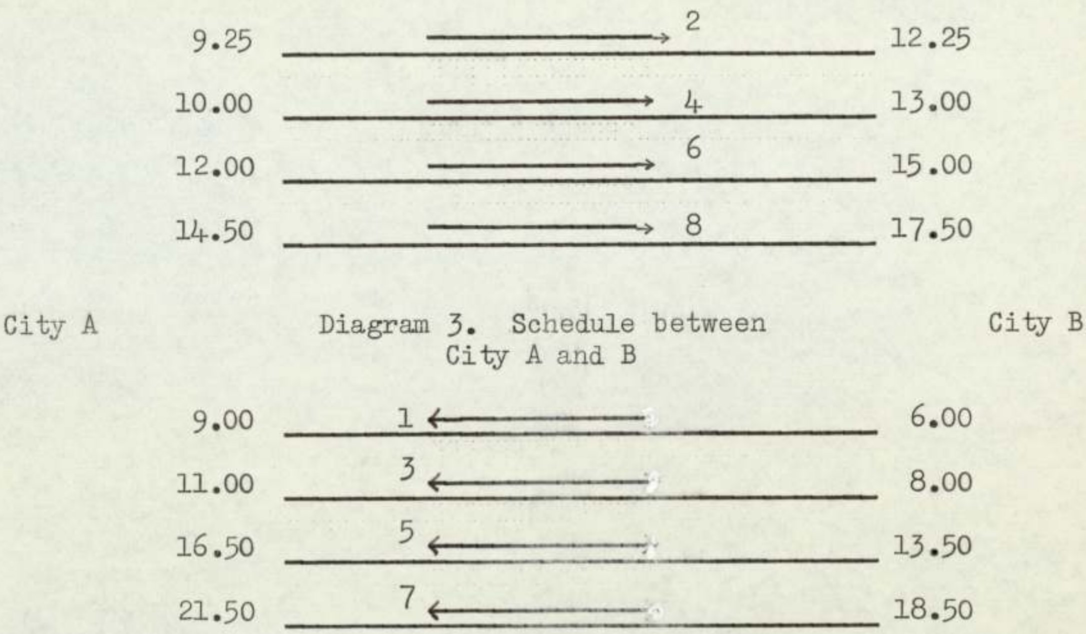


Diagram 3. Schedule between City A and B

There are certain limitations beyond which the crew cannot fly

i) A crew cannot fly more than 8 consecutive hours in 24 hours, and 10 hoursduty time in 24 hours.

ii) Every crew must have rest periods after each flight. Such rest periods should be equal to double the duty time.

iii) When a flight is delayed en route due to unforeseen circumstances and the delay is more than 3 hours, the Captain of the crew can declare a layover to ensure a minimum rest of eight hours. If any element of the matrix is more than 3 hours and less than 9 hours for domestic flights, then 24 hours are added to that element. In the case of international flights the upper limit is 9-30 hours.

To calculate the layover time matrix for each City, assuming first that all the crews are based at City A

2.3) contd.

2.3.2) contd.

and second assuming that all crews are based at B, the number in each cell represents the time away from their home base. The two layover time matrices are calculated by applying the 'two city method' as described above.

Layover time matrix at City A.

	a_1	a_3	a_5	a_7
d_2	24.25	22.25	16.75	11.75
d_4	21.00	23.00	17.00	12.50
d_6	3.00	1.00	19.50	14.50
d_8	29.50	27.50	22.00	17.00

Table 4-A Crew based at City B.

Layover time matrix at City B.

	d_1	d_3	d_5	d_7
a_2	17.75	17.00	15.00	12.50
a_4	19.75	19.00	17.00	14.50
a_6	1.75	2.50	22.50	20.00
a_8	30.75	31.50	20.00	1.00

Table 4-B Crew based at City A.

Another matrix is derived from these two matrices by choosing the smallest element from the respective corresponding cells of the matrics. Then the matrix so obtained is known as composite matrix and presented in Table No.4-C. Each selected element is marked with A or B

2.3) contd.

2.3.2) contd.

showing where the crew is based.

The composite matrix is as follows:-

Flight Nos.

Flight Nos.	1	3	5	7
2	A 17.75	A 17.00	A 15.00	B 11.75
4	B 1.00	A 19.00	A or B 17.00	B 12.50
6	A 1.75	B 1.00	B 19.50	B 14.50
8	B 29.50	B 27.50	A 20.00	B 1.00

Table 4-C Composite layover time matrix.

2.4) <u>One City Problem.</u>

If the flight time between the two cities is greater than eight hours in either direction i.e.

$$\vec{AB} \quad \text{OR} \quad \vec{BA} \quad > \quad 8 \text{ hours}$$

then according to the legal limitations, the crew is not allowed to fly more than eight hours. This means the Airline Authority have got to replace the crew somewhere between the two terminal points. Therefore, the two city problem is reduced to "one city problem."

In order to find a mathematical model, let us consider at a city 'x' arriving n flights from the terminal A bound for city B and m flights from the terminal B bound for city A. Altogether (n+m) flights can be represented diagramatically as follows:-

2.4) contd.

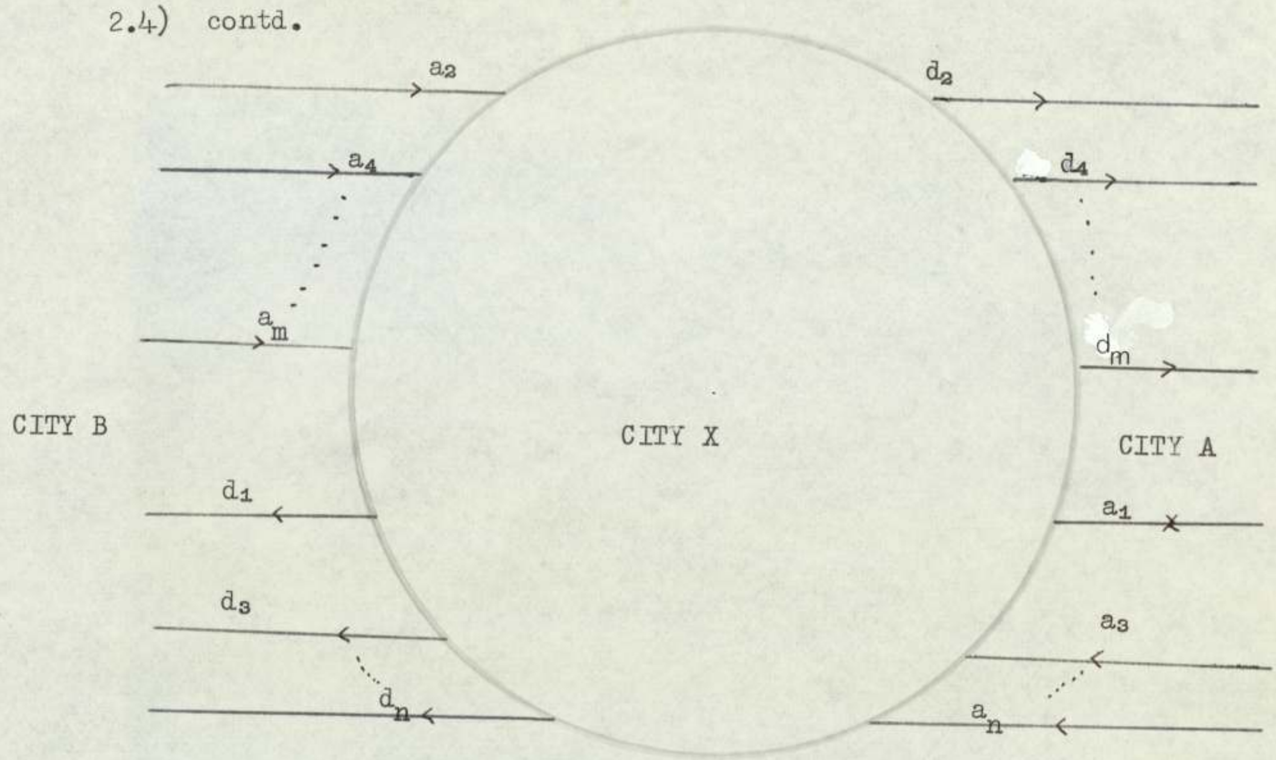


Diagram - 4 - "One City Problem".

where all a's and d's stand for arrivals and departures respectively.

The crew arriving at a_1 can be assigned for next duty to anyone of departures after taking the due rest. Such rest period will be equal twice the duty time and not less than eight hours in any case. The time matrix will take the form as given below:-

2.4) contd.

	a_1	a_3	.....	a_n	a_2	a_4	.....	a_m
d_1	L_{11}	L_{31}	.....	L_{n1}	L_{21}	L_{41}	.....	L_{m1}
d_3	L_{13}	L_{33}	.....	L_{n3}	L_{23}	L_{43}	.....	L_{m3}
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$
d_n	L_{1n}	L_{3n}	.....	L_{nn}	L_{2n}	L_{4n}	.....	L_{mn}
d_2	L_{12}	L_{32}	.....	L_{n2}	L_{22}		.....	L_{m2}
d_4	L_{14}	L_{34}	.....	L_{n4}	L_{24}		.....	L_{m4}
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$			$\vdots$
d_m	L_{1m}	L_{3m}	.....	L_{nm}	L_{2m}		.....	L_{mm}

Table 5. "Time matrix for One City Problem"
where L_{mn} is the layover time of the crew arriving on flight 'm' and departing on flight 'n'.

2.4.1) Example.

Suppose there are two cities A and B, and the flight time between two cities is greater than eight hours. According to the flying rules a crew cannot fly more than eight hours during any 24 consecutive hours. Such rest period will be equal to twice the number of hours on duty.

This means the airline authority have got to replace the crew at a city somewhere between two cities. Let this city be 'x'. Suppose there are six weekly flights at City X , three flights arriving from City A bound for City B and denoted by 2,4,6 and three flights from City B bound for City A and are denoted by 1,3,5.

The time table for these flights is given below:-

2.4) contd.

2.4.1) contd.

READ DOWNWARD						READ UPWARD		
WEDNESDAY	TUESDAY	MONDAY	CITY			TUESDAY	THURSDAY	SATURDAY
Flight No. 6	Flight No.4	Flight No. 2				Flight No.1	Flight No.3	Flight No. 5
7-30	7-30	7-30	DEP	A	ARR	4-45 WED	4-45 FRI	23-55
9-00	10-40	9-55	ARR	X	DEP	20-20	20-30	16-45
9-45	11-55	10-40	DEP		ARR	19-30	19-40	15-55
15.00	17-05	15-50	ARR	B	DEP	12-45	12-45	9-00

Table 6. Time table for City A and B.

2.4) contd.

2.4.1) contd.

Here one point to be noted is that the flying time for all flights is not equal because these flights come through different cities i.e. the route of each flight is not the same.

The flight duty time will commence from a time a crew reports at the airport, from which he will be operating a flight or a series of flights. The duty time starts 45 minutes before scheduled departure for domestic flights, and one hour before scheduled departure for international flights and ends at the destination airport allowing 15 minutes for domestic flights, and 30 minutes for international flights for completing the formalities.

This example relates to international flight. The duty time for each flight is shown in the following diagram:-

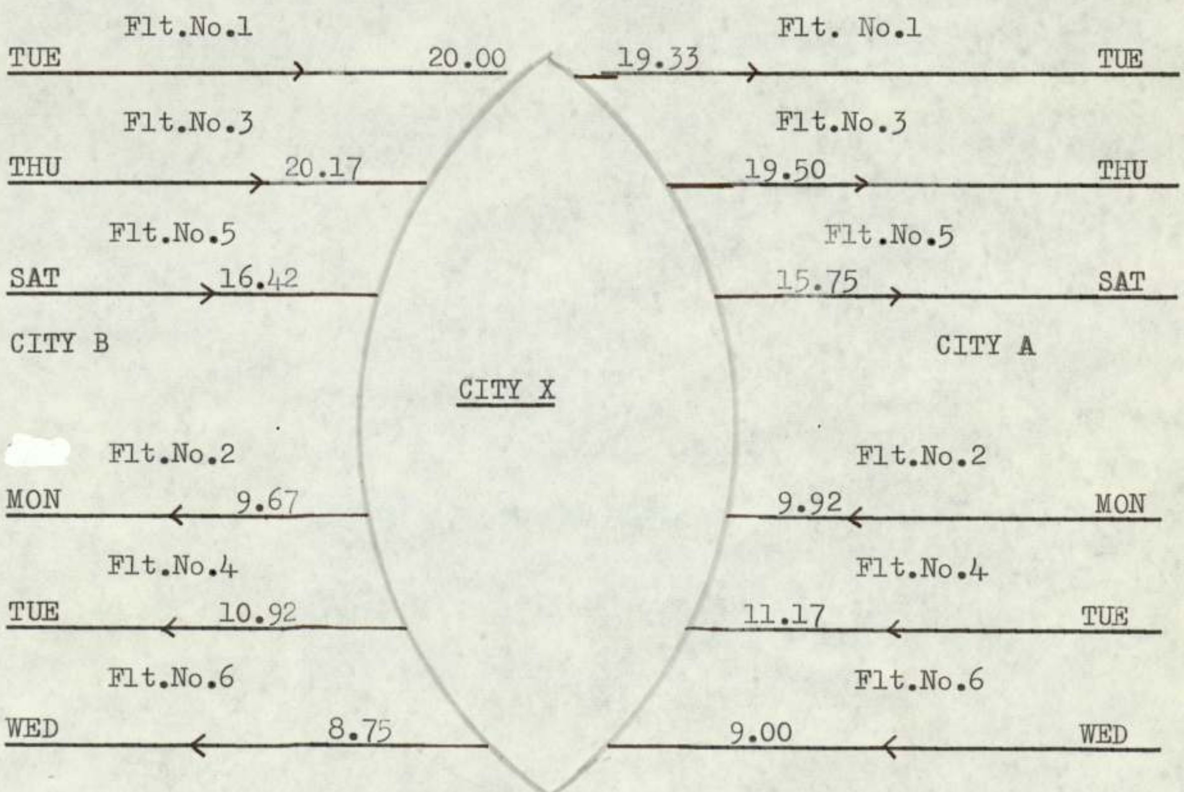


Diagram - 5 - Schedule at City X

2.4) contd.

2.4.1) contd.

The layover time matrix for the City 'X' is calculated by applying 'one city method'. The crew arriving on flight No.1 can go on any one of departures 1,3,5,2,4 and 6. Similarly taking all possible combinations of arrivals and departures, the layover time matrix for City X is calculated and given below:-

	a_2	a_4	a_6	a_1	a_3	a_5
d_2	167.25	142.50	120.17	133.67	85.50	41.25
d_4	24.50	167.75	145.42	158.92	110.75	66.50
d_6	46.33	21.58	167.25	180.75	132.58	88.33
d_1	32.91	176.16	153.83	167.33	119.16	74.91
d_3	81.08	56.33	34.00	47.50	167.33	123.08
d_5	125.33	100.58	78.25	91.75	43.58	167.33

Table-7 - Time Matrix for City X.

2.5) Computer Programs.

The application of computer techniques for the calculation of large layover time matrices is advisable to facilitate and speed up the calculations of the results. The following computer programs are developed to calculate the time matrics and attached in the Appendices (Nos.1,2,3,4).

1) Daily time matrix - row departures and column arrivals
2) Daily time matrix - row arrivals and column departures
3) Weekly time matrix- row departures and column arrivals
4) Weekly time matrix- row arrivals and column departures

There are three possible cases for calculating the layover time matrices:-

2.5) contd.

1) If the time table for all flights is the same throughout the week, and return flight time is less than or equal to 8 hours. then the 'two city method' is applied to calculate the layover time matrices for each city. The computer programmes Nos. 1 and 2 will be applied to calculate the daily layover time matrices. Then from these two matrices, a composite matrix will be obtained as described in the 'two city method'.

2) In the second case if the time table for all flights is not the same throughout the week, then the last two programmes can be applied to calculate the weekly layover time matrices. After that a composite matrix can be calculated as above.

3) Now in the third case if the flight time between two cities is greater than 8 hours, then according to the flying rules, the crew will have to be replaced somewhere along the route, then the 'one city method' is applicable to calculate the layover time matrix, and computer programmes Nos. 3 or 4 can be applied to calculate the weekly layover time matrix.

All these programmes run successfully on ICL 1900 and data cards are prepared for the examples which are described above after two city and one city method. The ICL 1900 took only 13 seconds for the calculation of the 20 × 20 matrix.

2.6.1) <u>Data cards for the two city method</u>.

Daily layover time matrix for City A //

2.6.1) contd.

24.50		0.50	
1		24	
4			
1	9.00	1	7.00
1	11.00	1	9.50
1	16.50	1	15.50
1	21.50	1	21.00

The program No.I is applied to get the layover time matrix for City A - departure row wise and arrival column wise.

2.6.2) Daily layover time matrix for City B//

24.50		0.50	
1		24	
4			
1	6.00	1	10.00
1	8.00	1	12.50
1	13.50	1	18.50
1	18.50	1	24.00

The program No.II is applied to get the layover time matrix for City B - arrival row wise and departure column wise.

In both data cards, the first and third column represent the day of the week such as Sunday = 1, Monday = 2 .... Saturday = 7. Second and fourth represent the arrival and departure time respectively. 4 shows the number of the flights in a day at City A and B. 1 and 24 show that the calculations are made at day 1 and 24.00 hours. As the return flying time between the cities

2.6.2) contd.

is less than 8 hours, the 0.50 is taken as lower limit of layover time at City A and B; and, at the most, the crew can go next day on the same flight, therefore 24.50 is the upper limit of layover time at City A and B.

The same results are obtained as in the case of examples described above.

2.6.3) "Data Cards for one city problem"

Weekly layover time matrix for City X//

180.00		12.00	
1		24	
6			
2	10.42	2	9.67
3	11.17	3	10.92
4	9.50	4	8.75
3	20.00	3	19.33
5	20.17	5	19.50
7	16.42	7	15.75

The computer program No.IV is applied to calculate the layover time matrix for City X. In this case 12.00 is the minimum rest period for the crews which is based on the duty time and 180.00 is the maximum rest period. The duty time for each flight is calculated and if any element of the matrix is less than twice the duty time, then 168 hours are added to that element, because the flying time is not the same for all flights due to the different routes. The same result is obtained as in the example of one city method (Table 7).

CHAPTER III

SELECTION OF SLIP STATIONS AND THEIR TIME MATRICES

3.1) <u>Size of the Problem.</u>

From looking to the time table of P.I.A. attached in the Appendix, all flights can be divided into the following sectors, showing the number of weekly flights in each sector:-

Name of Sector	No.of return flights
1) Western	11
2) Karachi-Dacca	20
3) Lahore-Dacca	5
4) China	2
5) Persian Gulf	4
6) Bangkok	2
Total flights	44

3.2) <u>Selection of Slip Stations.</u>

If the flying time between two cities is more than eight hours, then the flight crews have to be changed at the intermediate stops for safety requirements in accordance with the working and rest time regulations. The choice of slip station is made on arrivals and departures in a week. The off going crew is taking a rest at such a stop while waiting for the next flight. The rest time is usually based on the duty time and it should not be less than eight hours in any case. A route can be divided into parts between the terminal points.

Some places along the route cannot be used as slip station due to political reasons or lack of facilities. A few stations with poor hotels or without any facilities could be used as slip stations after large investments. Some stations are now regular slip stations with very good and cheap accommodation because they are airline company-owned hotels. With the introduction of faster aircrafts such as the Concorde or the Jumbo jets, these

3.2) contd.

stations might be overflown in the near future. Assuming the possible slip stations are known and their number is limited, the choice of these stations as slip stations has to be based on costs.

If station 'A' is considered as slip station the rest time needed by the crews at the station 'A' depends on the flight duty time before reaching this point. This means that the number of crews in a hotel at station 'A' depends on

i) flight duty time from previous slip station or base station to station 'A';

ii) number of flight calls at station (frequency)

With a given time table, the arrival and departure times of aircraft at station 'A' are known. Crews coming and going from station 'A' can be given the required rest time according to the regulations.

For the P.I.A. the flight time in the Western Sector is more than eight hours in either direction. P.I.A. Authority have got to replace their crews, somewhere between the two terminal points - Karachi and London. Keeping in view the legal limitations, frequency of flights and other aspects mentioned above, the following cities are being used by P.I.A. as slip stations

1. Beirut
2. Istanbul
3. Cairo
4. London
5. with Karachi as the home base.

The flight time between Karachi and London and

3.2) contd.

vice-versa in the Western Sector and Karachi - Shanghai - Karachi in the China Sector, is greater than eight hours. In all other remaining sectors the return flying time is less than eight hours, so the layover time matrices for these sectors will be calculated by the"two city method".

In the case of Western flights, the airline authority has got to replace the crew somewhere between London and Karachi according to the flying rules. The case of the China Sector is explained below:-

3.3) China and Moscow Flights.

There are two return weekly flights to China and one Moscow flight bound for London in the Western Sector. The flying time in both cases is more than eight hours. The crew arriving at Moscow on flight No. PK.720 will have to wait for the next assignment - about 6 days. In case of the China Sector the crew arriving on PK.752 will have to wait for the next assignment - about 5 days. In order to avoid this idle period, the airline authority and crews' association are agreed that a double crew should be used on these two sectors with the conditions that the crews of flight No. PK.717 and PK.720 will get at least three days off from any duty of the Company at London and Karachi respectively. One crew will operate the plane up to Moscow and the second will take the control of the plane from Moscow. This double crew will be used for China flights and will go to Dacca one day before the flight as dead head crew. This means that 12 crews are used for 11 Western flights and 2 crews are used for 2 return China flights.

3.4) Flight combinations at slip stations.

From a look at the time table attached in the Appendix the crews of West bound flights Nos. PK.713, 721 and East bound flights Nos. 706, 718 can be changed either at Beirut or Istanbul. The possible combinations are as follows:-

		Istanbul PK	Beirut PK
1.	W.B.F.	705,709,715,713	701,703,707,721
	E.B.F.	722,712,716,718	702,706,708,714
2.	W.B.F.	705,709,715,721	701,703,707,713
	E.B.F.	722,712,716,718	702,706,708,714
3.	W.B.F.	705,709,715,721	701,703,707,713
	E.B.F.	722,712,716,706	702,708,714,718
4.	W.B.F.	705,709,715,713	701,703,707,721
	E.B.F.	722,712,716,706	702,708,714,718
5.	W.B.F.	705,709,715	701,703,707,713,721
	E.B.F.	722,712,716	702,706,708,714,718
6.	W.B.F.	705,709,715,713,721	701,703,707
	E.B.F.	722,712,716,706,718	702,708,714

TABLE - 8:- Possible Combinations for Istanbul and Beirut

where W.B.F. West bound flights Karachi - London

E.B.F. East bound flights London - Karachi

3.5) "Minimum Rest Period of All Flights".

The duty time from the terminals to changeover places are calculated and given below. The duty time for international flights commence one hour before scheduled departure and end half an hour after scheduled arrival.

3.5) contd.

Route and Flight No.		Duty Time in Hours.		Minimum Rest Period (double the duty time)	
		hrs.	min.	hrs.	min.
KARACHI-BEIRUT					
	PK 701	6 -	55	13 -	50
	PK 703	7 -	40	15 -	20
	PK 707	6 -	00	12 -	00
	PK 713	7 -	15	14 -	30
	PK 721	7 -	15	14 -	30
LONDON-BEIRUT					
	PK 702	7 -	15	14 -	30
	PK 706	8 -	35	17 -	10
	PK 708	7 -	25	14 -	50
	PK 714	7 -	25	14 -	50
	PK 718	8 -	35	17 -	10
KARACHI-ISTANBUL					
	PK 705	8 -	10	16 -	20
	PK 709	9 -	25	18 -	50
	PK 715	8 -	10	16 -	20
	PK 721	9 -	40	19 -	20
	PK 713	9 -	40	19 -	20
LONDON-ISTANBUL					
	PK 722	6 -	25	12 -	50
	PK 712	6 -	25	12 -	50
	PK 716	6 -	25	12 -	50
	PK 718	6 -	20	12 -	40
	PK 706	6 -	15	12 -	30

3.5) contd.

Route and Flight No.		Duty Time in Hours.	Minimum Rest Period (double the duty time).
		hrs. min.	hrs. min.
KARACHI-CAIRO			
	PK 711	6 - 40	13 - 20
	PK 719	6 - 40	13 - 20
LONDON-CAIRO			
	PK 704	8 - 25	16 - 50
	PK 710	8 - 25	16 - 50
KARACHI-MOSCOW			
	PK 717	7 - 35	15 - 10
LONDON-MOSCOW			
	PK 710	6 - 45	13 - 30
BEIRUT-LONDON			
	PK 701	7 - 40	15 - 20
	PK 703	7 - 40	15 - 20
	PK 707	7 - 45	15 - 30
	PK 713	9 - 00	18 - 00
	PK 721	9 - 00	18 - 00
ISTANBUL-LONDON			
	PK 705	6 - 45	13 - 30
	PK 709	6 - 35	13 - 20
	PK 715	6 - 45	13 - 30
	PK 721	6 - 30	13 - 00
	PK 713	6 - 30	13 - 00
CAIRO-LONDON			
	PK 711	8 - 50	17 - 40
	PK 719	8 - 50	17 - 40
MOSCOW-LONDON			
	PK 717	6 - 50	13 - 40

For all above mentioned slip stations, "one city method" is used. The computer program IV is applied for the calculation of layover time matrices at Istanbul, Beirut, Cairo and London. All the time matrices so obtained are modified in the light of Section (3.5). If any element is less than the

3.5) contd.

minimum rest period, 168 hours are added to that element, because at the latest the crew can go next week on the same flight. All modified time matrices are as follows:-

3.5) contd.

BEIRUT PK 702, 706, 708, 714, 718, 713, 701, 703, 707, 721.

	a_{701}	a_{703}	a_{708}	a_{713}	a_{721}	a_{702}	a_{706}	a_{708}	a_{714}	a_{718}
d_{701}	167.25	142.50	120.17	70.92	22.92	133.67	108.34	85.50	41.25	184.09
d_{703}	24.50	167.75	145.42	96.17	48.17	158.92	133.59	110.75	66.50	41.34
d_{707}	46.33	21.58	167.25	118.00	70.00	180.75	155.42	132.58	88.33	63.17
d_{713}	95.66	70.91	48.50	167.33	119.33	62.08	36.75	181.91	137.66	112.50
d_{721}	143.66	118.91	96.50	47.33	167.33	110.08	84.75	61.91	17.66	160.50
d_{712}	32.91	176.16	153.83	104.58	56.58	167.33	142.00	119.16	74.91	49.75
d_{706}	58.25	33.50	179.17	129.92	81.92	24.67	167.34	144.50	100.25	75.09
d_{708}	81.08	56.33	34.00	152.75	104.75	47.50	22.17	167.33	123.08	97.92
d_{714}	125.33	100.58	78.25	29.00	149.00	91.75	66.42	43.58	167.33	142.17
d_{718}	150.50	125.75	103.42	54.17	174.17	116.92	91.59	68.75	24.50	167.34

Table 9 - Layover Time matrix.

3.5) contd.

(BEIRUT PK 701, 703, 707, 702, 714, 718)

	a_{701}	a_{703}	a_{707}	a_{702}	a_{708}	a_{714}
d_{701}	167.25	142.50	120.17	133.67	85.50	41.25
d_{703}	24.50	167.75	145.42	158.92	110.75	66.50
d_{707}	46.33	21.58	167.25	180.75	132.58	88.33
d_{702}	32.91	176.16	153.83	167.33	119.16	74.91
d_{708}	81.08	56.33	34.00	47.50	167.33	123.08
d_{714}	125.33	100.58	78.25	91.75	43.58	167.33

Table 10. Layover time matrix

(BEIRUT PK 701, 703, 707, 721, 702, 706, 708, 714)

	a_{701}	a_{703}	a_{707}	a_{721}	a_{702}	a_{706}	a_{708}	a_{714}
d_{701}	167.25	142.50	120.17	22.92	133.67	108.34	85.50	41.25
d_{703}	24.50	167.75	145.42	48.17	158.92	133.59	110.75	66.50
d_{707}	46.33	21.58	167.25	70.00	180.75	155.42	132.58	88.33
d_{721}	143.66	118.91	96.50	167.33	110.08	84.75	61.91	17.66
d_{702}	32.91	176.16	153.83	56.58	167.33	142.00	119.16	74.91
d_{706}	58.25	33.50	179.17	81.92	24.67	167.34	144.50	100.25
d_{708}	81.08	56.33	34.00	104.75	47.50	22.17	167.33	123.08
d_{714}	125.33	100.58	78.25	149.00	91.75	66.42	43.58	167.33

Table 11. Layover time matrix.

3.5) contd.

BEIRUT PK 701, 703, 707, 713, 702, 706, 708, 714

	a_{701}	a_{703}	a_{707}	a_{713}	a_{702}	a_{706}	a_{708}	a_{714}
d_{701}	167.25	142.50	120.17	70.92	133.67	108.34	85.50	41.25
d_{703}	24.50	167.75	145.42	96.17	158.92	133.59	110.75	66.50
d_{707}	46.33	21.58	167.25	118.00	180.75	155.42	132.58	88.33
d_{713}	95.66	70.91	48.50	167.33	62.08	36.75	181.91	137.66
d_{702}	32.91	176.16	153.83	104.58	167.33	142.00	119.16	74.91
d_{706}	58.25	33.50	179.17	129.92	24.67	167.34	144.50	100.25
d_{708}	81.08	56.33	34.00	152.75	47.50	22.17	167.33	123.08
d_{714}	125.33	100.58	78.25	29.00	91.75	66.42	43.58	167.33

Table 12. Layover time matrix.

BEIRUT PK 701, 703, 707, 713, 702, 708, 714, 718.

	a_{701}	a_{703}	a_{707}	a_{713}	a_{702}	a_{708}	a_{714}	a_{718}
d_{701}	167.25	142.50	120.17	70.92	133.67	85.50	41.25	184.09
d_{703}	24.50	167.75	145.42	96.17	158.92	110.75	66.50	41.34
d_{707}	46.33	21.58	167.25	118.00	180.75	132.58	88.33	63.17
d_{713}	95.66	70.91	48.50	167.33	62.08	181.91	137.66	112.50
d_{702}	32.91	176.16	153.83	104.58	167.33	119.16	74.91	49.75
d_{708}	81.08	56.33	34.00	152.75	47.50	167.33	123.08	97.92
d_{714}	125.33	100.58	58.25	29.00	91.75	43.58	167.33	142.17
d_{718}	150.50	125.75	103.42	54.17	116.92	68.75	24.50	167.34

Table 13. Layover time matrix.

3.5) contd.

BEIRUT PK 701,703,707,721,702,708,714,718

	a_{701}	a_{703}	a_{707}	a_{721}	a_{702}	a_{708}	a_{714}	a_{718}
d_{701}	167.25	142.50	120.17	22.92	133.67	85.50	41.25	184.09
d_{703}	24.50	167.75	145.42	48.17	158.92	110.75	66.50	41.34
d_{707}	46.33	21.58	167.25	70.00	180.75	132.58	88.33	63.17
d_{721}	143.66	118.91	96.50	167.33	110.08	61.91	17.66	160.50
d_{702}	32.91	176.16	153.83	56.58	167.33	119.16	74.91	49.75
d_{708}	81.08	56.33	34.00	104.75	47.50	167.33	123.08	97.92
d_{714}	125.33	100.58	78.25	149.00	91.75	43.58	167.33	142.17
d_{718}	150.50	125.75	103.42	174.17	116.92	68.75	24.50	167.34

Table 14 Layover time matrix.

3.5) contd.

ISTANBUL 705,709,713,721,715,722,706,712,716,718

	a_{705}	a_{709}	a_{713}	a_{715}	a_{721}	a_{722}	a_{706}	a_{712}	a_{716}	a_{718}
d_{705}	167.50	135.50	111.25	95.50	63.25	33.25	153.25	105.25	81.25	61.09
d_{709}	31.41	167.41	143.10	127.41	95.16	65.16	17.33	137.16	113.16	93.00
d_{713}	55.66	23.66	167.41	151.66	119.41	89.41	41.58	161.41	137.41	117.25
d_{715}	71.50	39.50	183.25	167.50	135.25	105.25	57.42	177.25	153.25	133.09
d_{721}	103.66	71.66	47.41	31.66	167.41	137.41	89.58	41.41	17.41	156.25
d_{722}	133.66	101.66	77.41	61.66	29.41	167.41	119.58	71.41	47.41	27.25
d_{706}	181.41	149.41	125.16	109.41	77.16	47.16	167.33	119.16	95.16	75.00
d_{712}	61.66	29.66	173.41	157.66	125.41	95.41	47.58	167.41	143.41	123.41
d_{716}	85.66	53.66	29.41	181.66	149.41	119.41	71.58	23.41	167.41	147.25
d_{718}	105.66	73.66	49.41	33.66	169.41	139.41	91.58	43.41	19.41	167.25

Table 15 Layover time matrix

3.5) contd.

ISTANBUL PK 705,709,715,713,722,706,712,716

	a_{705}	a_{709}	a_{713}	a_{715}	a_{722}	a_{706}	a_{712}	a_{716}
d_{705}	167.50	135.50	111.25	95.50	33.25	153.42	105.25	81.25
d_{709}	31.41	167.41	143.41	127.41	65.16	17.33	137.16	113.16
d_{713}	55.66	23.66	167.41	151.66	89.41	41.58	161.41	137.41
d_{715}	71.50	39.50	183.25	167.50	105.25	57.42	177.25	153.25
d_{722}	133.66	101.66	77.41	61.66	167.41	119.58	71.41	47.41
d_{706}	181.41	149.41	125.16	109.41	47.16	167.33	119.16	95.16
d_{712}	61.66	29.66	173.41	157.66	95.41	47.58	167.41	143.41
d_{716}	85.66	53.66	29.41	181.66	119.41	71.58	23.41	167.41

Table 16. Layover time matrix.

ISTANBUL PK 705,709,715,721,722,712,716,718

	a_{705}	a_{709}	a_{715}	a_{721}	a_{722}	a_{712}	a_{716}	a_{718}
d_{705}	167.50	135.50	95.50	63.25	33.25	105.25	81.25	61.09
d_{709}	31.41	167.41	127.41	95.16	65.16	137.16	113.16	93.00
d_{715}	71.50	39.50	167.50	135.25	105.25	177.25	153.25	133.09
d_{721}	103.66	71.66	31.66	167.41	137.41	41.41	17.41	165.25
d_{722}	133.66	101.66	61.66	29.41	167.41	71.41	47.41	27.25
d_{712}	61.66	29.66	157.66	125.41	95.41	167.41	143.41	123.25
d_{716}	85.66	53.66	181.66	149.41	119.41	23.41	167.41	147.25
d_{718}	105.66	73.66	33.66	169.41	139.41	43.41	19.41	167.25

Table 17 Layover time matrix.

3.5) contd.

ISTANBUL PK 705,709,715,721,722,706,712,716.

	a_{705}	a_{709}	a_{715}	a_{721}	a_{722}	a_{706}	a_{712}	d_{716}
d_{705}	167.50	135.50	95.50	63.25	33.25	153.42	105.25	81.25
d_{709}	31.41	167.41	127.41	95.16	65.16	17.33	137.16	113.16
d_{715}	71.50	39.50	167.50	135.25	105.25	57.42	177.25	153.25
d_{721}	103.66	71.66	31.66	167.41	137.41	89.58	41.41	17.41
d_{722}	133.66	101.66	61.66	29.41	167.41	119.58	71.41	47.41
d_{706}	181.91	149.91	109.91	77.66	47.66	167.83	119.66	95.66
d_{712}	61.66	29.66	157.66	125.41	95.41	47.58	167.41	143.41
d_{716}	85.66	53.66	181.66	149.41	119.41	71.58	23.41	167.41

Table 18. Layover time matrix.

ISTANBUL PK 705,709,715,713,722,712,716,718.

	a_{705}	a_{709}	a_{713}	a_{715}	a_{722}	d_{712}	d_{716}	d_{718}
d_{705}	167.50	135.50	111.25	95.50	33.25	105.25	81.25	61.09
d_{709}	31.41	167.41	134.10	127.41	65.16	137.16	113.16	93.00
d_{713}	55.66	23.66	167.41	151.66	89.41	161.41	137.41	117.25
d_{715}	71.50	39.50	183.25	167.50	105.25	177.25	153.25	133.25
d_{722}	133.66	101.66	77.41	61.66	167.41	71.41	47.41	27.25
d_{712}	61.66	29.66	173.41	157.66	95.41	167.41	143.41	123.25
d_{716}	85.66	53.66	29.41	181.66	119.41	23.41	167.41	147.25
d_{718}	105.66	73.66	49.41	33.66	139.41	43.41	19.41	167.25

Table 19. Layover time matrix.

3.5) contd.

ISTANBUL PK 705,709,715,722,712,716

	a_{705}	a_{709}	a_{715}	a_{722}	a_{712}	a_{716}
d_{705}	167.50	135.50	95.50	33.25	105.25	81.25
d_{709}	31.41	167.41	127.41	65.16	137.16	113.16
d_{715}	71.50	39.50	167.50	105.50	177.25	153.25
d_{722}	133.66	101.66	61.66	167.41	71.41	47.41
d_{712}	61.66	29.66	157.66	95.41	167.41	143.41
d_{716}	85.66	53.66	181.66	119.41	23.41	167.41

Table 20. Layover time matrix.

CAIRO PK 711,719,704,710.

	a_{711}	a_{719}	a_{704}	d_{710}
d_{711}	167.16	119.16	33.16	153.16
d_{719}	47.16	167.16	81.16	33.17
d_{704}	133.33	85.33	167.33	119.33
d_{710}	13.34	133.33	47.33	167.33

Table 21. Layover time matrix.

3.5) contd.

LONDON ALL FLIGHTS

	a_{701}	a_{703}	a_{705}	a_{707}	a_{709}	a_{711}	a_{713}	a_{715}	a_{717A}	a_{717B}	a_{719}	a_{721}
d_{722}	163.42	138.17	121.58	116.25	89.83	73.33	65.67	49.58	43.33	43.33	25.33	17.70
d_{702}	19.42	162.17	145.58	140.25	113.83	97.33	89.67	73.58	67.33	67.33	49.33	41.70
d_{704}	39.67	182.42	165.83	160.50	134.08	117.58	109.92	93.83	87.58	87.58	69.58	61.95
d_{706}	43.42	18.17	169.58	164.25	137.83	121.33	113.67	97.58	91.33	91.33	73.33	65.70
d_{708}	67.42	42.17	25.58	20.25	161.83	145.33	137.67	121.58	115.33	115.33	97.33	89.70
d_{710}	87.67	62.42	45.83	40.50	14.08	165.58	157.92	141.83	135.58	135.58	117.58	109.95
d_{712}	91.42	66.17	49.58	44.25	17.83	169.33	161.67	145.58	139.33	139.33	121.33	113.70
d_{714}	111.67	86.42	69.83	64.50	38.08	21.58	181.92	165.83	159.58	159.58	141.58	133.95
d_{716}	115.42	90.17	73.58	68.25	41.83	25.33	185.67	169.58	163.33	163.33	145.33	137.70
d_{718}	135.67	110.42	93.83	88.50	62.08	45.58	37.92	21.83	183.58	183.58	165.58	157.95
d_{720} A	139.42	114.17	97.58	92.25	65.83	49.33	41.67	25.58	187.33	187.33	169.33	161.70
d_{720} B	139.42	114.17	97.58	92.25	65.83	49.33	41.67	25.58	187.33	187.33	169.33	161.70

Table 22. Time matrix for London.

3.6) Selection of Bases.

The two city method is applied for the selection of the bases in case the return flight time between two cities is less than or equal to eight hours. The following sectors come in this category

1) Karachi/Dacca Sector
2) Dacca/Lahore "
3) Persian Gulf "
4) Bangkok "

The computer programs III and IV are applied for the calculation of layover time matrices for each sector.

Dacca/Lahore Sector.

Layover Time Matrix at Lahore

	MON a_{725}	TUE a_{725}	WED a_{725}	FRI a_{725}	SUN a_{725}
MON d_{726}	1.08	145.08	121.08	73.08	25.08
TUE d_{726}	25.08	1.08	145.08	97.08	49.08
WED d_{726}	49.08	25.08	1.08	121.08	73.08
FRI d_{726}	97.08	73.08	49.08	1.08	121.08
SUN d_{726}	145.08	121.08	97.08	49.08	1.08

Table 26-A Crew based at Dacca

3.6) contd.

Layover time matrix at Dacca

	MON d_{725}	TUE d_{725}	WED d_{725}	FRI d_{725}	SUN d_{725}
MON a_{726}	2.25	146.25	122.25	74.25	26.25
TUE a_{726}	26.25	2.25	146.25	98.25	50.25
WED a_{726}	50.25	26.25	2.25	122.25	74.25
FRI a_{726}	98.25	74.25	50.25	2.25	122.25
SUN a_{726}	146.25	122.25	98.25	50.25	2.25

Table 26-B Crew based at Lahore.

Composite Layover Time Matrix of Dacca/Lahore Sector

	MON PK 725	TUE PK 725	WED PK 725	FRI PK 725	SUN PK 725
MON PK 726	D 1.08	D 145.08	D 121.08	D 73.08	D 25.08
TUE PK 726	D 25.08	D 1.08	D 145.08	D 97.08	D 49.08
WED PK 726	D 49.08	D 25.08	D 1.08	D 121.08	D 73.08
FRI PK 726	D 97.08	D 73.08	D 49.08	D 1.08	D 121.08
SUN PK 726	D 145.08	D 121.08	D 97.08	D 49.08	D 1.08

Table 26. Composite layover time matrix.

3.6) contd.

Dacca/Bangkok Sector

Layover time Matrix at Bangkok

	THU a_{706-A}	SAT a_{712-A}
THU d_{711A}	1.25	96.25
SAT d_{719A}	49.25	1.25

Table 27-A

Crew Based at Dacca

Layover time Matrix at Dacca

	d_{706A}	d_{712A}
a_{711A}	161.25	42.25
a_{719A}	114.25	161.25

Table 27-B

Crew Basd at Bangkok

Composite Time Matrix

	THU PK 706-A	SAT PK 712-A
THU PK 711-A	D 1.25	B 42.25
SAT PK 719-A	D 49.25	D 1.25

Table 27 - Composite Time Matrix

3.6) contd.

KARACHI/PERSIAN GULF SECTOR

Layover time matrix at Karachi

	WED d_{745}	SUN d_{745}
WED a_{744}	159.50	63.50
SUN a_{744}	87.50	159.50

Table 28-A Crew based at Jaddah

Layover time matrix at Jaddah

	WED a_{745}	SUN a_{745}
WED d_{744}	0.83	72.83
SUN d_{744}	96.83	0.83

Table 28-B Crew based at Karachi

Composite Matrix

	WED - PK 745	SUN - PK 745
WED PK 744	K 0.83	J 63.50
SUN PK 744	J 87.50	K 0.83

Table 28 Compositve layover time matrix.

DACCA/KARACHI SECTOR

3.6.1 Layover time matrix at Dacca

-

		MONDAY			TUESDAY			WEDNESDAY			THURSDAY		FRIDAY			SATURDAY			SUNDAY		
		d_{731}	d723	d_{737}	d731	d735	d_{723}	d_{731}	d723	d737	d_{735-A}	d_{711}	d_{731}	d_{735}	d_{723}	d_{735}	d_{735-A}	d_{719}	d_{731}	d_{735}	d_{723}
MON	a_{724}	171.08	8.58	12.50	27.08	29.00	32.58	51.08	56.50	60.50	81.50	82.58	99.08	101.83	104.58	125.83	129.50	130.58	147.08	149.08	152.75
	a_{730}	1.08	174.58	10.50	25.08	27.83	30.58	49.08	54.58	58.50	79.50	80.58	97.08	99.83	102.58	123.83	127.50	128.58	145.08	147.83	150.75
	a_{738}	159.58	165.08	1.00	15.58	18.33	21.08	39.58	45.08	49.00	70.00	71.08	87.58	90.33	93.08	114.33	118.00	119.08	135.58	138.33	141.25
TUE	a_{724}	147.08	152.58	156.50	171.08	173.83	176.58	27.08	32.58	36.50	57.50	58.58	75.08	77.83	80.58	101.83	105.50	106.58	123.08	125.83	128.75
	a_{730}	145.08	150.581	154.50	1.08	171.83	174.58	25.08	30.58	34.50	55.50	56.58	73.08	75.83	78.58	99.83	103.50	104.58	121.08	123.83	126.75
	a_{736}	143.33	147.83	151.75	166.33	1.08	171.58	22.33	27.83	31.75	52.75	53.83	70.33	73.08	75.83	97.08	100.75	101.83	118.33	121.08	124.00
WED	a_{724}	123.08	128.58	132.50	147.08	149.83	152.58	171.08	176.58	12.50	33.50	34.58	51.08	53.83	56.58	77.83	81.75	82.75	81.75	99.08	101.83
	a_{730}	121.08	126.58	130.58	145.08	147.83	150.58	1.08	174.58	10.50	31.50	32.58	49.08	51.83	54.58	75.83	79.50	80.50	97.08	99.83	102.75
	a_{738}	111.58	117.08	121.00	135.58	138.33	141.08	159.58	165.08	1.00	22.33	23.08	39.58	42.33	45.08	66.33	70.00	71.08	81.58	90.33	93.25
THS	a_{704}	102.08	107.58	111.50	126.08	128.83	131.58	150.08	155.58	159.50	12.50	13.58	30.08	32.83	35.58	56.83	60.50	61.58	78.08	80.33	83.75
	a_{706}	97.08	102.58	106.50	121.08	123.83	126.58	145.08	150.58	154.50	175.50	176.58	25.08	27.83	30.58	51.83	55.50	56.58	73.08	75.83	78.75
FRI	a_{724}	75.58	80.75	84.50	9 9.08	101.83	104.58	123.08	128.58	132.50	153.50	154.52	171.08	173.83	176.58	29.83	35.50	34.58	51.08	53.83	56.75
	a_{730}	73.08	78.58	82.50	97.08	99.83	102.58	121.08	12[illegible].58	130.50	151.50	52.58	1.08	171.83	174.58	27.83	31.50	32.58	49.08	51.83	54.75
	736	70.33	75.83	79.75	94.33	97.08	99.83	118.33	123.83	127.75	148.75	149.83	166.33	1.08	171.83	25.08	28.75	29.83	46.33	49.08	52.00
SAT	a_{710}	54.08	59.58	63.50	78.08	80.83	83.58	102.08	107.58	111.50	132.50	133.58	150.08	152.83	155.58	176.83	12.50	13.58	30.08	32.83	35.75
	a_{712}	49.08	54.58	58.50	73.08	75.83	78.50	97.08	102.58	106.50	127.50	128.58	145.08	147.83	150.58	171.83	175.50	176.58	25.08	27.83	30.75
	a_{736}	46.33	51.83	75.75	70.33	73.08	75.83	94.33	99.83	103.75	124.50	125.83	142.33	145.08	147.83	1.08	172.75	173.83	22.33	25.08	28.00
SUN	a726	27.08	32.58	36.50	51.08	53.83	56.58	75.08	80.58	84.50	105.50	106.58	123.08	125.83	128.58	149.83	153.50	154.58	171.08	173.83	176.75
	a_{730}	25.08	32.58	34.50	49.08	51.83	54.58	73.08	78.58	82.50	103.50	104.58	121.08	123.83	126.58	147.83	151.50	152.58	1.08	171.83	176.75
	a_{736}	22.33	27.83	31.75	46.33	49.08	51.83	70.33	75.83	79.75	100.75	101.83	118.33	121.08	123.83	145.08	148.75	149.83	166.33	1.08	172.00

Table 23:- crew based at Karachi

3.6.2 LAYOVER TIME MATRIX AT KARACHI

		MONDAY			TUESDAY			WEDNESDAY			THURSDAY		FRIDAY			SATURDAY			SUNDAY		
		a_{731}	a_{723}	a_{737}	a_{731}	a_{735}	a_{723}	a_{731}	a_{723}	a_{737}	a_{735A}	a_{711}	a_{731}	a_{735}	a_{723}	a_{735}	a_{735A}	a_{719}	a_{731}	a_{735}	a_{723}
M	d_{724}	158.50	153.00	149.08	134.50	131.75	129.00	110.50	105.00	101.08	80.08	79.00	62.50	59.75	57.00	35.75	32.08	31.00	14.50	11.75	9.00
O	d_{730}	160.50	155.00	151.08	136.50	133.75	131.00	112.50	107.00	103.09	82.08	81.00	64.50	61.75	59.00	37.75	34.08	33.00	16.50	13.75	11.00
N	d_{738}	2.00	164.50	160.58	146.00	143.25	140.50	122.00	116.50	112.50	91.58	90.50	74.00	71.25	68.25	47.25	43.58	42.50	26.00	23.25	20.50
T	d_{724}	14.50	9.00	173.08	158.50	155.75	153.00	134.50	129.00	125.08	104.08	103.00	86.50	83.75	81.00	59.75	56.08	55.00	38.50	35.75	33.00
U	d_{730}	16.50	11.00	175.08	160.50	157.75	155.00	136.50	131.00	127.08	106.08	105.00	86.50	85.75	83.00	61.75	58.08	57.00	40.50	37.75	35.00
E	d_{736}	19.25	13.25	9.83	163.25	160.50	157.75	139.25	133.75	129.83	108.83	107.75	91.25	88.50	85.75	64.50	60.83	59.75	43.25	40.50	37.75
W	d_{724}	38.50	33.00	29.08	14.50	11.75	9.00	158.50	153.00	149.08	128.08	127.00	110.50	107.75	105.00	83.75	80.08	79.00	62.50	59.75	57.00
E	d_{730}	40.50	35.00	31.08	16.50	13.75	11.00	160.50	155.00	151.08	130.08	129.00	112.50	109.75	107.00	85.75	82.08	81.00	64.50	61.75	59.00
D	d_{738}	50.00	44.50	40.58	26.00	23.25	20.25	2.00	164.50	160.58	139.58	138.50	122.00	119.25	116.50	95.25	91.58	90.50	74.00	71.25	68.50
THU	d_{704}	59.50	54.00	50.08	35.50	32.75	30.00	11.50	174.00	2.08	149.08	148.00	131.50	128.75	126.00	104.75	101.08	100.00	83.50	80.75	78.00
	d_{706}	64.50	59.00	55.08	40.08	37.75	35.00	16.50	11.00	175.08	154.08	153.00	136.50	133.75	131.00	109.75	106.08	105.00	88.50	85.75	83.00
F	d_{724}	86.50	81.00	77.08	62.50	59.75	57.00	38.50	33.00	29.08	176.08	175.00	158.50	155.75	153.00	131.75	128.08	127.00	110.50	107.75	105.50
R	d_{730}	88.50	83.00	79.08	64.50	61.75	59.00	40.50	35.00	31.08	10.08	9.00	160.50	157.75	155.00	133.75	130.08	129.00	112.50	109.75	107.00
I	d_{736}	91.25	85.75	81.83	67.25	64.50	61.75	43.25	37.75	33.83	12.83	11.75	163.25	160.50	157.75	136.50	132.83	131.75	115.25	112.50	109.75
S	d_{710}	107.50	102.00	98.08	83.50	80.75	78.00	59.50	54.00	50.08	29.08	28.00	11.50	176.75	174.00	152.75	149.08	148.00	131.50	128.75	126.00
A	d_{712}	112.50	107.00	103.08	88.50	85.75	83.00	64.50	59.00	55.08	34.08	33.00	19.25	13.75	11.00	157.75	154.08	153.00	136.50	133.75	131.00
T	d_{736}	115.25	109.75	105.83	91.25	88.50	85.75	67.25	61.75	57.83	36.83	35.75	19.25	16.50	13.75	160.50	156.83	155.75	139.25	136.59	133.75
S	d_{724}	134.50	129.00	125.08	110.50	107.75	105.00	86.50	81.00	77.08	56.08	55.00	38.50	35.75	33.00	11.75	176.08	175.00	158.50	155.75	153.00
U	d_{730}	136.50	131.00	127.08	112.50	109.75	107.00	88.75	83.00	79.08	58.08	57.00	40.50	37.75	35.00	35.00	13.75	10.08	9.00	157.75	155.00
N	d_{736}	139.25	133.75	129.83	115.25	112.50	109.75	91.25	85.75	81.83	60.83	59.75	43.25	40.50	37.75	16.50	12.83	11.75	163.25	160.50	157.75

TABLE 24 CREW BASED AT DACCA

3.6.3 COMPOSITE LAYOVER TIME MATRIX FOR KARACHI - DACCA SECTOR

		MONDAY			TUESDAY			WEDNESDAY			THURSDAY		FRIDAY			SATURDAY			SUNDAY		
		PK 731	PK 723	PK 737	PK 731	PK 735	PK 723	PK 731	PK 723	PK 737	PK 735A	PK 711	PK 731	PK 735	PK 723	PK 735	PK 735A	PK 719	PK 731	PK 735	PK 723
M O N	PK 724	158.50 D	153.00 D	12.50 K	27.08 K	29.83 K	32.58 K	51.08 K	56.58 K	60.50 K	80.08 D	79.00 D	62.50 D	59.75 D	57.00 D	35.75 D	32.08 D	31.00 D	14.50 D	11.75 D	9.00 D
	PK 730	1.08 K	155.00 D	10.50 K	25.08 K	27.83 K	30.58 K	49.08 K	54.08 K	58.50 K	79.50 K	80.58 K	64.50 D	61.75 D	59.00 D	37.75 D	34.08 D	33.00 D	16.50 D	13.75 D	11.00 D
	PK 738	2.00 D	164.50 D	1.00 K	15.58 K	18.33 K	21.08 K	39.58 K	45.08 K	49.00 K	70.00 K	71.08 K	74.00 D	71.25 D	68.50 D	47.25 D	43.58 D	42.50 D	26.00 D	23.25 D	20.25 D
T U E	PK 724	14.50 D	9.00 D	156.50 K	158.50 K	155.75 K	153.00 K	27.08 K	32.58 K	36.50 K	57.50 K	58.50 K	75.08 K	77.83 K	80.58 K	59.75 D	56.08 D	55.00 D	38.00 D	35.75 D	33.00 D
	PK 730	16.50 D	11.00 D	154.50 K	1.08 K	157.75 D	155.00 D	25.08 K	30.58 K	34.50 K	55.50 K	56.58 K	73.08 K	75.83 K	78.58 K	61.75 K	58.08 K	57.00 K	40.50 K	37.75 K	35.00 K
	PK 736	19.25 D	13.75 D	9.83 D	163.25 D	1.08 K	157.75 D	22.33 K	27.83 K	31.75 K	52.75 K	53.83 K	70.00 K	73.08 K	75.83 K	64.50 D	60.83 D	59.75 D	43.25 D	40.50 D	37.75 D
P R I	PK 724	38.50 D	33.00 D	9.08 D	14.50 D	11.75 D	9.00 D	158.50 D	153.50 D	12.50 K	33.50 K	34.58 K	51.08 K	53.83 K	56.58 K	77.83 K	80.08 D	79.00 D	62.50 D	59.75 D	57.00 D
	PK 730	40.50 D	35.00 D	31.08 D	16.50 D	13.75 D	11.00 D	1.08 K	155.00 D	10.50 K	31.50 K	32.58 K	49.08 K	51.83 K	54.58 K	75.83 K	79.50 K	80.58 K	64.50 D	61.75 D	59.00 D
	PK 738	50.00 D	44.50 D	40.58 D	26.00 D	23.25 D	20.50 D	2.00 D	164.50 D	1.00 K	22.00 K	23.08 K	39.58 K	42.33 K	45.08 K	66.33 K	70.00 K	71.08 K	74.00 D	71.25 D	68.50 D
T H U	PK 704	59.50 D	54.00 D	50.08 D	35.50 D	32.75 D	30.00 D	11.50 D	155.58 K	2.08 D	12.50 K	13.58 K	30.08 K	32.83 K	35.58 K	56.83 K	60.50 K	61.58 K	78.08 K	80.75 K	78.00 D
	PK 706	64.50 D	59.00 D	55.08 D	40.50 D	37.75 D	35.00 D	16.50 D	11.00 D	154.50 K	154.08 D	153.00 D	25.08 K	27.83 K	30.58 K	51.83 K	55.50 K	56.50 K	73.08 K	75.83 K	78.75 K
P R I	PK 724	75.08 K	80.58 K	77.08 D	62.50 D	59.75 D	57.00 D	38.50 D	33.00 D	29.08	153.50 K	154.58 K	158.50 D	155.75 D	153.00 D	29.83 K	35.50 K	34.58 K	51.00 K	53.83 K	56.75 K
	PK 730	73.08 K	78.58 K	79.08 D	40.50 D	61.75 D	59.00 D	40.50 D	35.00 D	31.08 D	10.08 D	9.00 D	1.08 K	157.75 D	155.00 D	27.83 K	31.50 K	32.58 K	49.08 K	51.83 K	54.75 K
	PK 736	70.33 K	75.83 K	79.75 K	67.25 D	64.50 D	61.75 D	43.25 D	37.75 D	33.83 D	12.83 D	11.75 D	160.50 D	1.08 K	157.75 D	25.08 K	28.75 K	29.83 K	46.33 K	49.08 K	52.00 K
S A T	PK 710	54.08 K	59.58 K	63.50 K	78.08 K	80.75 D	78.00 D	59.50 D	54.00 D	50.08 D	29.08 D	28.00 D	11.50 D	152.83 D	155.58 K	152.75 D	12.50 K	13.58 K	30.08 K	32.83 K	35.75 K
	PK 712	49.08 K	54.58 K	58.50 K	73.08 K	75.83 K	78.58 K	64.50 D	59.00 D	55.08 D	34.08 D	33.00 D	16.50 D	13.75 D	11.00 D	157.75 D	154.08 D	153.00 D	25.08 K	27.83 K	30.75 K
	PK 736	46.33 K	51.83 K	55.75 K	70.33 K	73.08 K	75.83 K	67.25 D	61.75 D	57.83 D	36.83 D	35.75 D	19.25 D	16.50 D	13.75 D	1.08 K	156.83 D	155.75 D	22.32 K	25.08 K	28.00 K
S	PK 724	27.08 K	32.58 K	36.50 K	51.08 K	53.83 K	56.58 K	75.08 K	80.58 K	77.08 D	56.08 D	55.00 D	38.50 D	35.75 D	33.00 D	11.75 D	153.50 K	154.08 K	158.50 D	155.75 D	153.00 D
	PK 730	25.08 K	30.58 K	34.50 K	49.08 K	51.83 K	54.58 K	73.08 K	78.58 K	79.08 D	58.08 D	57.00 D	40.50 D	37.75 D	35.00 D	13.75 D	10.08 D	9.00 D	1.08 K	157.75 D	155.00 D
	PK 736	22.33 K	27.83 K	31.75 K	46.33 K	49.08 K	51.83 K	70.33 K	75.83 K	79.75 K	60.83 D	59.75 D	43.25 D	40.50 D	37.75 D	16.50 D	12.83 D	11.75 D	163.25 D	1.08 K	157.75 D

TABLE 25 :COMPOSITE MATRIX

SECTION II

CHAPTER IV.

LINEAR PROGRAMMING.

4.1) Introduction.

Linear programming is one of the most important optimization techniques developed in the field of operational research. From a theoretical point of view, as well as a practical point of view, it is the most thoroughly explored field of optimization. Much has been written on operational research and especially in the field of linear programming. Only the outlines of linear programming will be explained.

In order to appreciate the full potentialities of linear programming it may be as well to look briefly at operational research. Operational research was actually born during the Second World War when civilian scientists were called in by Military Generals to assist in solving their tactical problems.

The first particular problem in linear programming was formulated in 1941 - transportation problem by Hitchcock(18). In 1947, the general problem of linear programming was formulated in precise mathematical terms by Professor George B.Dantzig in collaboration with Marshall K.Wood in the U.S.Department of the Airforce, who then constituted a group called Project SCOOP (Scientific Computation of Optimal Programmes). In this department the application of linear programming was of a military nature. The systematic procedure, the simplex method(8) was presented during a historic conference in 1941 on linear programming organized by the Cowles Commission for research into economics.

The term linear programming was suggested by Koopman to Professor Dantzig as an alternative to the earliest form "programming in a linear structure"

4.1) contd.

After 1948, paralleling the activity of the Cowles Commission, an increasing number of mathematicians, statisticians and economists had contributed individually or in groups to the development of linear programming, such as Dantzig, Ford and Fulkerson, Kuhn, Tucker, Charness and Cooper etc., In 1958, Gomory[13] presented an algorithm for integer solution to linear programming. Then another mathematician, Harris[16] presented an algorithm for solving mixed integer programmes. One of the important improvements was made in integer solution to linear programming by Healey[17] and Balas[1], who developed an algorithm for solving linear programming problems where the variables are restricted to zero and one value. Although a number of modifications and extensions of the simplex method have been developed, since then the original technique remains the most general approach to the solution of linear programming problem. Furthermore the range of application has been greatly extended by many recent developments in the field of computing equipment, especially with the development of high-speed electronic computers.

4.2) Definition.

The planning of activities such as allocation of men, material or machine is called programming. Programming problems occur in a situation where the available resources may be limited or incapable of being utilized completely. The object of this is to determine the most efficient method of allocating these resources to activities so that a measure of performance is optimized. Whenever this measure is a linear function of the controlable variables and the restriction on availability of

4.2) contd.

resources are expressible as a system of linear equations or inequalities. Such type of problem is known as linear programming problem.

4.3) Mathematical Formulation.

The general linear programming problem can be expressed as follows:-

Find the values of $x_1, x_2 \dots x_n$ which maximize or minimize the linear form

$$z = c_1 x_1 + c_2 x_2 + \dots + c_n x_n \qquad (4.1)$$

subject to the conditions

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n & \quad (\leqslant = \geqslant)\ b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n & \quad (\leqslant = \geqslant)\ b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n & \quad (\leqslant = \geqslant)\ b_m \end{aligned} \qquad (4.2)$$

and

$$x_j \geqslant 0 \quad (j = 1,2, \dots n) \qquad (4.3)$$

where a_{ij}, b_i, c_j are constants and c_j are the cost associated with each variable and z is known as objective function.

The set of equations (4.2) can be converted in equations (if they are inequalities) by the addition of further variables with their proper sign - called slack variables.

4.4) Applications of Linear Programming.

After the Second World War two important de-

4.5) contd.

velopments have been made for the improvement and advancement in management planning and control. One was the development of digital computers and the other was the application of higher mathematics, statistics and economics to the problem of industrial management. The former made possible rapid and economical manipulation of massive data and later provided the necessary theoretical framework for the organization and alaysis of these data. The result is that the complicated problems can be solved and made a part of the decisions on which the success of business depends.

Linear programming has wide applications in solving such diverse problems as determining the optimum utilization, and establishing its importance almost in every field as an aid to decision making in business, industry, Government etc., The distribution of facility or machine, diet problems, scheduling distribution, distribution of commodities are but a few examples of the type of problems which can be solved by linear programming. Briefly linear programming is a method of determining an optimum programme of interdependent activities in view of available resources.

Linear Programming technique is applied to solve the problem of crews scheduling where the variables are restricted to 0-1 values. This technique can be divided into three main groups

1) Assignment technique
2) Transportation technique
3) Integer programming technique (0-1)

Simplex method cannot be used because it may give

4.5) contd.

the value of variables in real number. However, sometime this method may give the values in integers

4.5.1) General Statement of the Problem.

The problem of crew scheduling of a commerical airline can be stated as follows:-

$$x_{11} + x_{12} + \ldots\ldots + x_{1n} = 1$$
$$x_{21} + x_{22} + \ldots\ldots + x_{2n} = 1$$
$$\vdots$$
$$x_{m1} + x_{m2} + \ldots\ldots + x_{mn} = 1$$

where each variable is associated with time (t_{ij}) away from their home base and t_{ij} is the layover time of the crews arriving on i^{th} flight and departing on j^{th} flight. Such a layover time is not less than eight hours in any case according to the flying rules.

Here the objective function is to minimise the total layover time at a certain place other than the home base - given by the equation

$$z = t_{11}x_{11} + t_{12}x_{12} + \ldots\ldots + t_{mn}\, x_{mn}$$

is minimum, subject to the condition that $x_{ij} \geq 0$ for all i and j. More specifically x_{ij} is equal to one or zero.

CHAPTER V.

THE TRANSPORTATION MODEL.

5.1) Introduction.

The transportation problem is a special case of linear programming problems in which the objective is to "transport" a single commodity from various "origins" to different "destinations" at a minimum total cost. It was first studied by Hitchcock[18] in 1941, then separately by T.C.Koopmans[28] and finally placed in the framework of linear programming by Professor Dantzig[6] in 1951. Since then, various people have worked in this field, with the result that a lot of improvement has been made. It is now accepted as one of the most important analytical and planning tools in business and industry.

There are two approaches to this problem: the general and specific. In the former the transportation is considered as a special case of linear programming problem - the simplex method. In the later approach some independent methods have been developed by various authors, whih are much simpler and efficient for solving the transportation problem.

5.2) Statement of the Transportation Problem.

In the classical Hitchcock transportation problem, a homogeneous commodity is available at m origins and needed at n destinations. The cost of transporting a single commodity from the i^{th} origin to the j^{th} destination is c_{ij}. The quantity a_i is available at the i^{th} origin and b_j is required at the j^{th} destination is fixed and assumed that

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

5.2) contd.

that is the total supply is equal to the total demand.

In case of crew scheduling, the origins represent the crews and destinations represent the departures of the crews. For safety reasons the airline company have to replace the crews in the longer flights somewhere between the terminal cities. During this layover time the company is responsible for their meals and accommodation until their next assignment. Therefore, this layover time is associated with the cost on the crews, and t_{ij} represents the arrival of the crews on i^{th} flight and departure on the j^{th} flight.

Here the objective function is to minimize this layover time of the crews at a place other than the home base,

$$z = t_{11} x_{11} + t_{12} x_{12} + \dots + t_{1n} x_{1n} + t_{21} x_{21} + \dots + t_{mn} x_{mn}$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{n} t_{ij} x_{ij} = \text{minimum} \qquad (5.21)$$

subject to the conditions

$$\sum_{i=1}^{m} x_{ij} = b_j \qquad j = 1 \dots n \qquad (5.22)$$

$$\sum_{j=1}^{n} x_{ij} = a_i \qquad k = 1 \dots m \qquad (5.23)$$

and

$$x_{ij} \geq 0 \qquad \text{for all } i, j \qquad (5.24)$$

where x_{ij} represents the arrival of the crew on the i^{th} flight and departure on the j^{th} flight after due test.

5.2) contd.

Also a_i, b_j, m, n are all positive integers and t_{ij} is a real number.

A set of quantities x_{ij} satisfying the last three constraints will be called a feasible solution and will be optimal if also satisfying (5.1). The nature of the problem is such that there exists at least one optimal solution.

5.3) Structure of the Transportation Problem.

The important feature of the transportation problem is that it can be shown in the rectangular arrays, which show the value of x_{ij} and t_{ij} in row i and column j. The total of each row and column is presented by a_i and b_j respectively.

The following table shows the structure of the transportation problem.

		Destinations Departures			Crews available total
		D_1	D_2	.... D_n	
	A_1	t_{11} x_{11}	t_{21} x_{12}	.... t_{1n} x_{1n}	a_1
Arrival	A_2	t_{21} x_{21}	t_{22} x_{22}	.... t_{2n} x_{2n}	a_2
	⋮	⋮	⋮	⋮	⋮
	A_n	t_{n1} x_{m1}	t_{n2} x_{m2}	.... t_{mn} x_{mn}	a_m
Crews required total		b_1	b_2	.... b_n	

5.3) contd.

The square (i,j) contains t_{ij} in its upper left hand corner and x_{ij} in its lower right hand corner. Down along the right and across the bottom squares there are some other values a_i and b_j showing the total of each row and column respectively.

Now the problem is to place m+n-1 variables into these cells. If any box has no number then it means that this cell has zero values. All such types of cells are known as non-basic cells and occupied cells are known as basic cells. During the procedure if the occupied cells are not equal to m+n-1, then zero valued basic variables are indicated by a zero entry in order to distinguish from non-basic variables. During the algorithm, the rows total and columns total must be mintained in order to satisfy the equations

Row equations $$\sum_{j=1}^{n} x_{ij} = a_i \qquad i = 1 \ldots m \qquad (5.31)$$

Column equations $$\sum_{i=1}^{m} x_{ij} = b_j \qquad j = 1 \ldots n \qquad (5.32)$$

and

$$\sum_{i} a_i = \sum_{j} b_j = \sum x_{ij} \qquad (5.33)$$

5.4) Degenerate Case.

The degeneracy occurs in the transportation problem if a partial sum of a_i is equal to a partial sum of b_j, then at least one of the basic variables is equal to zero. This degeneracy can be avoided by the addition of a very small quantity ϵ to any row subject to the following rules:-

5.4) contd.

1) $\epsilon < x_{ij}$ for all $x_{ij} > 0$

2) $\epsilon + 0 = \epsilon$

3) $x_{ij} \pm \epsilon = x_{ij}$ $x_{ij} > 0$

5.5) General Technique of Solution.

As stated earlier that the transportation model is a special case of general linear programming model. The basic problem can be formulated and solved by the simplex technique, but some other independent methods have been developed in this field which are much simpler and faster in computation. Some of the important methods are

1) North-Western Corner rule(6)
2) Vogel's approximation method(2)
3) Stepping-Stone method(4)
4) Modified-distribution method (MODI)(6)
5) Least cost method(7)
6) Ford-Fulkerson method(11) etc.,

Now the initial basic feasible solution for a given transport problem can be obtained by any one of the methods described above. The least cost method is advisable to find a good starting feasible solution and number of iterations for obtaining the optimal solution is less than compared to other methods.

The outline of this method is that allocation is made to that cell whose cost per unit is lowest. This lowest cost is loaded as much as possible in view of the origin capacity of its row and the destination requirements of its column. Then move to next lowest cell and make an allocation

5.5) contd.

as large as consistent with row and column total.

Continue this process until m+n-1 entries are made subject to the satisfaction of rows and columns total.

5.6) Determining the fictitious costs.

The first feasible solution can be set up by any one of the methods described above. Now the problem is how to find out whether a basic feasible solution is optimal and if it is not, how to improve the feasible solution.

The fictitious costs are calculated to test the feasible solution. It is convenient to denote the i^{th} row equations and j^{th} column equations by u_i and v_j respectively. Since one equation is redundant, any row or column having a greater number of basic variables sets its corresponding cost at zero. The remaining u_i and v_j are calculated by the relation

$$c_{ij} = u_i + v_j \qquad \text{if } x_{ij} \text{ is a basic.}$$

Then every unit cost c_{ij} is compared with the sum of the fictitious costs of its row and column. If their differences are all non-negative i.e.

$$\bar{c}_{ij} = c_{ij} - u_i - v_j$$

for every cell, then the solution is optimal and the problem is finished.

If some values of $\bar{c}_{ij}$ are negative, then a non-basic variable is replaced by one of m+n-1 basic variables, which is dropped from the basic set and becomes another non-basic variable.

The new basic variable is selected by this relation

5.6) contd.

$$c_{st} - u_s - v_t = \underset{i,j}{\text{Min}} \left[(c_{ij} - u_i - v_j) < 0 \right]$$

The symbol θ is entered in cell (s,t) to indicate that a value θ will be given to non-basic variable x_{st} and some entries are adjusted by additions and subtractions of θ in order to balance the rows and columns total and leave others unchanged. The smallest entry with minus θ will determine the value of θ so that the number of basic variables remain the same i.e., m+n-1. The new improved feasible solution so obtained is checked and the same process is repeated if it is not optimal.

5.7) <u>Computer Program</u>.

A computer program is developed from the procedure(15) to speed up the calculations and attached in the Appendix (Program No.5). This program is run successfully on ICL 1900 and it took only 30 seconds to solve the problem of table 25 (20×20 matrix).

In order to show the method for the preparation of data cards, Table 7 is considered. In this example all the rows and columns totals are equal to unity.

(City X)

800000

6

6

167.25	142.50	120.17	133.67	85.50	41.25
24.50	167.75	145.42	158.92	110.75	66.50
46.33	21.58	167.25	180.75	132.58	88.33
32.91	176.16	153.83	167.33	119.16	74.91
81.08	56.33	34.00	47.50	167.33	123.08
125.33	100.58	78.25	91.75	43.58	167.33
1	1	1	1	1	1
1	1	1	1	1	1

<u>Table 29</u>. Data cards for City X.

CHAPTER VI.

THE ASSIGNMENT MODEL

6.1) Introduction.

The assignment model deals with a special case of linear programming problems in which the objective is to assign a number of origins to the same number of destinations at the minimum total cost. The assignment is to be made on a one-to-one basis. That is, each origin can associate with one and only one destination. This fact shows two characteristics in a linear programming problem which give rise to an assignment problem. First, the number of columns is equal to the number of rows in a matrix - nxn matrix. Secondly, the optimal solution for the problem is such that there can be one and only one assignment in a given row or column of the matrix.

All the elements of the matrix are assumed to be known and independent of each other. Now the problem is to choose the elements of the matrix in such a way as to optimize the objective function. In other words, the assignment problem is a problem of proper matching between the origins and the destinations. A value 1 is assigned to those cells for which a match has been made; a value zero is assigned to the remaining cells. This means that there are only two values 'one and zero' which are to be assigned.

6.2) The Assignment Problem - As a Special Case of Transportation Problem.

As mentioned above, the assignment problem is a special case of the general linear programming problem. As a matter of fact the assignment problem is a special case of the transportation problem, which in turn is itself a special case of the general linear programming problem.

6.2) contd.

In the transportation problem, degeneracy occurs whenever one or more basic variables are zero. The transportation problem is equivalent to the assignment problem, in case, exactly n basic variables must receive unit value and the remaining n-1 basic variable should all be zero. Therefore, the assignment problem can be considered a complete degenerate form of the transportation problem.

6.3) Definition.

In the assignment problem, there is a group of men and a number of jobs, each man is assigned to only one job and each job requires only one man. The arrangement should be done in such a way that given measure of effectiveness can be optimized.

In the problem of crew scheduling, each crew can be assigned to any flights $f_1 \ldots f_n$ and each flight requires any one of the crews $c_1 \ldots c_n$, the permutation may be done in such a way that the layover time

$$\sum_{i=1}^{n} \sum_{j=1}^{n} t_{ij} x_{ij}$$

is minimum. Evidently there are n ways in which to choose f_1, (n-1) ways remaining to choose $f_2 \ldots f_n$ or $n! = n(n-1) \ldots 3.2.1$ different possible arrangements.

It is possible for small n to enumerate all possible arrangements and select the arrangements having minimum value. As the size of the matrix increases, the number of possible arrangements also increases.

6.3) contd.

e.g. 6! = 720, 10! = 3,628,800

which is manually impossible to enumerate all arrangements even on present day electronic computers, it is not practicable to enumerate all the permutations. Hence the problem arises to find an efficient algorithm for obtaining an optimal assignment in minimum time.

6.4) <u>Mathematical Model.</u>

The mathematical statement of the assignment problem is to choose a set of n independent elements from a square matrix of order n subject to the condition that no two elements lie in the same line and the sum of these elements is minimum.

Mathematically it can be stated as follows.

Given n^2 matrix $T = \| t_{ij} \|$ as time matrix with

$t_{ij} \geqslant 0 \quad i,j = 1,2 \dots n$

It is to find an n^2 matrix $x = \| x_{ij} \|$ as assignment or permutation non-negative matrix and the value of each element is zero or 1, such that

$$x_{ij} = x^2_{ij} \quad i,j = 1 \dots n \qquad (6.41)$$

$$\sum_{j=1}^{n} x_{ij} = \sum_{i=1}^{n} x_{ij} = 1 \text{ for all } i,j \text{ and } n \geqslant 2 \qquad (6.42)$$

and the object function

$$z = \sum_{i=1}^{n} \sum_{j=1}^{n} t_{ij}\, x_{ij} \text{ is minimum} \qquad (6.43)$$

The equations (6.41) and (6.42) are the basic conditions which jointly specify that

6.4) contd.

(a) $x_{ij} = \begin{cases} 1 \text{ if } i^{th} \text{ crew is assigned to the } j^{th} \text{ flight} \\ \vdots \\ 0 \text{ otherwise} \end{cases}$ (6.44)

(b) In the permutation or the assignment matrix, each row and column will have only one element unity and all other elements are zero.

If the condition (6.41) is relaxed as $x_{ij} \geq 0$ then it will become the transportation problem

6.5) <u>Solution to the Assignment Problem</u>.

Several methods have been suggested by various authors, some of the important methods are presented by Flood[10], Ford and Fulkerson [11], James Munkers[19], Kuhn[21] etc.,

Kuhn was the first person who developed a computational algorithm for solving the assignment problem based on the Konig theorem as stated by Ergervary.

Konig theorem: "If the elements of a matrix are partly zero and partly numbers different from zero, then the minimum number of lines that contain all the non-zero elements of the matrix is equal to the maximum number of non-zero elements that can be chosen with no two on the same line."

In this theorem a line means a row or a column of a matrix. This theorem, together with the following two theorems is the basis for Kuhn's algorithm for solving the assignment problem.

The first theorem proves that the solution is

6.5) contd.

unchanged if we add or subtract a constant to any one row or column of t_{ij} matrix. It can be stated more precisely as follows:

Theorem I.

If $x_{ij} = X_{ij}$

minimizes $$Z = \sum_{i=1}^{n} \sum_{j=1}^{n} t_{ij} x_{ij} \quad \text{for all } x_{ij}$$

such that $x_{ij} \geq 0$

and $$\sum_{i=1}^{n} x_{ij} = \sum_{j=1}^{n} x_{ij} = 1,$$

then $x_{ij} = x_{ij}$

also minimizes

$$Z' = \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} t'_{ij}$$

where $t'_{ij} = t_{ij} - u_i - v_j$ for all $i,j = 1 \ldots n$

Theorem II.

If all $t_{ij} \geq 0$

then $$Z = \sum_{i=1}^{n} \sum_{j=1}^{n} t_{ij} x_{ij} \geq 0$$

If some $t_{ij} = 0$ and find a set of x's that are all zero except perhaps where $t_{ij} = 0$, then it is optimal, for the corresponding

6.5) contd.

$$\sum_{i=1}^{n} \sum_{j=1}^{n} t_{ij} x_{ij} = 0$$

Second theorem is obvious to prove the first

$$Z' = \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} (t_{ij} - u_i - v_j)$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} t_{ij} x_{ij} - \sum_{i=1}^{n} u_i \sum_{j=1}^{n} x_{ij}$$

$$- \sum_{j=1}^{n} v_j \sum_{i=1}^{n} x_{ij}$$

$$= Z - \sum_{i=1}^{n} u_i - \sum_{j=1}^{n} v_j$$

M.Flood has outlined this method in such a way that it gives a rapid result as compared to all other available methods for hand computation. The algorithm is as follows:-

Step (i) Calculate the $\sum_j \min_i a_{ij}$ and $\sum_i \min_j a_{ij}$. The transformation should be started according to the greater value of sum of row minima or column minima. Subtract the smallest element of the line from each line of A, obtaining a reduced matrix A_1 with non-negative elements and at least one null element in each line. Here line means row or column.

6.5) contd.

Step (ii) Find the minimal set S_1 of lines, n_1 in number, which contain all null elements of A_1. If $n_1 = n$, then there is a set of n independent zero elements and the elements of A in these n positions constitute the required solution.

Step (iii) If $n_1 < n$, let h_1 denote the smallest element of A_1 which is not in any line of S_1. Then $h_1 > 0$. For each line in S_1, add h_1 to every element of that line i.e. intersection element of row and column, then subtract h_1 from every element of A_1, not in S_1. Call this new reduced matrix A_2.

Step (iv) Repeat steps (ii) and (iii), using A_2 in place of A_1, until at stage 'k' $n_k = n$. The process will terminate after a finite number of steps. The sum of elements of the new matrix is decreased by

$$n(n - n_k)h_k \quad \ldots \qquad (6.45)$$

By applying this technique it is quite possible to obtain more than one optimal solution, but numerically all have the same value.

6.6) Numerical Example:-

This technique is applied to the layover time matrix Table 7 of City X. Now the problem is to find pairing of flights, so that the time away from the home base may be minimized. Flood's technique gave the following optimal solution.

6.6) contd.

IN	OUT	LAYOVER TIME
a_3	d_2	85.50
a_2	d_4	24.50
a_4	d_6	21.58
a_5	d_1	74.91
a_1	d_3	47.50
a_6	d_5	78.25
TOTAL TIME =		332.24 HRS.

Table 30. Crew's Schedule at City X.

6.7) Computer Program.

The computer program No.7 is developed in Fortran and attached in the Appendix. This program is quite useful for large matrices and time saving. The program is run successfully on ICL 1900 and took only 14 seconds to find the optimal solution from 20 × 20 matrix and in case of the smaller matrix it took only a few seconds, The data cards are prepared as follows:-

(CITY X)
6

167.25	142.50	120.17	133.67	85.50	41.25
24.50	167.75	145.42	158.92	110.75	66.50
46.33	21.58	167.25	180.75	132.58	88.33
32.91	176.16	153.83	167.33	119.16	74.91
81.08	56.33	34.00	47.50	167.33	123.08
125.33	100.58	78.25	91.75	43.58	167.33

Table - 31. Data cards for City X.

The computer will give the result in the column

6.7) contd.

vector form (5,1,2,6,4,3). The sum of these elements is 332.24 hours which is the same as in the above example. There may be more than one optimal solution having the same numerical value.

CHAPTER VII

INTEGER PROGRAMMING.

7.1) Introduction.

In this Chapter integer programming is applied to the scheduling of aircraft crews of a commercial airline to minimize the layover time of the crews at slip stations other than the home base. In this case the values of the variable are restricted to 0 and one only. If the crew is assigned to any flight, then the value is one and zero if the crew is not assigned to any flight. In mathematical notation it can be written as

$$x_{ij} = \begin{cases} 1 & \text{if } i^{th} \text{ crew is assigned to } j^{th} \text{ flight} \\ 0 & \text{otherwise} \end{cases}$$

Several methods are available for solving the linear programming problem where the variables are restricted to integers. First of all R.E.Gomory[11,12] presented a systematic procedure for solving such a problem. Further improvement was made in this field by the following men by presenting a procedure when the variables are restricted to zero and one value only.

In 1964, Healey[17] presented a method. Its mephasis is on problems in which the sum of non-negative integer valued variables must add up to one. In fact the integer constrained variables should belong to disjoint sets [X], [Y], [Z] etc. such that $\Sigma X_i = 1$, $\Sigma Y_i = 1$ etc., hence the name Multiple Choice Programming.

He does not give any proof of convergences. Thus the theoretical status of the method is unclear. First, the problem is to be solved by simplex method, if the solution is in integer form (0-1), then the process is terminated. Otherwise some modifications have to be made.

7.1) contd.

In 1965, Balas[1] proposed an algorithm of solving the linear programming problem with 0-1 variables. It consisted of a systematic procedure of successively assigning to certain variables the value 1. After few iterations one gets either an optimal solution or the evidence that no feasible solution exists. This algorithm requires the additions and subtraction only. Hence the name "Additive Algorithm".

In 1967, Geoffrion[12] presented a paper in which he described the simplified version of Balas additive algorithm. This method will be applied to solve the crews scheduling.

7.2) Geoffrion Method.

Minimise c_x subject to $b + A_x \geqslant 0$ (7.2.1)

and $x_j = 0$ or 1 where c is an n-vector, b and 0 are m-vectors, A is an m × n matrix and x is a binary n-vector to be chosen. A solution that satisfies the constraints $A + b_x \geqslant 0$ will be called a feasible solution and a solution that minimizes c_x over all feasible solutions will be called an optimal feasible solution.

Before explaining the procedure for solution, it is necessary to explain a few preliminary definitions. A partial solution S is defined as an assignment of binary values to a sub-set of n variables. A variable which is not assigned a value by S is called free variable. Also the symbol j denotes $x_j = 1$ and symbol -j denotes $x_j = 0$. For example if n = 5 and

$$S = [3, 2, -1]$$

then it means that $x_3 = 1$, $x_2 = 1$, $x_1 = 0$ and x_4 and x_5 are free. The order of elements of S are written according to the order in which the elements are generated. A completion of a partial

7.2) contd.

solution S is defined as a solution that is determined by S together with a binary value of the free variables. The four possible completions of S are given by

$$(0 \quad 1 \quad 1 \quad 0 \quad 0)$$
$$(0 \quad 1 \quad 1 \quad 1 \quad 0)$$
$$(0 \quad 1 \quad 1 \quad 0 \quad 1)$$
$$(0 \quad 1 \quad 1 \quad 1 \quad 1)$$

The number of different completions are determined by 2^{n-s}. In the above example $2^{5-3} = 4$ solutions. When there are no free variables, then there is only one completion of S.

During the algorithm, a sequence of partial solutions are generated and considered all completions. Some feasible solutions are obtained and the best feasible completion of S is stored as an incumbant. If a better feasible solution is obtained as compared to previously stored solution, then the new feasible solution is stored by replacing the previous one. The other possibility is that S has no feasible completion better than the incumbant. In either case, S is fathomed.

The problem is started with $S^o = \phi$ where ϕ indicates the empty set. If S^o is fathomed, the problem is finished - either there is no feasible solution or the best feasible solution can be found. The infinity is taken as the value of Z, unless some better upper bound is found.

Geoffrion has simplified Balas algorithm and it is represented in the following flow diagram

7.2) contd.

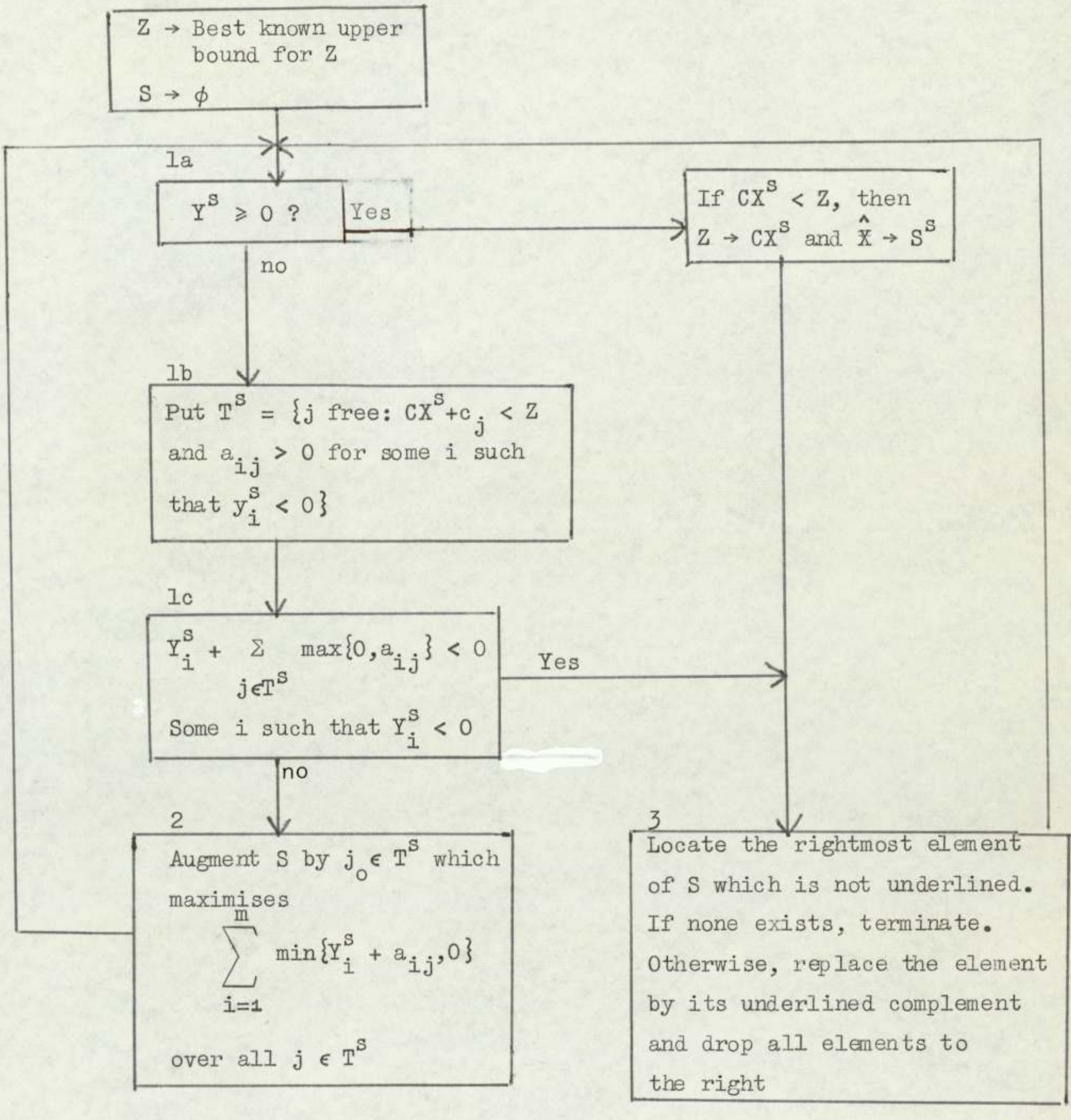


Diagram - 6 A simplified version of Balas additive algorithm.

7.3) <u>Example.</u>

The above technique is applied to the layover time matrix for Cairo(Table 21) for allocating the crews to flights, so that the layover time of the crews is minimised.

7.3) contd.

Minimise

$$167.16x_1 + 119.16x_2 + 33.17x_3 + 153.16x_4 + 47.16x_5 + 167.16x_6$$
$$81.16x_7 + 33.17x_8 + 133.33x_9 + 85.33x_{10} + 167.33x_{11} + 119.33x_{12}$$
$$+ 13.34x_{13} + 133.33x_{14} + 47.33x_{15} + 167.33x_{16}$$

subject to the conditions

$$\begin{array}{lllll} -1+x_1+x_2+x_3+x_4 & & & & = 0 \\ -1 & +x_5+x_6+x_7+x_8 & & & = 0 \quad (7.3.1) \\ -1 & & +x_9+x_{10}+x_{11}+x_{12} & & = 0 \\ -1 & & & +x_{13}+x_{14}+x_{15}+x_{16} & = 0 \end{array}$$

and

$$x_j = 0 \text{ or } 1$$

Applying the above procedure, put

$$Z = \infty$$

$$S^0 = \phi \text{ (empty set)}$$

1a $y = (-1,-1,-1,-1) \not\geq 0$

1b $T = [x_1, x_2 \; \ldots\ldots \; x_{16}]$

1c $i = 1 : \; -1 + 4 \geq 0$

$i = 2 : \; -1 + 4 \geq 0$

$i = 3 : \; -1 + 4 \geq 0$

$i = 3 : \; -1 + 4 \geq 0$

2 $j = 1 \quad : -3$

⋮

$j = 16 \quad : -3$

As all x_j have the same value, any one can be selected for the next iteration. x_{13} is selected because it has minimum layover time.

Hence $S^1 = [13]$

Substituting the value of $x_{13} = 1$ in a set of equations

7.3) contd.

(7.3.1) it will become

$$
\begin{aligned}
-1 &+ x_1+x_2+x_3+x_4 &&= 0\\
-1 &+ x_5+x_6+x_7+x_8 &&= 0\\
-1 &+ x_9+x_{10}+x_{11}+x_{12} &&= 0\\
0 &+ x_{14}+x_{15}+x_{16} &&= 0
\end{aligned}
$$

Therefore

1a $y = (-1,-1,-1,0) \ngeq 0$

1b $T = (x_1, x_2 \; \ldots\ldots \; x_{12})$

1c $i = 1 \; : \; -1 + 4 \geq 0$

$i = 2 \; : \; -1 + 4 \geq 0$

$i = 3 \; : \; -1 + 4 \geq 0$

2 $j = 1 \; : \; -2$

$\vdots$

$j = 12 \; : \; -2$

Again any x_j can be selected. However, x_3 is selected due to minimum layover time.

Hence

$$S^2 = [13,3]$$

Substituting the value of x_{13} and $x_3 = 1$ in the set of equations (7.3.1) it will take the form of

$$
\begin{aligned}
& x_1+x_2+x_4 &&= 0\\
-1 &+ x_5+x_6+x_7+x_8 &&= 0\\
-1 &+ x_9+x_{10}+x_{11}+x_{12} &&= 0\\
&+ x_{14}+x_{15}+x_{16} &&= 0
\end{aligned}
$$

1a $y = (0,-1,-1,0) \ngeq 0$

1b $T = (x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12})$

1c $i = 2 \; : \; -1 + 4 \geq 0$

$i = 3 \; : \; -1 + 4 \geq 0$

7.3) contd.

2 $j = 5, \ldots\ldots 12 = -1$

Hence

$$S^3 = [13,3,5]$$

Substituting the values of $x_{13}, x_3, x_5 = 1$ in a set of equations (7.3.1) gives

$$\begin{array}{llllll} & x_1+x_2+x_4 & & & & = 0 \\ & & +x_6+x_7+x_8 & & & = 0 \\ -1 & & & +x_9+x_{10}+x_{11}+x_{12} & & = 0 \\ & & & & +x_{14}+x_{15}+x_{16} & = 0 \end{array}$$

1a $y = (0,0,-1,0) \ngeq 0$

1b $T = [x_9, x_{10}, x_{11}, x_{12}]$

1c $i = 3 \quad : \quad -1+4 \geq 0$

2 $j = 9 \ldots\ldots 12 = 0$

Hence

$$S^4 = [13,3,5,10]$$

$$y = (0,0,0,0) = 0$$

$\therefore$ S^4 is fathomed

$$C^*X^S = Z' = 13.34 + 33.17 + 33.17 + 85.33 = 165.00 < Z$$

$$\hat{x} = (0,0,1,0,1,0,0,0,0,0,1,0,0,1,0,0,0)$$

3 $S^5 = (13,3,5,-\underline{10})$

Substituting the values of $x_{13} = 1$, $x_3 = 1$, $x_5 = 1$ and $x_{10} = 0$ in equations (7.1.1) gives

$$\begin{array}{llllll} & x_1+x_2+x_4 & & & & = 0 \\ & & +x_6+x_7+x_8 & & & = 0 \\ -1 & & & +x_9+x_{11}+x_{12} & & = 0 \\ & & & & +x_{14}+x_{15}+x_{16} & = 0 \end{array}$$

7.3) contd.

1a $y = (0,0,-1,0) \ngeq 0$

1b $T = (x_9, x_{11}, x_{12})$

1c $i = 3 : -1+3 \geqslant 0$

2 $j = 9 : -1+1 = 0$

$j = 11 : -1+1 = 0$

$j = 12 : -1+1 = 0$

Here maximum value is selected, but all j's are equal. However x_{12} is selected due to less layover time. Hence

$$S^6 = (13,3,5,-\underline{10},12)$$

$$y = (0,0,0,0) = 0$$

$$\therefore \quad S^6 \text{ is fathomed}$$

$$CX^5 = 13.4 + 33.17 + 33.17 + 0 + 119.33$$

$$= 212.99 > Z'$$

This value of the objective function is higher than the previously incumbant value. Terminate the process and the optimum solution is

$$(0,0,1,0,1,0,0,0,0,1,0,0,1,0,0,0)$$

with 165.00 as the value of the objective function.

7.4) <u>Computer Program</u>.

A computer program No.6 is developed from the procedure [3] and attached in the Appendix. This program is quite useful when the number of rows is less than the number of columns. The data cards for this program are prepared while considering the above example.

7.4) contd.

8000000
4
16
167.16 119.16 33.17 153.16 47.16 167.16 81.16 33.17 133.33
85.33 167.33 119.33 13.33 133.33 47.33 167.33

1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
1	1	1	1												

Table - 33 Data cards for Cairo

CHAPTER VIII

SOLUTION OF LAYOVER TIME MATRICES AND SELECTION OF BASES

8.1) Introduction

In Section II it is shown that any method of Linear Programming can be used to solve the problem of aircraft crews of a commercial airline. Here the question arises as to which method is efficient and gives the best result in the minimum of time. A problem was taken and solved manually by three methods of Linear Programming. The assignment technique took less time as compared with other methods.

Then three computer programs were applied to Table 25 for comparison. The size of time matrix was 20 × 20. The computer gave the result as follows:-

Assignment program	=	14	seconds
Transportation program	=	30	seconds
Integer program	=	28	seconds

In addition to this time, the last method required 3 hours to punch the data cards. As it is shown the assignment program took less time compared to others. So this method will be used to solve the problem of crew scheduling and also took less time for punch the data cards.

8.2) Western Sector.

In this Sector first of all, the data cards for layover time matrices Tables Nos. 9-17 - that is all the combinations of Istanbul and Beirut are prepared. The computer program No.7 is applied to calculate the minimum layover time for each city and each combination and is shown in the following table.

The combinations Nos. 2, 3 and 5 have the same layover time. However, combination No.3 is selected for OR Schedule.

8.2.1)

Combinations at Beirut and Istanbul

BEIRUT		ISTANBUL		TOTAL LAYOVER TIMES IN HRS.
Flight No. PK.	Min.Layover Time in Hrs.	Flight Number.	Min.Layover	
1) 701,703,707,702, 708,714	332.24	705,709,713,715,721 722,706,712,716,718	497.98	830.22
2) 701,703,707,713, 721,702,706,708, 714,718	329.50	705,709,715,722,712 716	332.64	662.14
3) 701,703,707,713 702,708,714,718	330.83	705,709,715,721,722 706,712,716	331.30	662.13
4) 701,703,707,721, 702,706,708,714	330.83	705,709,715,713,722, 712,716,718	490.24	821.07
5) 701,703,707,713, 702,708,714,706	330.83	705,709,715,721,722, 712,716,718	331.30	662.13
6) 701,703,707,721, 702,708,714,718	330.83	705,709,715,713,722, 712,716,706	499.38	830.21

Table No.34 Combinations of Beirut and Istanbul

8.3.1) Crew Schedule at Istanbul

PK 705,709,715,721,722,706,712,716

	Crew		Layover time in hours.
	In	Out	
From Karachi	PK 705	PK 709	31.41
	PK 709	PK 715	39.50
	PK 715	PK 721	31.66
	PK 721	PK 706	77.66
From London			
	PK 722	PK 705	33.25
	PK 706	PK 712	47.58
	PK 712	PK 716	23.41
	PK 716	PK 722	47.41
Total layover time			331.30 hrs

Table No.35 Crew's Schedule at Istanbul

8.3.2) Crew Schedule at Beirut.

PK 701,703,707,713,702,708,714,718.

	Crew		Layover time in hours.
	In	Out	
From Karachi	PK 701	PK 703	24.50
	PK 703	PK 707	21.58
	PK 707	PK 713	48.50
	PK 713	PK 718	54.17
From London			
	PK 702	PK 708	47.50
	PK 708	PK 714	43.58
	PK 714	PK 701	41.25
	PK 718	PK 702	49.75
Total layover time			330.83 hours

Table No.36 Crew's Schedule at Beirut.

8.3.3) Crew Schedule at Cairo.

PK 717, 719, 704, 710

	Crew		Layover time in hours.
	In	Out	
From Karachi			
	PK 711	PK 710	13.34
	PK 719	PK 704	85.33
From London			
	PK 704	PK 711	33.16
	PK 710	PK 719	33.17
Total layover time			165.00

Table No.37 Crew Schedule at Cairo

8.4.) London.

In the Western Sector London is a terminal point. The crew schedules at Istanbul, Beirut and Cairo show that the crews of flights Nos. PK 722/PK 705, PK 710/PK 719, PK 704/PK 711 and PK 714/PK 704 should be based at London. Therefore, the layover time matrix of Table 22 is reduced to the following table and the result of this matrix is presented in Table No.39.

8.4.1) LONDON SCHEDULE

	a_{703}	a_{707}	a_{709}	a_{713}	a_{715}	$a\text{-}A_{717}$	$a\text{-}B_{717}$	a_{721}
d_{702}	162.17	140.25	113.83	89.67	73.58	67.33	67.33	41.70
d_{706}	18.17	164.25	137.83	113.67	97.58	91.33	91.33	65.70
d_{708}	42.17	20.25	161.83	137.67	121.58	115.33	115.33	89.70
d_{712}	66.17	44.25	17.83	161.67	145.58	139.33	139.33	113.70
d_{716}	90.17	68.25	41.83	185.67	169.58	163.33	163.33	137.70
d_{718}	110.42	88.50	62.08	37.92	21.83	183.58	183.58	157.95
$d\text{-}A_{720}$	114.17	92.25	65.83	41.67	25.58	187.33	187.33	161.70
$d\text{-}B_{720}$	114.17	92.25	65.83	41.67	25.58	187.33	187.33	161.70

Table No.38 Layover time matrix for London.

Crew		Layover time in hours
In	Out	
PK 703	PK 708	42.17
PK 707	PK 712	44.25
PK 709	PK 720-A	65.83
PK 713	PK 720-B	41.67
PK 715	PK 718	21.83
PK 717-A	PK 706	91.33
PK 717-B	PK 716	163.33
PK 721	PK 702	41.70
Total layover time		512.11 hrs.

Table No.39 Crew's Schedule at London.

8.4.1) contd.

The crew arriving on flight No. PK 717 is entitled to have at least 72 hours rest according to the agreement between the Crews' Association and Airline Authority. This condition is fulfilled. In the computer result it is shown that the crew arriving on PK 717-B will get a rest of 163.33 hours. During this period, the crew can go to Cairo on flight No.PK 704 and come back on PK 711. This means the only crews on flight Nos. PK 722/PK 705, PK 710/PK 719 and PK 714/PK 701 should be based at London. The minimum layover time at London is reduced to

512.11 hours - 50.42 hours = 461.69 hours

All the above statements are consolidated into two tables only showing crews based in Karachi and London respectively.

8.4.1) contd.

TABLE 40-1 OR Schedule with two bases and without crew slip at MOSCOW

KARACHI BASE

MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON	
701 K-B	703 B-L	L	708 L-B	B	714 B-K										
	703 K-B	707 B-L	L	712 L-I	716 I-K										
		705 K-I	709 I-L	L	L	720 L-M-K									
		707 K-B	B	713 B-L	L	720 L-M-K									
			709 K-I	I	715 I-L	718 L-B	B	702 B-K							
				711 K-C	710 C-K										
				713 K- B	B	718 B-K									
					715 K-I	721 I-L	L	702 L-B	B	708 B-K					
					717 K-M-L	L	L	L	706 L-I	I	712 I-K				
					717 K-M-L	L	L	L	704 L-C	C	711 C-L	716 L-I	I	722 I-K	
						719 K-C	C	C	704 C-K						
						721 K-I	I	I	706 I-K						

8.4.1) contd.

LONDON BASE

MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON
722 L-I	I	705 I-L												
				710 L-C	C	719 C-L								
					714 L-B	B	701 B-L							

Table 40-2 OR Schedule for London

8.4.1) contd.

Conclusion (Western Sector only)

Mathematical technique is applied to frame the schedule which is free from human bias when this OR Schedule is compared with the P.I.A.Schedule (Table 41) a net saving of 351.91 hours per week is shown

P.I.A.Schedule	=	1640.73	hours
OR Schedule	=	1288.82	hours
Difference saved per week	=	351.91	hours

8.4.3) contd.

TABLE 41:- PIA SCHEDULE - CREW BASED AT KARACHI

MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON
701 K-B	703 B-L	L	L	L	714 L-B	718 B-K								
	703 K-B	707 B-L	L	710 L-C	C	719 C-L	722 L-I	I	706 I-K					
		705 K-I	709 I-L	L	L	720 L-K								
		707 K-B	B	713 B-L	L	718 L-B	B	702 B-K						
			709 K-I	I	715 I-L	L	L	702 L-B	B	708 B-K				
				711 K-C	710 C-K									
				713 K-B	B	B	701 B-L	L	L	L	712 L-I	716 I-K		

8.4.3) contd.

TABLE 41 (contd)

MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON
					715 K-I	721 I-L	L	L	L	708 L-B	B	714 B-K		
					717 K-L	L	L	L	704 L-C	C	711 C-L	716 L-I	I	722 I-K
					717 K-L	L	L	L	706 L-I	I	712 I-K			
						719 K-C	C	C	704 C-K					
						721 K-I	I	I	705 I-L	L	L	L	720 L-K	

8.5) O.R.Schedule with one home base - KARACHI

When this OR Schedule was shown to the Chief Scheduling Officer of P.I.A. he agreed and said that it fulfilled all the requirements of P.I.A.

Then he suggested that another schedule should be framed which showed that all the crews have been based at Karachi - home base in the Western Sector. As stated above, there may be more than one optimal solution in the given matrix. Therefore, by shifting some optimal values, the following schedules are prepared for each slip station, having the same numerical values.

8.5.1) Crew Assignment at Beirut, Istanbul, Cairo and London.

Beirut PK 701,703,707,713,702,708,714,718

Crew No.	Crew		Layover Time in Hours
	In	Out	
From Karachi			
301	PK 701	PK 703	24.50
302	PK 703	PK 707	21.58
304	PK 707	PK 713	48.50
306	PK 713	PK 701	70.92
From London			
307	PK 702	PK 708	47.50
301	PK 708	PK 714	43.58
302	PK 714	PK 718	24.50
304	PK 718	PK 702	49.75

Total Time = 330.83 hrs.

Table 42. Crew's Schedule at Beirut.

8.5.2) Istanbul. PK 705,709,715,721,722,706,712,716

Crew No.	Crew		Layover Time in Hours
	In	Out	
From Karachi			
303	PK 705	PK 709	31.41
305	PK 709	PK 715	39.50
307	PK 715	PK 721	31.66
309	PK 721	PK 705	63.25
From London			
303	PK 722	PK 706	47.16
308	PK 706	PK 712	47.50
309	PK 712	PK 716	23.41
306	PK 716	PK 722	47.41
Total Layover Time =			331.30 hrs

Table 43. Crew's Schedule at Istanbul.

8.5.3) Cairo. PK 717,719,704,710

Crew No.	Crew		Layover Time in Hours
	In	Out	
From Karachi			
101	PK 711	PK 710	13.34
103	PK 719	PK 704	85.33
From London			
306	PK 704	PK 711	33.16
303	PK 710	PK 719	33.17
Total Layover Time =			165.00 hrs.

Table 44. Crew's Schedule at Cairo

8.5.4) London.

Crew No.	Crew		Layover Time in Hrs.
	In	Out	
306	PK 701	PK 708	67.42
301	PK 703	PK 710	62.42
309	PK 705	PK 712	49.58
302	PK 707	PK 720	92.25
303	PK 709	PK 714	38.08
308	PK 711	PK 716	25.33
304	PK 713	PK 720	41.67
305	PK 715	PK 718	21.83
308	PK 717	PK 704	87.58
309	PK 717	PK 706	91.33
301	PK 719	PK 722	25.33
307	PK 721	PK 702	41.70

Total Time = 644.52

Table 45. Crew's Schedule at London.

All the above statements are consolidated in Table 46.

8.5.4) contd.

TABLE 46. OR Schedule with one base - Karach in Western Sector

MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON
701 K-B	703 B-L	L	L	710 L-C	C	719 C-L	722 L-I	I	706 I-K					
	703 K-B	707 B-L	L	L	L	720 L-K								
		705 K-I	709 I-L	L	714 L-B	718 B-K								
		707 K-B	B	713 B-L	L	720 L-K								
			709 K-I	I	715 I-L	718 L-B	B	702 B-K						
				711 K-C	710 C-K									
				713 K-B	B	B	701 B-L	L	L	708 L-B	B	714 B-K		
					715 K-I	721 I-L	L	702 L-B	B	708 B-K				
					717 K-L	L	L	L	704 L-C	C	711 C-L	716 L-I	I	722 I-K
					717 K-L	L	L	L	706 L-I	I	712 I-K			
						719 K-C	C	C	704 C-K					
						721 K-I	I	I	705 I-L	L	712 L-I	716 I-K		

8.5.4) contd.

When this OR Schedule is compared with PIA Schedule, a net saving of 169 hours per week is shown as below:-

PIA Schedule	=	1640.73 hours
OR Schedule	=	1471.65 hours
Difference saved per week	=	169.08 hours

8.6) Karachi-Dacca Sector.

In this Sector the return flying time is less than eight hours. The computer program No.7 is applied to get the optimal solution and this solution also shows where the crew should be based.

8.6.1) Karachi-Dacca Sector.

Days	In	Out	Layover time in hours	Based
Monday	PK 723 (Sun)	PK 724	9.00	Dacca
	PK 730	PK 731	1.00	Karachi
	PK 738	PK 737	1.00	Karachi
Tuesday	PK 723 (Mon)	PK 724	9.00	Dacca
	PK 730	PK 731	1.08	Karachi
	PK 736	PK 735	1.08	Karachi
Wednesday	PK 723 (Tue)	PK 724	9.00	Dacca
	PK 730	PK 731	1.08	Karachi
	PK 738	PK 737	1.00	Karachi
Thursday	PK 704	PK 735-A	12.50	Karachi
	PK 723 (Wed)	PK 706	11.00	Dacca
Friday	PK 724	PK 719(Sat)	34.58	Karachi
	PK 711 (Thu)	PK 730	9.00	Dacca
	PK 736	PK 735	1.08	Karachi
Saturday	PK 736	PK 735-A	12.50	Karachi
	PK 723 (Fri)	PK 712	11.00	Dacca
	PK 731 (Fri)	PK 736	19.25	Dacca
Sunday	PK 735 (Sat)	PK 724	11.75	Dacca
	PK 730	PK 731	1.08	Karachi
	PK 736	PK 735	1.08	Karachi

Total layover time = 158.06 hours per week

Table 47. Crew's Schedule for Karachi-Dacca Sector.

8.6.2) Karachi - Persian Gulf Sector.

	Out	In	Layover time in hours.	Based
Wednesday	PK 745	PK 744	0.83	Karachi
Sunday	PK 745	PK 744	0.83	do

Total layover time = 1.66 hours per week

Table 48. Crew's Schedule for Karachi - Persian Gulf Sector

8.6.2) contd.

KARACHI BASE

Schedule for Karachi-Dacca-Persian Gulf-China

MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON
730/ 731	730/ 731		207/ 206											
738/ 737	736/ 735	738/ 737				745/ 744								
209/ 208			704/ 735			730/ 731								
		745/ 744		724	719									
		730/ 731		736/ 735	736/ 735	736/ 735								
		(730)	750/ 751	D	752/ 753	(723)								
		(730)	750/ 751	D	752/ 753	(723)								

Table No.49 Crew's Schedule for Karachi-Dacca-Persian Gulf-China

8.6.3) Dacca-Lahore Sector.

	Crew		Layover time in hours	Based.
	Out	In		
Monday	PK 726	PK 725	1.08	Dacca
Tuesday	PK 726	PK 725	1.08	do
Wednesday	PK 726	PK 725	1.08	do
Friday	PK 726	PK 725	1.08	do
Sunday	PK 726	PK 725	1.08	do

Total layover time = 5.40 hours per week

Table 50. Crew's Schedule for Dacca-Lahore Sector.

8.6.4) Bangkok Sector.

	Crew		Layover time in hours	Based
	Out	In		
Thursday	PK 706-A	PK 711-A	1.25	Dacca
Saturday	PK 712-A	PK 719-A	1.25	do

Total layover time = 2.50 hours per week

Table 51. Crew's Schedule for Dacca-Bangkok Sector

All the above statements for Dacca based are consolidated into one table as given below.

8.6.4) contd.

DACCA BASE

Table 52. Schedule for Dacca-Karachi-Lahore-Bangkok Sector.

MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON
724		725/ 726		725/ 726		723								
723	724			723	712	725/ 726								
	723	724		731	736									
725/ 726		723	706		712/ 719-A									
	725/ 726			711	730									
			706/ 711		735	724								

8.7) <u>Conclusion</u>.

When all the above mentioned OR Schedules are compared with the PIA Schedule, a net saving of 570.08 hours per week is shown. The OR technique also eliminates the human bias in framing the Schedule. Any addition or cancellation of flights, the OR Schedules can be prepared in a few minutes on a computer.

	Sector	PIA Schedule in hours	OR Schedule in hours	Difference in hours (saved)
1)	Karachi-Dacca Dacca-Lahore	219.23	158.06 5.40	55.77
2)	Persian Gulf	3.33	3.33	-
3)	China	187.33	187.33	-
4)	Dacca-Bangkok	164.90	2.50	162.40
5)	Western Sector	1640.73	1288.82	351.91
	Total layover time	2215.52	1645.44	570.08

<u>Table 53</u> Net saving table.

8.7) contd.

PIA SCHEDULE

MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON
724/ 725 LHR	726/ 723			724/ 725	LHR	726/ 723								
	724/ 725 LHR	726/ 723				724/ 725 LHR	726/ 723							
		724/ 725 LHR	LHR	726/ 723										
730/ 731	730/ 731	730/ 731		730/ 731		730/ 731								
	736/ 735			736/ 735	736/ 735	736/ 735								
738/ 737		738/ 737												
			704/ 711		710/ 719									
209/ 208			207/ 206											
		(730)	750/ 751	D	752/ 753	(723)								
		(730)	750/ 751	D	752/ 753	(723)								

Table 54. PIA Schedule for Eastern Sector.

8.7) contd.

MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON
			706/ 706-A	BNK	719-A/ 735-A									
					712/ 712-A	BNK	BNK	BNK	BNK	711-A/ 735-A				

SECTION III

CHAPTER IX

OR TIME TABLE

9.1) Introduction.

Several limited factors have to be taken into consideration while framing the time table. Among these the important factors are given below:-

1) Legal Constraints:-

The routes are allowed to fly and frequencies of the services are subject to international agreement.

2) Commercial Constraints:-

The passengers - who are the source of revenue - have preference for certain times of arrival and departure.

3) Technical Constraints:-

Certain airports are closed to traffic at night, and other airports have not sufficient technical facilities.

The time table department is faced with requirements which are often contradictory and in many cases several time tables are drafted. Now the point arises as to which will provide the best regularity. If the time table is heavily loaded, the revenue may be high but many cancellations and delays will occur, caused by a lack of aircraft at the main base. Then this time table may be termed as "tight schedule" and it may be harmful to the company's reputation. There are also other factors which affect the time table - these are:-

1) the duration of flight varies
2) Operational delay at any station due to traffic congestion, late arrival etc.,

9.1) contd.

3) technical hitches at the moment of departure.

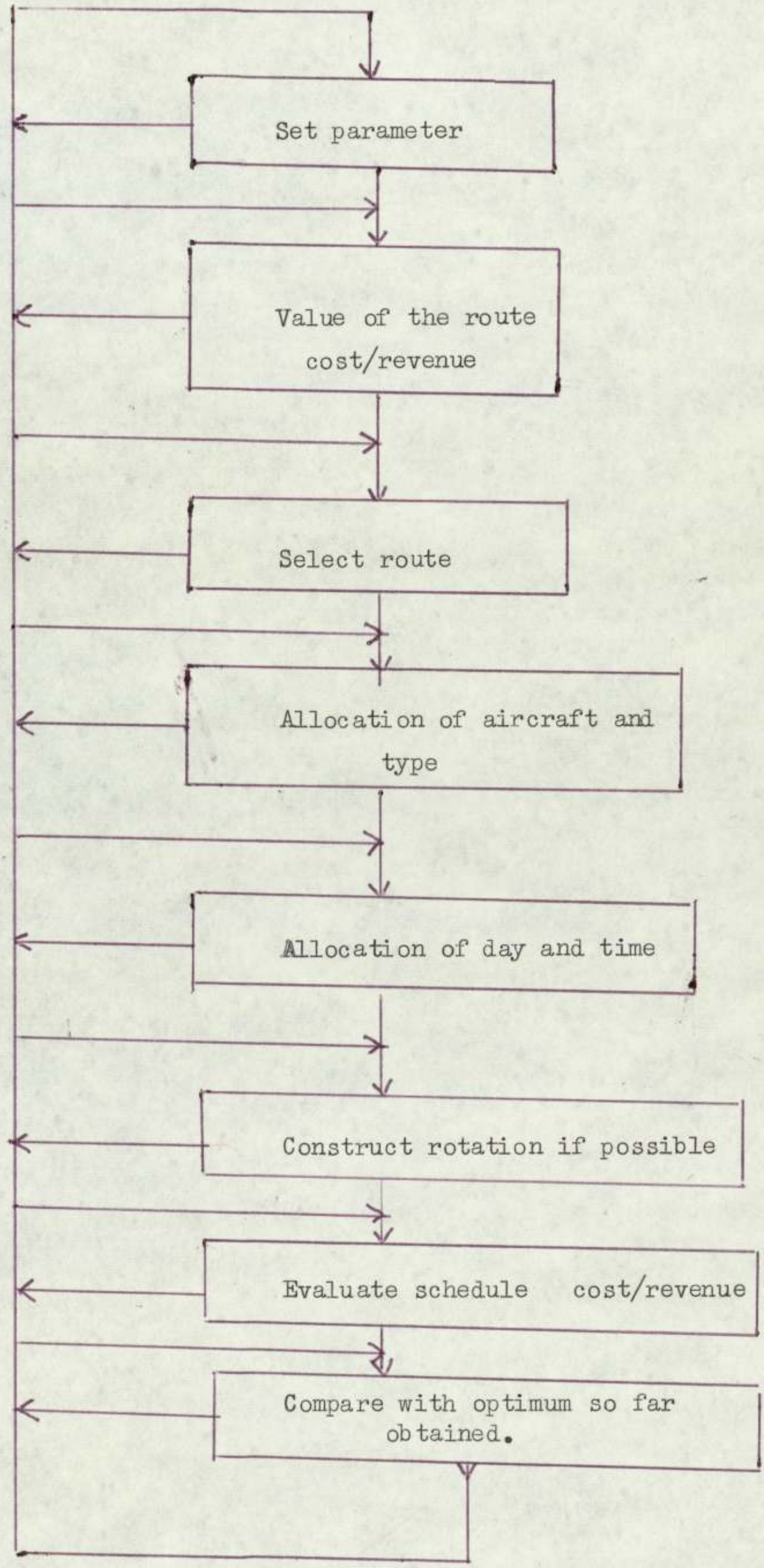
The general aim of all the World Airlines is to find the most profitable solution, that is, the expression revenue minus cost, should be maximized. Here the profit means overall profit. It is the duty of the Marketing Research Section to frame a schedule while considering all of the above mentioned constraints.

Now the next important point is to select the route while considering certain parameters:-

a) cost of landing
b) cost of flight
c) expected revenue of the flight
d) maintenance facilities
e) day time and seasonal conditions
f) connection for other flights
g) attraction for the visitors (publicity)

The overall profit depends upon certain parameters. For optimal time table some kind of iterative procedure should be adopted, running through each step again and again while changing the parameters so as to increase the profit at every cycle. The parameters can be framed as follows:-

9.1) contd.


Set parameter
Value of the route
cost/revenue
Select route
Allocation of aircraft and
type
Allocation of day and time
Construct rotation if possible
Evaluate schedule cost/revenue
Compare with optimum so far
obtained.

9.1) contd.

Four months ahead a time table is framed by the Market Research Station while considering the above mentioned parameters. In order to get the ideal time table, some relaxation should be provided to deviate the time table, for reducing the layover time of the crews at slip stations and provide a time for maintenance of the aircraft etc., Then this time table should be analysed by the OR Section and Technical Section. In the present case, only the layover time of the crews will be considered.

If the time table is drafted while considering the layover time of the crews, this might become unattractive from a commercial point of view.

PIA Authority has allowed deviation on the present time table of up to half an hour only.

It is suggested that a change in the departure time of flight Nos. PK 724, 706, 712 in Karachi-Dacca Sector should be made by 25 minutes only. The proposed time table for this Sector is as follows.

TIME FROM GMT		MONDAY			TUESDAY			WEDNESDAY			THURSDAY		FRIDAY			SATURDAY			SUNDAY		
		PK 724	PK 730	PK 738	PK 724	PK 730	PK 735	PK 724	PK 730	PK 738	PK 704	PK 706	PK 724	PK 730	PK 736	PK 710	PK 712	PK 736	PK 724	PK 730	PK 736
+5	KARACHI dep	6-05	8-30	18-00	6-05	8-30	11-15	6-05	8-30	18-00	3-30	8-05	6-05	8-30	11-15	3-30	8-05	11-15	8-05	8-30	11-15
+6	DACCA arr	10-15	12-40	22-10	10-15	12-40	15-25	10-15	12-40	22-10	7-40	12-15	10-15	12-40	15-25	7-40	12-15	15-25	10-15	12-40	15-25

TIME FROM GMT		MONDAY			TUESDAY			WEDNESDAY			THURSDAY		FRIDAY			SATURDAY			SUNDAY		
		PK 731	PK 723	PK 737	PK 731	PK 737	PK 723	PK 731	PK 723	PK 737	PK 735A	PK 711	PK 731	PK 735	PK 723	PK 735	PK 735A	PK 719	PK 731	PK 735	PK 723
+6	DACCA dep	13-45	19-15	23-10	13-45	16-30	19-25	13-45	19-15	23-10	20-10	21-15	13-45	16-30	19-15	16-30	20-10	21-15	13-45	16-30	19-15
+5	KARACHI arr	16-00	20-30	TUE 1-25	16-00	18-45	21-30	16-00	21-30	THU 1-25	22-25	23-30	16-00	18-45	21-30	18-45	22-25	23-30	16-00	18-45	21-30

TABLE 55 :-PROPOSED TIME TABLE FOR KARACHI - DACCA SECTOR

9.1) contd.

The layover time matrices for Karachi and Dacca are calculated and modified as explained in (3.5). Then a composite matrix is obtained from these two matrices. The computer program No.7 is applied to get the optimal solution. This optimal solution has dual advantages of giving minimum layover time of the crews and the element selected in the solution shows the base for the crews.

9.2 :-LAYOVER TIME MATRIX FOR KARACHI - OR TIME TABLE

		MONDAY			TUESDAY			WEDNESDAY			THURSDAY		FRIDAY			SATURDAY			SUNDAY		
		a_{731}	a_{723}	a_{737}	a_{731}	a_{735}	a_{723}	a_{731}	a_{723}	a_{737}	a_{735A}	a_{711}	a_{731}	a_{735}	a_{723}	a_{731}	a_{735A}	a_{719}	a_{731}	a_{735}	a_{723}
M	d724	158.08	153.58	148.58	134.08	131.33	128.58	110.08	104.58	100.66	79.66	78.58	62.08	59.33	56.58	35.33	31.66	30.59	14.08	11.33	176.58
O	d_{730}	160.50	156.00	151.00	136.50	133.75	131.00	112.50	107.00	103.08	82.08	81.00	64.50	61.75	59.00	37.75	34.08	33.00	16.50	13.75	11.00
N	d_{738}	2.00	165.50	160.50	146.00	143.25	140.50	122.00	116.50	112.58	91.58	90.50	74.00	71.25	68.50	47.25	43.58	42.50	26.00	23.25	20.50
T	d_{724}	14.08	9.58	172.58	158.08	155.33	152.58	134.08	128.58	124.66	103.66	102.58	86.08	83.33	80.5 8	59.33	55.66	54.58	38.00	35.33	32.56
U	d_{730}	16.50	12.00	175.00	161.50	157.75	155.00	136.50	131.00	127.08	106.08	105.50	88.00	85.75	83.00	61.75	59.08	57.00	40.50	37.75	35.00
E	d_{736}	19.25	14.75	9.75	163.25	160.50	157.75	139.25	133.75	129.83	108.83	107.75	91.25	88.50	85.75	64.50	60.83	59.75	43.25	40.50	37.75
W	d_{724}	38.08	33.58	28.58	14.08	11.33	9.00	158.08	152.58	148.66	127.66	126.58	110.08	107.33	104.58	83.33	79.66	78,58	82.08	59.33	56.58
E	d_{730}	40.50	36.00	31.00	16.50	13.75	11.00	160.50	155.00	151.08	130.08	129.00	112.50	109.75	107.00	85.75	82.08	81.00	64.50	61.75	59.00
D	d_{738}	50.00	45.50	40.50	26.00	23.25	20.50	2.00	164.50	164.50	139.58	138.50	122.00	119.25	116.50	95.25	91.58	90.50	74.00	71.25	68.50
THU	d_{704}	59.50	55.00	50.00	35.35	32.75	30.00	11.50	174.00	2.08	149.08	148.00	131.50	128.75	126.00	104.75	101.08	100.00	83.50	80/75	78.00
	d_{706}	64.08	59.58	54.56	40.08	37.33	34.58	16.08	10.58	174.66	153.66	152.58	136.58	133.33	130.58	109.33	105.66	104.58	88.08	85.33	82.58
F	d_{724}	86.08	81.58	76.58	62.08	59.33	56.58	38.08	32.58	28.66	175.66	174.58	158.08	155.33	152.58	131.33	127.66	126.58	110.08	107.33	104.58
R	d_{730}	88.50	84.00	79.00	64.50	61.75	59.00	40.50	35.00	31.08	10.08	9.00	160.50	157.75	155.00	133.75	130.08	129.00	112.50	109.75	107.00
I	d_{736}	91.25	86.75	81.75	67.25	64.50	61.75	43.25	37.75	33.83	12.83	11.75	163.25	160.70	157.75	136.50	132.83	131.75	115.25	112.50	109.75
S	d_{710}	107.50	103.00	98.00	83.50	80.75	78.00	59.50	74.00	50.08	29.08	28.00	11.50	176.75	174.00	152.75	149.08	148.00	131.50	128.75	126.00
A	d_{712}	112.08	107.58	102.58	88.08	85.33	82.58	64.08	58.58	54.66	33.66	32.58	16.08	13.33	10.58	157.33	153.66	152.58	136.08	133.33	130.58
T	d_{730}	115.25	110.75	105.75	91.25	88.50	85.75	67.25	61.75	57.83	36.83	35.75	9.25	16.50	13.75	160.50	156.83	155.75	139.25	136.50	133.75
S	d_{724}	134.08	129.58	124.58	110.08	107.33	104.58	86.08	80.58	76.66	55.66	54.58	38.08	35.33	32.58	11.33	175.66	174.58	158.08	155.33	152.58
U	d_{730}	136.08	132.00	127.00	112.50	109.75	107.00	88.50	83.00	79.08	58.08	57.00	40.50	37.75	35.00	13.75	10.08	9.00	160.50	157.75	155.00
N	d_{736}	139.25	134.75	129.75	115.25	112.50	109.75	91.25	85.75	81.83	60.83	59.75	43.25	40.50	37.75	16.50	12.83	11.75	163.25	160.50	171.500

TABLE 56:- CREW BASED AT DACCA

9.3. LAYOVER TIME MATRIX AT DACCA OR TIME TABLE

	MONDAY			TUESDAY			WEDNESDAY			THURSDAY		FRIDAY			SATURDAY			SUNDAY		
	d_{731}	d_{723}	d_{737}	d_{731}	d_{735}	d_{723}	d731	d_{723}	d_{737}	d_{735-A}	d_{711}	d_{731}	d_{735}	d_{723}	d_{735}	d_{735A}	d_{719}	d_{731}	d_{735}	d_{72}
a_{724}	171.50	9.00	12.92	27.50	30.50	33.00	51.50	57.00	60.92	81.92	83.00	99.50	102.25	105.00	126.25	129.92	131.00	147.50	135.25	153.0
a_{730}	1.08	174.58	10.50	25.08	27.83	30.58	49.08	54.58	58.50	79.50	80.58	97.08	99.83	102.58	123.83	127.50	128.58	145.08	147.83	150.5
a_{738}	159.58	165.08	1.00	15.58	18.33	21.08	39.58	45.08	49.00	70.33	71.08	87.58	90.33	93.08	114.33	118.00	119.08	135.58	138.33	141.0
a_{724}	147.50	153.00	156.92	171.50	174.25	9.00	27.50	33.00	36.92	57.92	59.00	75.50	78.25	81.00	102.25	105.92	107.00	123.50	126.25	129.0
a_{730}	145.08	150.58	154.50	1 .08	171.83	174.58	25.08	30.58	34.50	55.50	56.58	73.08	75.83	78.58	99.83	103.50	104.58	121.08	123.83	126.5
a_{736}	142.33	147.83	151.75	166.33	1.08	171.83	22.33	27.83	31.75	52.75	53.83	70.33	73.08	75.83	97.08	100.75	101.83	118.33	121.08	123.8
a_{724}	123.50	129.00	132.92	147.50	150.25	153.00	171.50	9.00	12.92	33.92	35.00	51.50	54.25	57.00	78.25	81.92	83.00	99.50	102.25	105.0
a_{730}	121.08	126.58	130.50	145.08	147.83	150.58	1.08	174.58	10.50	31.50	32.50	49.08	51.83	54.58	75.83	79.00	80.58	97.08	99.83	102.5
a_{738}	111.58	117.08	121.00	135.58	138.33	141.08	159.58	165.08	1.08	22.00	23.08	39.58	42.33	45.08	66.33	70.00	71.08	87.58	90.33	93.0
a_{704}	102.08	107.58	111.50	126.08	184.83	131.58	150.08	155.58	159.50	12.50	13.58	30.08	32.83	35.58	56.83	60.50	51.58	78.08	80.83	83.5
a_{706}	97.50	103.00	106.92	121.50	124.25	127.00	145.50	151.00	154.92	175.92	9.00	25.50	28.25	31.00	52.25	55.92	57.08	73.50	76.25	78.0
a_{724}	75.50	81.00	84.92	99.50	102.25	105.00	123.50	129.00	132.92	153.92	155.00	171.50	174.25	9.00	30.25	33.92	35.00	51.50	54.25	57.0
a_{730}	73.08	78.58	82.50	97.08	99.83	102.58	121.08	126.58	130.50	151.50	152.58	1.08	171.83	174.58	27.83	31.50	32.58	49.08	51.83	54.5
a_{736}	70.33	75.83	79.75	94.33	97.08	99.83	118.33	123.83	127.75	148.75	149.83	166.33	1.08	171.83	25.08	28.75	29.83	46.33	49.08	51.8
a_{710}	54.08	59.58	63.50	78.08	80.83	83.58	102.08	107.58	111.50	132.50	133.58	150.08	152.83	155.58	9.00	12.50	13.58	30.08	32.83	35.5
a_{712}	49.50	55.00	58.92	73.50	76.25	79.00	97.50	103.00	106.92	127.92	129.00	145.50	148.25	151.00	172.25	175.92	9.00	25.50	28.25	31.0
a_{730}	46.33	51.83	55.75	70.33	73.08	75.83	94.33	99.83	103.75	124.75	125.83	142.33	145.08	147.83	1.08	172.75	173.83	22.33	25.08	27.8
a_{724}	27.50	33.00	36.92	51.50	54.25	57.00	75.50	81.00	84.92	105.92	107.00	123.50	126.24	129.00	150.25	153.92	155.00	171.50	174.25	9.00
a_{730}	25.08	30.58	34.50	49.08	51.83	54.58	73.08	78.58	82.50	103.50	104.58	121.08	123.83	126.58	147.83	151.50	152.58	1.08	171.83	174.58
a_{736}	22.33	27.83	31.75	46.33	49.08	51.83	70.33	75.83	79.75	100.75	101.83	118.33	121.08	123.83	145.08	148.75	149.83	166.33	1.08	171.58

TABLE 57: CREW BASED AT KARACHI

9.4 : COMPOSITE LAYOVER TIME MATRIX FOR KARACHI - DACCA SECTOR OR TIME TABLE

		MONDAY			TUESDAY			WEDNESDAY		
		PK 731	PK 723	PK 737	PK 731	PK 735	PK 723	PK 731	PK 723	PK 737
MON	PK 724	158.08 K	9.00 K	12.50 K	27.08 K	29.83 K	32.58 K	51.08 K	56.58 K	60.50 K
	PK 730	1.08 K	156.00 D	10.50 K	25.08 K	27.83 K	30.58 K	49.08 K	54.08 K	58.50 K
	PK 738	2.00 D	105.08 K	1.00 K	15.58 K	18.33 K	21.08 K	39.58 K	45.08 K	49.00 K
TUE	PK 724	14.08 D	9.58 D	156.50 K	158.50 D	155.75 D	9.00 K	27.08 K	32.58 K	36.50 K
	PK 730	16.50 D	12.00 D	154.50 K	1.08 K	157.75 D	155.00 D	25.08 K	30.58 K	34.50 K
	PK 736	19.25 D	14.75 D	9.75 D	163.25 D	1.08 K	157.75 D	22.33 K	27.83 K	31.75 K
WED	PK 724	38.08 D	33.58 D	28.58 D	14.08 D	11.33 D	9.00 D	158.08 D	9.00 K	12.50 K
	PK 730	40.50 D	36.00 D	31.08 D	16.50 D	13.75 D	11.00 D	1.08 K	155.00 D	10.50 K
	PK 738	50.00 D	45.50 D	40.50 D	26.00 D	23.25 D	20.50 D	2.00 D	164.50 D	1.00 K
THU	PK 704	59.50 D	55.00 D	50.00 D	35.50 D	32.75 D	30.00 D	11.50 D	155.58 K	2.08 D
	PK 706	64.08 D	59.58 D	54.00 D	40.08 D	37.33 D	34.58 D	16.08 D	10.58 D	154.50 K
FRI	PK 724	75.08 K	80.58 K	76.58 D	62.08 D	59.33 D	56.58 D	38.00 D	32.58 D	28.66 D
	PK 730	73.08 K	78.58 K	79.00 D	64.50 D	61.75 D	59.00 D	40.50 D	35.00 D	31.08 D
	PK 736	70.33 K	75.83 K	79.75 K	67.25 D	64.50 D	61.75 D	43.25 D	37.75 D	33.83 D
SAT	PK 710	54.08 K	59.58 K	63.50 K	78.08 K	80.75 D	78.00 D	59.58 D	54.00 D	50.08 D
	PK 712	49.08 K	54.58 K	58.50 K	73.08 K	75.83 K	78.58 K	64.08 D	58.58 D	54.66 D
	PK 736	46.33 K	51.83 K	55.75 K	70.33 K	73.08 K	75.83 K	67.25 D	61.75 D	57.83 D
SUN	PK 724	27.08 K	32.58 K	36.50 K	51.08 K	53.83 K	56.58 K	75.08 K	80.58 K	76.66 D
	PK 730	25.08 K	30.58 K	34.50 K	49.08 K	51.83 K	54.58 K	73.08 K	78.58 K	79.08 D
	PK 736	22.33 K	27.83 K	31.75 K	46.33 K	49.08 K	51.83 K	70.33 K	75.83 K	79.75 K

		THURSDAY		FRIDAY			SATURDAY			DUNDAY		
		PK 735A	PK 711	PK 731	PK 735	PK 723	PK 735	PK 735A	PK 719	PK 731	PK 735	PK 72
MON	PK 724	79.66 D	78.58 D	62.08 D	59.33 D	56.58 D	35.33 D	31.66 D	30.58 D	14.08 D	11.33 D	152.58 K
	PK 730	79.50 K	80.58 K	64.50 D	61.75 D	59.00 D	37.75 D	34.08 D	33.00 D	16.50 D	13.75 D	11.00 D
	PK 738	70.00 K	71.08 K	74.00 D	71.25 D	68.50 D	47.25 D	43.58 D	42.50 D	26.00 D	23.25 D	20.25 D
TUE	PK 724	57.50 K	58.58 K	75.08 K	77.83 K	80.58 K	59.33 D	55.66 D	54.58 D	38.08 D	35.33 D	32.58 D
	PK 730	55.50 K	56.58 K	73.08 K	75.83 K	80.58 K	61.75 K	58.08 K	57.00 K	40.50 K	37.75 K	35.00 K
	PK 736	52.75 K	53.83 K	70.33 K	73.08 K	75.83 K	64.50 D	60.83 D	59.75 D	43.25 D	40.50 D	37.75 D
WED	PK 724	33.50 K	34.58 K	51.08 K	53.83 K	56.58 K	77.83 K	81.50 D	78.58 D	62.08 D	59.33 D	56.58 D
	PK 730	31.50 K	32.58 K	49.08 K	51.83 K	54.58 K	75.83 K	79.50 K	81.00 K	64.50 D	61.75 D	59.00 D
	PK 738	22.00 K	23.08 K	39.58 K	42.33 K	45.08 K	66.33 K	70.00 K	71.08 K	74.00 D	71.25 D	68.50 D
THU	PK 704	12.50 K	13.58 K	30.08 K	32.83 K	35.58 K	56.83 K	60.50 K	61.58 K	78.08 K	80.83 K	78.00 K
	PK 706	154.08 D	9.00 K	25.08 K	27.83 K	30.58 K	51.83 K	55.50 K	56.58 K	73.08 K	75.83 K	58.75 K
FRI	PK 724	153.50 K	154.58 K	158.08 D	155.33 D	9.00 K	29.83 K	35.50 K	34.58 K	51.00 K	53.83 K	56.75 K
	PK 730	10.08 D	9.00 D	1.08 K	157.75 D	155.00 D	27.83 K	31.50 K	32.58 K	49.08 K	51.83 K	54.58 K
	PK 736	12.83 D	11.75 D	152.58 D	1.08 K	157.75 D	25.08 K	28.75 K	29.83 K	46.33 K	49.08 K	51.83 K
SAT	PK 710	29.08 D	28.00 D	11.50 D	152.83 D	155.58 K	152.75 D	12.50 K	13.58 K	30.08 K	32.83 K	35.75 K
	PK 712	33.66 D	32.58 D	16.08 D	13.33 D	10.58 D	157.75 D	153.66 D	9.00 K	25.08 K	27.83 K	30.58 K
	PK 736	36.83 D	35.75 D	19.25 D	16.50 D	13.75 D	1.08 K	156.83 D	155.75 D	22.33 K	25.08 K	28.00 K
SUN	PK 724	55.66 D	54.58 D	38.08 D	35.33 D	32.58 D	11.33 D	153.50 K	154.58 K	158.08 D	155.33 D	9.00 K
	PK 730	58.08 D	57.00 D	40.50 D	37.75 D	35.00 D	13.75 D	10.08 D	9.00 D	1.08 K	157.75 D	155.00 D
	PK 736	60.83 D	59.75 D	43.25 D	40.50 D	37.75 D	16.50 D	12.83 D	11.75 D	163.25 D	1.08 K	157.75 D

TABLE 58: COMPOSITE TIMEMATRIX

10.5 Optimal Flight Combination of
Karachi-Dacca Sector

	Crews out	Crews in	Layover time in hours	Crew based at
MONDAY	PK 724	PK 723	9.00	Karachi
	PK 730	PK 731	1.08	do
	PK 738	PK 737	1.00	do
TUESDAY	PK 724	PK 723	9.00	do
	PK 730	PK 731	1.08	do
	PK 736	PK 735	1.08	do
WEDNESDAY	PK 724	PK 723	9.00	do
	PK 730	PK 731	1.08	do
	PK 738	PK 737	1.00	do
THURSDAY	PK 704	PK 735-A	12.50	do
	PK 706	PK 711	9.00	do
FRIDAY	PK 724	PK 723	9.00	do
	PK 730	PK 731	1.08	do
	PK 736	PK 735	1.08	do
SATURDAY	PK 710	PK 735-A	12.50	do
	PK 712	PK 719	9.00	do
	PK 736	PK 735	1.08	do
SUNDAY	PK 724	PK 723	9.00	do
	PK 736	PK 735	1.08	do
	PK 730	PK 731	1.08	do

Total layover time = 99.72 hours

Table 59 Layover time of Karachi/Dacca Sector.

9.3) <u>SCHEDULE FOR EASTERN SECTOR KARACHI/DACCA/CHINA</u>

Base = Karachi

MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON
724/ 723	724/ 723	724/ 723			710/ 735-A									
730/ 731	730/ 731	730/ 731			712/ 719									
738/ 737	736/ 735	738/ 737			736/ 735									
			704/ 735-A	724/ 723		724/ 723								
			706/ 711	730/ 731		730/ 731								
				736/ 735		730/ 735								
			(730)	750/ 751	D	752/ 753	(723)							
			(730)	750/ 751	D	752/ 753	(723)							

The layover time for this Schedule is 99.72 hours and PIA Schedule shows 219.23 hours. This means a net saving of 119.51 hours per week shown in Karachi-Dacca Sector only. There is no need to change the time in the remaining sectors because flight time has to be changed from more than half an hour which is not allowed by the Airline Authority.

CHAPTER X.

MONTHLY ASSIGNMENT.

10.1) Introduction.

This Chapter describes the progress made in developing a mathematical technique for allocation of monthly duties to the air crews. This technique will eliminate the human bias or favour in assigning the crews to various flights. Before explanation of mathematical technique it is necessary to explain some important factors. It is the policy of PIA that every crew should go on each route, however, some routes are popular as compared to other routes.

10.2) Type of Duties to be Allocated.

A duty, as the term is used in this Chapter, consists of a day of work or succession of days of work. It may start and finish at any time in 24 hours and may be any number of days in length. It is usual to describe the duty of two or more than 2 days as tour. Another term also used in this Chapter is "flying duty time", this is equal to actual flying time plus one hour before scheduled departure plus half an hour after scheduled arrival at the terminating point. This flying duty time will determine the minimum rest period that must follow before the next duty.

10.3) Essential Flying Rules.

In preparing the monthly roaster, it is necessary to describe the essential rules such as

1) Limitation of maximum flying time
   a) 30 hours in one calendar week
   b) 70 hours in one calendar month
   c) 700 hours in one calendar year

10.3) contd.

2) Minimum of ten days in each calendar month free from all duties at the home base should be provided in the monthly roaster. Ten days should be distributed in such a way that only once every 24 hours are allowed in a month and all other remaining period should be at least 48 hours duration.

3) A crew cannot fly more than eight hours in any 24 consecutive hours, unless he is given an intervening rest period at or before the termination of eight scheduled hours of duty aloft. Such rest period will twice the number of hours on duty since the last rest period, and in no case will the rest period be less than eight hours.

10.4) "Flying time between two terminal cities".

Suppose there are two citues A and B and actual flying time between them is as follows

i) $\vec{AB} + \vec{BA} \leq$ 8 hours

ii) $\vec{AB} + \vec{BA} >$ 8 hours

iii) $\vec{AB}$ OR $\vec{BA} \geq$ 8 hours

iv) $\vec{AB}$ OR $\vec{BA} >$ 8 hours

10.5) Size of the problem.

To give some idea of the size of the problem that has to be handled, the number of duties covered in one year actual flying time in a year. The actual flying time of the year will determine the number of pilots to operate a schedule which is based on the given time table.

10.5) contd.

It is suggested that there should be two bases Karachi and Dacca. The number of duties and actual flying time in a year are given below:

10.5.1) Karachi base.

i) Western Sector.

Actual flying time during the week

MONDAY	=	1365 mts.
TUESDAY	=	1410 "
WEDNESDAY	=	2615 "
THURSDAY	=	1450 "
FRIDAY	=	2400 "
SATURDAY	=	3500 "
SUNDAY	=	3380 "
Total for one week	=	16120 "
Total for one year	=	16120 × 52 = 838240 mts.
Total number of flying duties in a year	=	2080

10.5.2) Eastern Sector.

MONDAY	=	1155 mts.
TUESDAY	=	1155 "
WEDNESDAY	=	1155 "
THURSDAY	=	2020 "
FRIDAY	=	1155 "
SATURDAY	=	2395 "
SUNDAY	=	1155 "

10.5) contd.

10.5.2) contd.

Total flying time for one week = 10190 mts.

" " " " " year = 10190×52 = 529880 mts

Total no. of flying duty in a year = 1248.

10.5.3) Middle East Sector.

Actual flying time during the week.

MONDAY	=	510 mts.
TUESDAY	=	-
WEDNESDAY	=	510 "
THURSDAY	=	510 "
FRIDAY	=	310 "
SATURDAY	=	285 "
SUNDAY	=	840 "

Total flying time in a week = 2965 mts.

" " " " " year 2965 ×52 = 154180 mts.

Total no. of flying duties in a year = 416

Grand total flying time in a year = 1522300 mts.

" " " duty " " " = 3744 "

3744 duties require 1522300 mts. in a year. There are seven pilots who are assigned ground duties in addition to their flying duty. Due to this extra work their yearly flying time is reduced to 400 hours and they are not entitled to have 10 days free from all the duties of the airline. Therefore, their monthly flying time is reduced to 40 hours at Karachi, i.e. 10% of annual flying time. In order to determine the number of pilots at Karachi base, their yearly

10.5) contd.

10.5.3) contd.

flying time (7×400×60 = 168000 mts) should be subtracted from the grand total. So the grand total is

1522300 - 168000 = 1354300 mts.

$\therefore$ No. of pilots $= \frac{1358200}{42000} = 32.2452$

$= 33$

The total number of pilots who are to be posted at Karachi base = 33 + 7 = 40.

10.5.4) Dacca base.

Actual flying time during the week on

MONDAY	=	305 mts.
TUESDAY	=	305 "
WEDNESDAY	=	305 "
THURSDAY	=	270 "
FRIDAY	=	305 "
SATURDAY	=	270 "
SUNDAY	=	305 "
Total for week	=	2065 "

$\therefore$ Total flying time for year = 2065×52

= 107380 mts.

Total no. of duty in a year= 364

No. of pilots who are to be based at Dacca $= \frac{107380}{42000} = 2.5566$

$= 3$

10.6) Monthly Schedule.

There are 11, 24 & 4 weekly flights to the Western,

10.6) contd.

Eastern and Middle East Sector respectively. In the Western Sector 12 crews are used to operate 11 flights. The reason is explained in Chapter III. The crews of two flights PK 711 and PK 719 come back to Karachi in flight Nos. PK 710 and PK 704 after taking the due rest at Cairo. Therefore, these two crews can be included in the Middle East Sector raising the total number of flights to 6. Only 40 crews are required to operate the schedule from the Karachi Sector. They are distributed into 4 equal groups and allocation is made as shown in the following pattern.

	I Group	II Group	III Group	IV Group
1st week	Western	Eastern	Middle East	-
2nd week	-	Western	Eastern	Middle East
3rd week	Middle East	-	Western	Eastern
4th week	Eastern	Middle East	-	Western

The OR Schedules for the Western Sector is not finished in one week. Therefore, the week following to this Sector is left unassigned so that this Sector may be completed in next week.

10.7) Mathematical Model.

Sometimes it happens that a pilot does not want to go on certain flights because he may have some important work to do at the home base or some routes are very popular and every one desires to go on that route. For this purpose a bid preference card is prepared, which is to be filled in by every pilot. The card may be as given below:-

10.7) contd.

Base Karachi

SECTOR WESTERN/EASTERN/MIDDLE EAST

MONTH YEAR

NAME OF PILOT_____________________ CODE NO_________

TOTAL PREVIOUS MONTHS FLYING HOURS________

DATE	1	2	3	4	-	-	-	-	31	
DAY										
FLIGHT NO.										

First of all OR Schedule will be prepared and distributed among all the pilots. According to their discretion, all the information would be transferred on to a separate paper sectorwise and weekwise giving a matrix of OS and 1S in which row represents the pilot and column represents the flight number. '1' occupies a cell when a pilot wants to go and '0' when he does not want to go on certain flights.

Now the set theory will be applied to the matrix so obtained for allocation of the duties to the pilots, which is entirely based on a mathematical technique. This method will eliminate the personal bias in assigning the duties to the pilots.

Select a row having a maxim number of 1S as a set. If there are more than one such row, take the first one. Compare this set row with the remaining rows (known as sub-set), if any sub-set row or a set of sub-set rows have zero element and set rows have non-zero-element in the respective column can

10.7) contd.

be eliminated. A sub-set row or a set of sub-set rows can also be eliminated.

In set theory terminology, consider S_n as set and S_m as sub-set, such that

$$S_m \subset S_n$$

This means if a row S_m have some non-zero elements and at least all these non-zero elements must be in S_n. A column in which S_n have non-zero elements and a set of columns in which S_m have zero-elements cannot occur in the feasible solution.

The above statement can be written as

$$\bar{S}_m \cap S_n$$

and cannot occur in the feasible solution.

In this way the original matrix would be reduced, and this process is repeated. At a certain stage, if any row has only one non-zero element (x_{ij}) this element must be included in the solution. Therefore the i^{th} row and j^{th} column can be eliminated from the reduced matrix, because the total of all rows and columns is unity.

If at a certain stage there is a tie, calculate the flying time of each flight and assign the crew on the basis of previous months total e.g. the greater flying time of the flight to that crew having less previous months total.

This process is repeated until no further sub-matrix is obtained.

10.8) The above technique is explained with an example in which row represents the pilot and column represents the flight number. A matrix of 1 and 0 is framed while considering the bid preference cards. Now set theory technique is applied to get the unbiased allocation of pilot duties.

FLIGHT NO. PK.

Pilot No.	701	703	705	707	709	713	715	717-A	717-B	721
1	1	0	0	1	0	0	1	1	0	1
2	1	1	1	1	0	0	1	0	1	1
3	0	0	0	0	1	1	1	0	0	0
4	1	0	0	0	1	1	1	0	0	0
5	1	0	0	1	1	1	0	0	0	0
6	1	1	0	1	1	0	0	0	1	0
7	1	1	0	1	0	0	0	1	1	0
8	0	0	1	1	0	0	0	1	1	1
9	0	1	1	1	0	1	0	1	1	1
10	0	1	1	1	0	1	0	1	1	1

Table 61. Bid preference

After some iterations, the following result is obtained.

Pilot No.		Flight No.PK
1	should go on	721
2	" " "	703
3	" " "	715
4	" " "	701
5	" " "	707
6	" " "	717-B
7	" " "	717-A
8	" " "	705
9	" " "	713
10	" " "	709

As there are 40 pilots at Karachi base, the bid preference

10.7) contd.

cards are filled weekwise by the pilots for Western, Eastern and Middle East Sector. Then "set theory" technique is applied for the allocation of monthly duties as explained above.

The monthly OR Schedule is given in the following table for the Karachi base.

10.8) Karachi Base

NO.	NAME OR CODE NO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
		MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE
1	04	PK 701 K - B 325	PK 703 B - L 370	L	L	PK 710 L - C 415	C	PK 719 C - L 440	PK 722 L - I 295	I	PK 706 I - K 455						
2	02		PK 703 K - B 370	PK 707 B - L 375	L	L	L	PK 720 L - K 710									
3	08			PK 705 K - I 400	PK 709 I - L 305	L	PK 714 L - B 355	PK 718 B - K 305								PK 209/208 510	
4	05			PK 707 K - B 270	B	PK 713 B - L 450	L	PK 720 L - K 710									
5	10				PK 709 K - I 475	I	PK 715 I - L 315	PK 718 L - B 425	B	PK 702 B - K 325							
6	09					PK 713 K - B 345	B	B	PK 701 B - L 370	L	L	PK 708 L - B 355	B	PK 714 B - K 270			
7	03						PK 715 K - I 400	PK 721 I - L 300	L	PK 702 L - B 345		PK 708 B - K 315					
8	07						PK 717 K - L 745	L	L	L	PK 704 L - C 415	C	PK 711 C - L 440	PK 716 L - I 295	I	PK 722 I - K 375	
9	06						PL 717 K - L 745	L	L	L	PK 706 L - I 385	I	PK 712 I - K 455				
10	01							PK 721 K - I 490	I	I	PK 705 I - L 315	L	PK 712 L - I 295	PK 716 I - K 375			

NO.	NAME OR CODE NO	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	TOTAL
		WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	TIME IN MTS.
1	04						PK 724/723 385	PK 724/723 385	PK 724/723 385			PK 710/735A 385				PK 705 K-I 400	3840
2	02									PK 704/735A 385	PK 724/723 385		PK 724/723 385				2610
3	08		PK 207/206 510											PK 701 K - B 325	PK 703 B - L 370	L	3085
4	05						PK 730/731 385	PK 730/731 385	PK 730/731 385			PK 712/719 385					2970
5	10			PK 711 K - C 310	PK 710 C - K 285					PK 706/711 385	PK 730/731 385		PK 730/731 385				3290
6	09					PK 719 K - C 310	C	C	PK 704 C - K 285					PK 703 K - B 370	PK 707 B - L 375	L	2650
7	03	PK 745/744 510				PK 745/744 460					PK 736/735 385		PK 736/735 385				3100
8	07								(730)	PK 750/751 625	D	PK 752/753 625	(723)				3515
9	06								(730)	PK 750/751 625	D	PK 752/753 620	(723)				2830
10	01						PK 738/737 35	PK 736/735 385	PK 738/737 385			PK 736/735 385				PK 707 K - B 270	3285

Table 61:- Monthly OR schedule for Karachi base

Table 61 contd.

NO.	NAME OR CODE NO.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	TOTAL TIME IN MTS.
		MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	
11	27								PK 701 K-B 325	PK 703 B- L 370	L		PK 710 L - C 415		PK 719 C - L 440	PK 722 L - I 295	I	PK 704 I - K 455							PK 745/744				PK 745/744				3220
12	11	PK 738/737 385	PK 736/735 385	PK 736/737 385			PK 736/735 385			PK 703 K - B 370	PK 707 B - L 375	L	L	L	PK L - K 710												PK 711 K - C 310	PK 710 C - K 285					3590
13	17				PK 706/711 385	PK 730/731 385		PK 730/731 385			PK 705 K - I 400	PK 709 I - L 305	L	PK 714 L - B 355	PK 713 B - K 305								PK 209/208 510			PK 207/206 510							3540
14	15	PK 730/731 385	PK 730/731 385	PK 730/731 385			PK 712/719 385				PK 702 K - B 270	B	PK 713 B - L 450	L	PK 720 L - K 710															PK 738/737 385	PK 738/737 385	PK 738/737 385	4125
15	12				PK 704/735A 385	PK 724/723 385		PK 724/723 385				PK 709 K - I 475	I	PK 715 I - L 315	PK 713 L - B 425	B	PK 702 B - K 325													PK 724/723 385	PK 724/723 385	PK 724/723 385	3850
16	18	PK 724/723 385	PK 724/723 385	PK 724/723 385			PK 710/735A 385						PK 713 K - B 345	B	B	PK 701 B - L 370	L	L	PK 708 L-B 355	B	PK 714 B - K 270									PK 730/731 385	PK 730/731 385	PK 730/731 385	4035
17	14					PK 736/735 385		PK 736/735 385						PK 715 K - I 400	PK 721 I - L 300	L	PK 702 L - B 345	B	PK 708 B - K 315														2030
18	16			(730)	PK 750/751 625	D	PK 752/753 625	(723)						PK 717A K - L 745	L	L	L	PK 704 L - C 415	C	PK 711 C - L 440	PK 716 L - I 295	I	PK 722 I - K 375										3515
19	13			(730)	PK 750/751 625	D	PK 752/753 620	(723)						PK 717B K - L 745	L	L	L	PK 706 L - I 385	I	PK 712 I - K 455													2830
20	19														PK 721 K - I 35	I	I	PK 705 I - L 295	L	PK 712 L - I 375	PK 716 I - K								PK 719 K - C 310	C	C	PK 704 C - K 285	2070

Table 61 contd.

NO	NAME OR CODE NO.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	TOTAL TIME IN MTS
		MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	
21	22							PK 719 K - C 310	C	C	PK 704 C - K 285					PK 701 K - B 325	PK 703 B - L 370	L	L	PK 710 L - C 415	C	PK 719 C - L 440	PK 722 L - I 295	I	PK 706 I - K 455								2895
22	21								PK 738/737 385	PK 736/735 385	PK 738/737 385			PK 736/735 385			PK 703 K - B 370	PK 707 B - L 375	L	L	L	PK 720 L - K 710											2995
23	24											PK 704/735A 385	PK 724/723 385		PK 724/723 385			PK 705 K - I 400	PK 709 I - L 305	L	PK 714 L - B 355	PK 718 B - K 305											2520
24	28	PK 209/208 510			PK 207/206 510								PK 736/735 385		PK 736/735 385			PK 707 K - B 270	B	PK 713 B - L 405	L	PK 720 L - K 710								PK 209/208 510			3630
25	30					PK 711 K - C 310	PK 710 C - K 285												PK 709 K - I 475	I	PK 715 I - L 315	PK 718 L - B 425	B	PK 702 B - K 325									2135
26	25								PK 730/731 385	PK 730/731 385	PK 730/731 385			PK 712/719 385						PK 713 K - B 345	B	B	PK 701 B - L 370	L	L	PK 708 L - B 355	B	PK 714 B - K 270					2650
27	27			PK 745/744 510				PK 745/744 510				PK 706/711 385	PK 730/731 385		PK 730/731 385						PK 715 K - I 400	PK 721 I - L 300	L	PK 702 L - B 345	B	PK 708 B - K 315						PK 745/744 510	[illegible]
28	23										(730)	PK 750/751 625	D	PK 752/753 625	(723)						PK 717 K - L 745	L	L	L	PK 704 L - C 415	C	PK 711 C - L 440	PK 716 L - I 295	I	PK 722 I - K 375			3515
29	26										(730)	PK 750/751 625	D	PK 752/753 620	(723)						PK 717 K - L 745	L	L	L	PK 706 L - I 385	I	PK 712 I - K 455						2830
30	29								[illegible]/723 385	PK 724/723 385	PK 724/723 385			PK 710/735A 385								PK 721 K - I 490	I	I	PK 705 I - L 315	L	PK 712 L - I 295	PK 716 I - K 375					3015

Table 61 contd.

NO.	NAME OR CODE NO.	1 MON	2 TUE	3 WED	4 THU	5 FRI	6 SAT	7 SUN	8 MON	9 TUE	10 WED	11 THU	12 FRI	13 SAT	14 SUN	15 MON
31	35														PK 719 K – C 310	C
32	32												PK 711 K – C 310	PK 710 C – K 285		
33	39								PK 209/208 510			PK 207/206 510				
34	34										PK 745/740 510				PK 745/744 510	
35	36															PK 724/723 385
36	31															PK 738/737 385
37	33															PK 730/731 385
38	40															
39	37															
40	38															

NO.	NAME OR CODE NO.	16 TUE	17 WED	18 THU	19 FRI	20 SAT	21 SUN	22 MON	23 TUE	24 WED	25 THU	26 FRI	27 SAT	28 SUN	29 MON	30 TUE	31 WED	TOTAL TIME IN MTS.
31	35	C	PK 704 C – K 285					PK 701 K – B 325	PK 703 B – L 370	L	L	PK 710 L – C 415	C	PK 719 C – L 440	PK 722 L – I 295	I	PK 706 I – K 455	2895
32	32				PK 736/735 385		PK 736/735 385		PK 703 K – B 370	PK 707 B – L 375	L	L	L	PK 720 L – K 710				2720
33	39									PK 705 K – I 400	PK 709 I – L 305	L	PK 714 L – B 355	PK 718 B – K 305				2385
34	34			PK 706/711 385	PK 730/731 385		PK 730/731 385			PK 707 K – B 270	B	PK 713 B – L 450	L	PK 720 L – K 710				3495
35	36	PK 724/723 385	PK 724/723 385			PK 710/735A 385					PK 709 K – I 475	I	PK 715 I – L 315	PK 718 L – B 425	B	PK 702 B – K 325		3170
36	31	PK 736/735 385	PK 738/837 385			PK 736/735 385						PK 713 K – B 345	B	B	PK 701 B – L 370	L	L	2255
37	33	PK 730/731 385	PK 730/731 385			PK 736/735 385							PK 715 K – I 400	PK 721 I – L 300	L	PK 702 L – B 345	B	2585
38	40		(730)	PK 750/751 625	D	PK 752/753 620	(723)						PK 717 K – L 745	L	L	L	PK 704 L – C 415	2405
39	37		(730)	PK 750/751 625	D	PK 752/753 620	(723)						PK 717 K – L 745	L	L	L	PK 706 L – I 355	2375
40	38			PK 704/735A 385	PK 724/723 385		PK 724/723 385							PK 721 K – I 490	I	I	PK 705 I – L 315	1960

TABLE 61 :- MONTHLY CR SCEDULE FOR KARACHI BASE

10.9) Dacca Base

Shows that there should be only three pilots to operate the Schedule at Dacca base. The set theory technique is not suitable for such a small Schedule. They can be assigned on the basis of turn by turn. On this principle the monthly OR Schedule is prepared for Dacca base which is as follows.

NO.	NAME OR CODE NO.	1	2	3	4	5	6	7	8	9	10
		MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED
1	1	PK 725/726 305	PK 725/726 305		PK 706/711A 270						
2	2			PK 725/726 305			PK 712/719A 270		PK 725/726 305	PK 725/726 305	
3	3					PK 725/726 305		PK 725/726 305			PK 725/726 305

11	12	13	14	15	16	17	18	19	20	21	22
THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON
	PK 725/726 305		PK 725/726 305			PK 725/726 305			PK 712/719A 270		
PK 706/711A 270								PK 725/726 305		PK 725/726 305	
		PK 712/719A 270		PK 725/726 305	PK 725/726 305		PK 706/711A 270				

23	24	25	26	27	28	29	30	31	TOTAL TIME IN MTS.
TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	
PK 725/726 305		PK 706/711A 270							2945
	PK 725/726 305			PK 712/719A 270		PK 725/726 305	PK 725/726 305		3250
			PK 725/726		PK 725/726			PK 725/726	2880

TABLE 61:- MONTHLY OR SCHEDULE FOR DACCA BASE

CHAPTER XI

ESTIMATION OF RESERVE CREWS

11.1) Introduction.

In this Chapter, a study of reserve crew scheduling is undertaken to determine the possible means of achieving satisfactory reserve coverage at the least cost. Satisfactory coverage may be interpreted as having enough reserve crew available each day to cover the expected demands for emergency flying. Utilization of reserve crew can vary greatly. However, a large amount is given to reserve crews. Here problems arise to minimise the number of reserve crews by some mathematical method.

Multiple regression analysis can be applied to estimate the reserve crew in a given month on the basis of previous months information.

11.2) Factors Influencing Reserve Utilization.

Reserve crew is related to several factors such as weather, vacations, regular crew loads, sickness, leaves of absence, training, meetings etc.

It is better to include the important factors rather than all to estimate the reserve crews for future planning. The important factors are

X_1 = Number of regular flights in a month

X_2 = Ratio of reserve to regulars in a month

X_3 = Number of training days in a month

X_4 = Number of leaves (sickness, holidays, casual etc.,)

The equation of the model for estimating the number of reserve crew is

No. of Reserve crew (y)

$$= \text{Const} + \beta_1 X_1 + \beta_2 X_2 + \beta_3 x_3 + \beta_4 X_4$$

11.3) <u>Multiple Regression Analysis.</u>

The linear regression relation or straight line relation between two variables is very simple and is probably familiar to all who have occasion to consider the relation between variables. It requires only elementary technique for its estimation. Sometimes it is required to have a relationship between more than two variables. It is useful to express such a relationship in the form of mathematical equations connecting the variables. Then the value of the dependent variable can be predicted from the knowledge of other independent variables. If there are only two variables, then it is called a simple regression analysis. If there are more than two variables, then it is multiple regression analysis.

In general, with one dependent variable y and p independent variables $x_1, x_2 \ldots\ldots x_p$. The relationship will take the form

$$E_x(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

Before determining a multiple regression equation, it is worth while to give some thought to the selection of independent variables. First of all it is worth while to include only those variables that are likely to make an important contribution to the effectiveness of the relationship. Secondly, independent variables that are readily measurable or obserable should be selected.

It is undesirable to include so many variables in the regression equation. Three or four variables if suitably chosen will give a satisfactory relationship.

11.4) Estimation of Multiple Regression Coefficients.

The same principle of estimating the multiple regression coefficients is the same as in the case of simple regression coefficients. If there are n sets of values of one dependent variable and p independent variables, the sum of square

$$\Sigma (y - \beta_0 - \beta_1 x_1 - \beta_2 x_2 \;\ldots\ldots - \beta_p x_p)^2$$

is to be minimized.

This is reduced by using the following notations

$$u = \Sigma (y - \bar{y})^2$$

$$p_i = \Sigma y(x_i - \bar{x}_i)$$

$$t_{ii} = \Sigma (x_i - \bar{x}_i)^2$$

$$t_{hi} = \Sigma (x_n - \bar{x}_n)(x_i - \bar{x}_i)$$

to

$$u - \Sigma \beta_i p_i + \Sigma \beta_i (\beta_1 t_{i1} + \beta_2 t_{i2} + \ldots \beta_p t_{ip} - p_i) \qquad (11.4.1)$$

The normal equations are obtained by differentiating (11.4.1) and setting the equations equal to zero and also replacing the β_i by their estimate b_i. The above relation becomes

$$b_1 t_{i1} + b_2 t_{i2} + \ldots\ldots + b_p t_{ip} = p_i \qquad (11.4.2)$$

$$i = 1,2 \ldots p$$

or $$\Sigma b_h t_{ih} = p_i$$

Here the question arises to determine the value of b_i. Various methods are available to estimate the regression coefficients. Two methods will be used to calculate the value of b_i.

1) Crout method

2) Fisher method.

These methods will be explained by an example. Unfortunately

11.4) contd.

the data for this problem was not supplied by PIA. However, fictitious data is taken to demonstrate the estimation of the crew utilization. Only X_1 (number of regular flights) is calculated from the time table.

11.5) Example.

The variables are explained in the beginning of this Chapter.

Y	X_1	X_2	X_3	X_4
60	351	22.79	47	160
50	316	22.78	70	125
47	352	22.72	60	165
55	337	22.84	65	131
47	352	22.72	71	150
59	338	22.78	77	141
67	347	22.77	57	147
49	355	22.82	63	167
63	346	22.83	79	175
70	339	22.71	80	190
63	342	22.81	57	170
57	346	22.83	63	157

To speed up the calculation a computer program No. 10 is developed calculating the mean of variables and attached in the Appendix. The computer took only 8 seconds for calculating all means.

$$\bar{X}_1 = 343.42$$
$$\bar{X}_2 = 22.78$$
$$\bar{X}_3 = 65.75$$
$$\bar{X}_4 = 156.500$$
$$\bar{Y} = 57.25$$

Now the problem is to calculate the sum of squares and products of deviations from the mean. These values are calculated for each of the variables and set out in the following

11.5) contd.

table. Another computer program No.9 is developed and attached in the Appendix. The computer book only 13 seconds to calculate the sum of squares and products.

	x_1	x_2	x_3	x_4	y
x_1	1208.9168	-0.3868	-405.7500	1124.500	-62.2500
x_2	- 0.3868	0.0233	- 0.9600	-2.0900	0.2600
x_3	-405.7500	-0.9600	1104.2500	219.5000	89.7500
x_4	1124.5000	-2.0900	219.5000	3857.0000	679.5000

Table 62. Sum of Squares and Products.

The matrix so obtained is a square symmetric matrix. There are many methods available to calculate the regression co-efficients, but here two methods can be used to calculate the regression coefficients which are given by Crout[5] and Fisher[9]. In the former, the desk calculator is used and it took a considerable time. At present, the time factor plays an important role, so it is advisable to use the electronic computer to speed up the calculations. In the second method a computer program No.10 is developed[25] to invert the original matrix to calculate the regression co-efficients.

11.6) Crout Method.

The work of solving a system of original matrix is largely concentrated in the determination of an auxiliary matrix. This method is particularly good when the calculations are made on the desk calculator. Each element is determined by one continuous machine operation i.e. the sum of the products

11.6) contd.

with or without final division. If the matrix is symmetrical, then the calculations are cut almost in half.

In this method, the original matrix is transformed into auxiliary matrix. The procedure for obtaining the auxiliary matrix from the given matrix is described below. The auxiliary matrix is:-

Table 63-Auxiliary Matrix

	x_1	x_2	x_3	x_4	y
x_1	1208.9168	-.0003199	-.3356	.9301	-.0515
x_2	- 0.3868	.0233	-46.7725	-74.2575	9.8755
x_3	-405.75	-1.0898	917.7076	0.5624	.0868
x_4	1124.500	-1.7302	515.9564	2392.3967	0.2966

1) Take the first element of the first column and the remaining elements of the first row, that is, the first column of the auxiliary matrix is idential with the first column of the original matrix. This is only due to the symmetrix matrix.
2) The first row of the auxiliary matrix, apart from the first term, is obtained by dividing the corresponding element in the original matrix by the first element in the uaxiliary matrix.
3) Each element on or below the principal diagonal is equal to the corresponding elements of the original matrix minus

11.6) contd.

3) contd.

the sum of products of elements in its row and corresponding elements in its column in the uaxiliary matrix that involve only previously computed elements.

4) For each to the right of the principal diagonal the value so obtained by step 3 is divided by the diagonal elements, that is, it is divided by the diagonal element in the auxiliary matrix.

Each element to the right of the principal diagonal is seen to be equal to the corresponding element below the principal diagonal,divided by the diagonal element. This fact reduces the computation considerably when the original matrix for the independent variables is symmetric. The operation is carried right through to the final column, that is, the sum of the products with the dependent variable.

Now the remaining step is to calculate the one column final matrix from the auxiliary matrix which actually consists of the column of partical regression coefficient. The elements are determined in reverse order to the elements of the auxiliary matrix, that is, the last element of the last column in the auxiliary matrix will become the last element in the final matrix. Each other element in the final matrix is equal to the corresponding element of the last column of the auxiliary matrix, minus the sum of the products of elements in its row in the auxiliary matrix and the corresponding element in its column in the final matrix that have been previously computed.

From the auxiliary matrix, the regression coefficients are calculated:-

11.6) contd.

$b_1 = - 0.3453$

$b_2 = 28.161$

$b_3 = -0.08004$

$b_4 = 0.2966$

One thing is to be noted in this method, that the elements of the auxiliary matrix are used which lie to the right of the principal diagonal and to the left of the final column.

11.7) Continuous Check on Calculations.

It is often desirable to carry a check column to ensure accuracy at each stage of the calculations. Crout has suggested to apply a check during the calculations. First the check columns are calculated as below:-

Original Matrix	Auxiliary Matrix	Final Matrix
1865.0300	1.5430	0.6547
- 3.1634	-110.1547	29.161
1006.7900	1.6494	0.9199
5878.4100	1.2966	1.2966

In the original matrix, all the elements of a row are added to get the corresponding element of the check column.

In the auxiliary matrix every element in the check column is equal to one plus the sum of other elements in its row which lie to the right of the principal diagonal.

In the final matrix, every element in the check column is equal to one plus the sum of the other element.

Thus the check column is written at the right of the

11.7) contd.

given matrix. This column is now treated exactly in the same manner as the last column of the given matrix, the calculations being carried along with those for other columns.

If the last column of the given matrix is replaced by the check column for transforming the original matrix into auxiliary matrix, then the final column of the auxiliary matrix will be identical to the check column of the auxiliary matrix. The same procedure is applied to check the calculations of transforming the auxiliary matrix into the final matrix.

11.8) Fisher Method.

In the first method only those elements are used which lie on the right of the diagonal elements and to the left of the final columns. Fisher[9] suggested that the final column of the original matrix should be replaced by

$$\begin{bmatrix} t^{11} & t^{12} & \cdots & t^{1n} \\ t^{21} & t^{22} & \cdots & t^{2n} \\ \vdots & & & \\ t^{m1} & t^{m2} & \cdots & t^{mn} \end{bmatrix}$$

and set the whole in this form

11.8) contd.

$$\begin{bmatrix} \Sigma x_1^2 & \Sigma x_2x_1 & \cdots & \Sigma x_1x_n \\ \Sigma x_1x_2 & \Sigma x_2^2 & \cdots & \Sigma x_2x_n \\ \vdots & \vdots & \vdots & \vdots \\ \Sigma x_1x_m & \Sigma x_2x_m & \cdots & \Sigma x_mx_n \end{bmatrix} \begin{bmatrix} t^{11} & t^{12} & \cdots & t^{1n} \\ t^{21} & t^{22} & \cdots & t^{2n} \\ \vdots & & & \\ t^{m1} & t^{m2} & & t^{mn} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & & 0 \\ \vdots & \vdots & 1 & & \vdots \\ \vdots & \vdots & 0 & & \vdots \\ \vdots & \vdots & & & 1 \end{bmatrix}$$

This means $(T) \times (T)^{-1}$ = unit matrix

After transforming the original matrix into inverse matrix, the regression coefficients are obtained directly by multiplying the successive sum of the products of the dependent variable with the independent variables by the corresponding elements in a column of the inverse matrix, i.e.

$$b_i = \sum_h p_h t^{hi}$$

The original matrix is transformed into the inverse matrix and given in the undermentioned table.

Table No. 64 Inverse matrix.

0.0014784	0.0095054	0.00064188	-0.00046135
0.0095054	46.316	0.039770	0.020063
0.00064188	0.039770	0.0012228	-0.00023517
-0.00046135	0.020063	-0.00023517	0.00041803

Here again a computer program No.10 is developed from the procedure[25] to invert the matrix so that calculations can be made easily and correctly. The computer took only 10 seconds

11.8) contd.

to invert the matrix.

Estimation of regression coefficients.

As stated above the regression coefficients can be estimated from the inverse matrix as follows:-

$$b_1 = (0.0014748)(-62.25) + (0.0095054)(0.25) + (0.00064188)(89.75) + (-0.00046135)(679.5)$$

$$= -0.345$$

$$b_2 = (0.0095054)(-62.25) + (46.316)(0.25) + (0.03977)(89.75) + (0.020063)(679.5)$$

$$= 28.18$$

$$b_3 = (0.00064188)(-62.25) + (0.03977)(0.25 + (0.0012228)(89.75) + (-0.00023517)(679.5)$$

$$= -0.08004$$

$$b_4 = (-0.00046135)(-62.25) + (0.020063)(0.25)+(-.00023517)(89.75) + (0.00041803)(679.5)$$

$$= 0.2966$$

The same result is obtained as in the first method. Therefore, the estimated regression equation is

$$Y = -507.2284 - 0.3453\ X_1 + 28.18\ X_2 - .08004\ X_3 + .2966\ X_4$$

If the values of X_1, X_2, X_3 and X_4 are known, then the value of Y (reserved crew) can be estimated for the future.

In this equation two regression coefficients b_1 and b_3 are negative. The former is the coefficient of regular number of flights in a month which cannot be increased or decreased because it is a fixed number. The latter is the coefficients of the number of training days in a month which can be increased or decreased. It is obvious if the more number of training days are reserved for training of crews, then the number of reserve crew will have to be decreased in order to meet the requirements of the regular flight.

11.9) Tolerance Limits.

It is advisable to have some limits on the estimation of reserve crew. The tolerance limits of Y can be calculated by this relation

$$Y \pm t \sqrt{[s^2 + V(Y)]}$$

where s^2 is the mean of sum of squares and can be estimated by this relation

$$s^2 = \frac{u - \sum_i b_i p_i}{n-p-1}$$

The variance of Y can be estimated by this relation

$$V(Y) = s^2 \left[\frac{1}{n} + \sum_h \sum_i (x_h - \bar{x}_h)(x_i - \bar{x}_i) t^{hi} \right]$$

The tabulated value of t will determine the percentage of tolerance limits.

A P P E N D I X

REFERENCES.

1) BALAS, E., "An Additive Algorithm for solving Linear Programs with zero-one Variable". Operations Research, 1965, pp.517-546.

2) BUFFA, E.S., "Operations Management", John Wiley & Sons, 1968.

3) BYRNE, J.L. & PROLL, "Solution of Linear Programming 0-1" Comm. ACM Vol.11, No.11, 1964, pp.782.

4) CHARNES A., COOPER, W.W., "The Stepping-Stone Method of Explaining Linear Programming Calculations in Transportation Problem". Management Science Vol.1, No.1, Oct. 1954.

5) CROUT, P.D. "A Short Method for Evaluating determinants and solving systems of Linear Equations with Real or Complex Coefficients". Trans. A.I.E.E. 61, 1940, pp.1235-1241.

6) DANTZIG, B.G. "Application of Simplex Method to a Transportation Problem". "Activity and Analysis of Production and Allocation" Edited by Koopman, T.C.

7) ——— "Linear Programming and Extension" Princeton University Press, 1963.

8) ——— "Maximization of Linear function of Variables Subject to Linear Equalities". "Activity and Analysis of Production and Allocation" Edited by Koopman,T.C.

9) FISHER, R.A., "Statistical Methods for Research Worker", Oliver and Boyd, Edinburgh.

10) FLOOD, M.M., "Travelling Salesman Problem". Operations Research 4, No.1, 1956, pp.61-75.

11) FORD, L.P., and FULKERSON, D.L., "Solving the Transportation Problem", "Management Science", 1956-1957, pp.24-32.

12) GEOFFRION, A., "Integer Programming by Implicit Enumeration and Balas Method". SIAM Review Vol.9, No.2, April 1967, pp.178-190.

13) GOMORY, R.E., "Outline of an Algorithm for Integer Solution to Linear Programs". Bulletin of the American Math.Soc. Vol.64, 1958, pp.275-278.

14) ——— "All-Integer Programming Algorithm in Industrial Scheduling". Editors Muth & Thompson, Princeton-Hall, 1963 pp.193-206.

15) HADLEY, G., "Linear Programming". Addison Wesley Publishing Co., London, and CACM 258.

16) HARRIS, M.J., "An Algorithm for Solving mixed Integer Programs". Operational Research Quarterly. Vol.15, 1964, pp.117-132.

17) HEALEY, W.C., "Multiple Choice Programming". Operations Research, Vol.12, 1964, pp.122-138.

18) HITCHCOCK, F.L., "The Distribution of a product from Several Sources to Numerous Locality". J.Math.Phys. 20, 1941, pp.224-250.

19) JAMES MUNKERS, "Algorithm for the Assignment and Transportation Problem". J.Soc.Indus.App.Math.Vol.5, No.1, March 1957 pp.32-38.

20) KING, W.R., "A Stochastic Personnel Assignment Model", Operations Research, Vol.13, pp.67-81

21) KUHN, W.R., "Hungarian Method for Assignment Problem", Naval Research Logistic Quarterly, 2, Nos.1 and 2, 1955, pp.83-98.

22) "Variants of the Hungarian Methods for the Assignment Problem". Naval Research Logistic Quarterly, Vol.3, No.4, 1956, pp.253-258.

23) LAGEMAN, J.J., "A Method for Solving the Transportation Problem", Naval Research Logistic Quarterly, Vol.14, Part I, 1967, pp.81-99.

24) MORTON KLEIN, "A Primal Method for Minimal Cost Flows with Applications to Assignment and Transportation Problem". Management Science, Vol.14, No.3, 1967, pp.205-220.

25) RUTHISHAUSER, H. Comm.ACM 6, Feb. 1963, pp.67.

<u>COMPUTER PROGRAMS</u>

PROGRAM NO. 1

```
BEGIN'      'COMMENT'   DAILY LAYOVER TIME MATRIX;
    'INTEGER' I,J,N,ZERODAY;
    'REAL' ZEROHOURS,T1,T2,TIME;
COMMENT'
    THE DATA CARDS SHOULD BE PREPARED AS FOLLOWS:
    CITY //
    (THAT IS, A TITLE TERMINATED BY //)
    DAY OF WEEK AND TIME OF DAY FROM WHICH TIME IS TO START
    (SUNDAY=1, MONDAY=2,.......SATURDAY=7    MIDNIGHT IS 24.00 HOURS)
    NUMBER OF ARRIVALS CONSIDERED (SIZE OF MATRIX)
    FOR EACH AIRCRAFT THE DAY OF WEEK CODE NUMBER OF ARRIVAL, TIME OF
    ARRIVAL IN HOURS PAST MIDNIGHT, CODE OF DEPARTURE DAY, AND
    DEPARTURE TIME IN HOURS PAST MIDNIGHT;

 L1: NEWLINE(1);
    COPYTEXT('('//')');
    T1:=READ;
    T2:=READ;
    ZERODAY:=READ;
    ZEROHOURS:=READ;
    NEWLINE(4);
    WRITETEXT('('CALCULATIONS%ARE%MADE%FROM%DAY')');
    WRITE(0,FORMAT('('-DS')'),ZERODAY);
    WRITETEXT('('AT')');
    WRITE(0,FORMAT('('-DD.DDS')'),ZEROHOURS);
    WRITETEXT('('HOURS'('4C')'')');
    N:=READ;
    'BEGIN'
          'REAL' 'ARRAY' ARRTIME,DEPTIME[1:N];
          'INTEGER' 'ARRAY' DAYARRIVAL,DAYDEPARTURE[1:N];
          'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
          'BEGIN'
                DAYARRIVAL[I]:=READ;
                ARRTIME[I]:=READ;
                DAYDEPARTURE[I]:=READ;
                DEPTIME[I]:=READ;
          'END';
          'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
          'BEGIN'
                'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
                'BEGIN'
                      TIME:=(DAYDEPARTURE[I]-DAYARRIVAL[J])*24+DEPTIME
                         [I]-ARRTIME[J]+24;
                         'IF' TIME>T1 'THEN' TIME:=TIME-24;
                         'IF' TIME<T2 'THEN' TIME:=TIME+24;
                         WRITE(0,FORMAT('('-NNNND,DD')'), TIME);
                'END';
                 NEWLINE(2);
          'END';
    'END';
    'GOTO' L  1;
 'END';

****
```

PROGRAM NO.2

```
BEGIN'    'COMMENT' TRANPOSED  DAILY LAYOVER TIME MATRIX;
     'INTEGER' I,J,N,ZERODAY;
     'REAL' ZEROHOURS,T1,T2,TIME;
COMMENT'
     THE DATA CARDS SHOULD BE PREPARED AS FOLLOWS:
     CITY //
     (THAT IS, A TITLE TERMINATED BY //)
     DAY OF WEEK AND TIME OF DAY FROM WHICH TIME IS TO START
     (SUNDAY=1, MONDAY=2,.......SATURDAY=7    MIDNIGHT IS 24.00 HOURS)
     NUMBER OF ARRIVALS CONSIDERED (SIZE OF MATRIX)
     FOR EACH AIRCRAFT THE DAY OF WEEK CODE NUMBER OF ARRIVAL, TIME OF
     ARRIVAL IN HOURS PAST MIDNIGHT, CODE OF DEPARTURE DAY, AND
     DEPARTURE TIME IN HOURS PAST MIDNIGHT;

  L1: NEWLINE(1);
     COPYTEXT('('//')');
     T1:=READ;
     T2:=READ;
     ZERODAY:=READ;
     ZEROHOURS:=READ;
     NEWLINE(4);
     WRITETEXT('('CALCULATIONS%ARE%MADE%FROM%DAY')');
     WRITE(O,FORMAT('('-DS')'),ZERODAY);
     WRITETEXT('('AT')');
     WRITE(O,FORMAT('('-DD.DDS')'),ZEROHOURS);
     WRITETEXT('('HOURS'('4C')'')');
     N:=READ;
     'BEGIN'
          'REAL' 'ARRAY' ARRTIME,DEPTIME[1:N],C[1:N,1:N];
          'INTEGER' 'ARRAY' DAYARRIVAL,DAYDEPARTURE[1:N];
          'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
          'BEGIN'
               DAYARRIVAL[I]:=READ;
               ARRTIME[I]:=READ;
               DAYDEPARTURE[I]:=READ;
               DEPTIME[I]:=READ;
          'END';
          'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
          'BEGIN'
               'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
               'BEGIN'
                    TIME:=(DAYDEPARTURE[I]-DAYARRIVAL[J])*24+DEPTIM
                       [I]-ARRTIME[J]+24;
                       'IF' TIME>T1 'THEN' TIME:=TIME-24;
                       'IF' TIME<T2 'THEN' TIME:=TIME+24;
                    C[I,J]:= TIME ;
               'END';
              NEWLINE(2);
               'END';
          WRITETEXT('('TRANSPOSE '('2C')'')');
          'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
               'BEGIN'
            'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
              PRINT (C[J,I],2,2);
               NEWLINE(2);
          'END';

     'END';
  'GOTO' L1;
END';
```

PROGRAM NO. 3

```
BEGIN'
'COMMENT'  WEEKLY LAYOVER TIME MATRIX;
    'INTEGER' I,J,N,ZERODAY;
    'REAL' ZEROHOURS,T1,T2,TIME;
    'COMMENT'
    THE DATA CARDS SHOULD BE PREPARED AS FOLLOWS:
    CITY //
    (THAT IS, A TITLE TERMINATED BY //)
    DAY OF WEEK AND TIME OF DAY FROM WHICH TIME IS TO START
    (SUNDAY=1, MONDAY=2,.......SATURDAY=7     MIDNIGHT IS 24.00 HOURS)
    NUMBER OF ARRIVALS CONSIDERED (SIZE OF MATRIX)
    FOR EACH AIRCRAFT THE DAY OF WEEK CODE NUMBER OF ARRIVAL, TIME OF
    ARRIVAL IN HOURS PAST MIDNIGHT, CODE OF DEPARTURE DAY, AND
    DEPARTURE TIME IN HOURS PAST MIDNIGHT;
    L1: NEWLINE(1);
    COPYTEXT('('//')');
    T1:=READ;
    T2:=READ;
    ZERODAY:=READ;
    ZEROHOURS:=READ;
    NEWLINE(4);
    WRITETEXT('('CALCULATIONS%ARE%MADE%FROM%DAY')');
    WRITE(0,FORMAT('('-DS')'),ZERODAY);
    WRITETEXT('('AT')');
    WRITE(0,FORMAT('('-DD.DDS')'),ZEROHOURS);
    WRITETEXT('('HOURS'('4C')'')');
    N:=READ;
    'BEGIN'
        'REAL' 'ARRAY' ARRTIME,DEPTIME[1:N];
        'INTEGER' 'ARRAY' DAYARRIVAL,DAYDEPARTURE[1:N];
        'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
        'BEGIN'
            DAYARRIVAL[I]:=READ;
            ARRTIME[I]:=READ;
            DAYDEPARTURE[I]:=READ;
            DEPTIME[I]:=READ;
        'END';
        'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
        'BEGIN'
            'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
            'BEGIN'
                TIME:=(DAYDEPARTURE[I]-DAYARRIVAL[J])*24+DEPTIM
                    [I]-ARRTIME[J]+168;
                'IF' TIME>T1 'THEN' TIME:=TIME-168;
                'IF' TIME<T2 'THEN' TIME:=TIME+168;
                WRITE(0,FORMAT('('-NNNNND.DD')'),TIME);
            'END';
            NEWLINE(2);
        'END';
    'END';
            'GOTO' L1;
END';
***
```

PROGRAM NO. 4

```
BEGIN'                'COMMENT' TRANSPOSED WEEKLY TIMEMATIX;
     'INTEGER' I,J,N,ZERODAY;
     'REAL' ZEROHOURS,T1,T2,TIME;
     'COMMENT'
     THE DATA CARDS SHOULD BE PREPARED AS FOLLOWS:
     CITY //
     (THAT IS, A TITLE TERMINATED BY //)
     DAY OF WEEK AND TIME OF DAY FROM WHICH TIME IS TO START
     (SUNDAY=1, MONDAY=2,.......SATURDAY=7    MIDNIGHT IS 24.00 HOURS)
     NUMBER OF ARRIVALS CONSIDERED (SIZE OF MATRIX)
     FOR EACH AIRCRAFT THE DAY OF WEEK CODE NUMBER OF ARRIVAL, TIME OF
     ARRIVAL IN HOURS PAST MIDNIGHT, CODE OF DEPARTURE DAY, AND
     DEPARTURE TIME IN HOURS PAST MIDNIGHT;

L1:  NEWLINE(1);
     COPYTEXT('('//')');
     T1:=READ;
     T2:=READ;
     ZERODAY:=READ;
     ZEROHOURS:=READ;
     NEWLINE(4);
     WRITETEXT('('CALCULATIONS%ARE%MADE%FROM%DAY')');
     WRITE(O,FORMAT('('-DS')'),ZERODAY);
     WRITETEXT('('AT')');
     WRITE(O,FORMAT('('-DD.DDS')'),ZEROHOURS);
     WRITETEXT('('HOURS'('4C')'')');
     N:=READ;
     'BEGIN'
           'REAL' 'ARRAY' ARRTIME,DEPTIME[1:N],C[1:N,1:N];
           'INTEGER' 'ARRAY' DAYARRIVAL,DAYDEPARTURE[1:N];
           'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
           'BEGIN'
                 DAYARRIVAL[I]:=READ;
                 ARRTIME[I]:=READ;
                 DAYDEPARTURE[I]:=READ;
                 DEPTIME[I]:=READ;
           'END';
           'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
           'BEGIN'
                 'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
                 'BEGIN'
                       TIME:=(DAYDEPARTURE[I]-DAYARRIVAL[J])*24+DEPTIME
                             [I]-ARRTIME[J]+168;
                       'IF' TIME>T1 'THEN' TIME:=TIME-168;
                       'IF' TIME<T2 'THEN' TIME:=TIME+168;
                        C[I,J]:= TIME ;
                 'END';
            'END';
           WRITETEXT('('TRANSPOSE '('2C')'')');
           'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
           'BEGIN'
             'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
                PRINT (C[J,I],2,2);
           'END';
     'END';
      'GOTO' L1;

END';
**
```

PROGRAM NO. 5

```
'TRACE' 2
'BEGIN'
     'COMMENT' TRANSPORT PROGRAM;
  'INTEGER' I,J,M,N,INF,COST;
  'PROCEDURE' TRANSPORT (C,X,A,B,M,N,INF,COST);
  'VALUE' M,N,INF;
  'INTEGER' M,N,INF,COST;
  'INTEGER' 'ARRAY' C,X,A,B;
  'BEGIN'
    'INTEGER' I,J,P,H,K,Y,T,L;
    'INTEGER' 'ARRAY' V,XSJ,S,R, LISTV[1:N],U,XIS,D,G,LISTU[1:M];
    'BOOLEAN' 'ARRAY' XB[1:M,1:N];
    'INTEGER' 'PROCEDURE' SUM(I,A,B,X);
    'VALUE' A,B;
    'INTEGER' I,A,B,X;
    'BEGIN'
     'INTEGER' S;
     S := 0;
     SELECTOUTPUT(0);
     'FOR' I := A 'STEP' 1 'UNTIL' B 'DO'
     S := S+X;
     SUM := S
    'END';
    'FOR' I := 1 'STEP' 1 'UNTIL' M 'DO'
     XIS[I] :=A[I];
    'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
     XSJ[J] := B[J];
    'FOR' I := 1 'STEP' 1 'UNTIL' M 'DO'
    'BEGIN'
      H := INF;
     'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
     'BEGIN'
        X[I,J] := 0;
        P := C[I,J];
       'IF' P < H 'THEN' H := P
     'END';
      U[I] := H;
     'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
      XB[I,J] := 'IF' C[I,J] = H  'THEN' 'TRUE' 'ELSE' 'FALSE'
    'END' U[I];
    'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
    'BEGIN'
      H := INF;
     'FOR' I := 1 'STEP' 1 'UNTIL' M 'DO'
     'BEGIN'
       'IF' XB[I,J] 'THEN'
       'BEGIN'
          V[J] := 0;
          'GOTO' AA
       'END';
       D[I]:=P:=C[I,J] - U[I];
```

P.NO.5) Contd.

```
        'IF' P<H 'THEN' H := P
       'END';
        V[J] := H;
       'FOR' I := 1 'STEP' 1 'UNTIL' M 'DO'
       'BEGIN'
         'IF' D[I] = H 'THEN' XB[I,J] :=  'TRUE'
       'END';
   AA:'END' V[J];
     'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
      LISTV[J] := 0;
     'FOR' I := 1 'STEP' 1 'UNTIL' M  'DO'
      LISTU[I] := 0;
   S2: 'FOR'  I := 1 'STEP' 1 'UNTIL' M 'DO'
     'BEGIN'
       'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
       'BEGIN'
         'IF'  XB[I,J]  'THEN'
         'BEGIN'
            H := X[I,J] := 'IF' XSJ[J]  'LE' XIS[I] 'THEN'
                                XSJ[J] 'ELSE' XIS[I];
            XSJ[J] := XSJ[J] - H;
            XIS[I] := XIS[I] - H;
         'END'
       'END'
     'END';
   S03: 'IF' SUM(J,1,N,XSJ[J])=0 'THEN' 'GOTO' S6;
      'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
       S[J] := R[J] := 0;
       H := 0;
       K := 1;
    S3: 'FOR' I := 1 'STEP' 1 'UNTIL' M 'DO'
      'BEGIN'
        'IF' XIS[I] >0 'THEN'
        'BEGIN'
           D[I] := XIS[I];
           G[I] := 2*N;
          'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
          'BEGIN'
             'IF' XB[I,J] 'AND'  R[J]=0 'THEN'
             'BEGIN'
                S[J] := D[I];
                R[J] := I;
                LISTV[K] := J;
                K := K+1;
               'IF' XSJ[J] > H 'THEN'
               'BEGIN'
                  H := XSJ[J];
                  P := J
               'END'
             'END'
          'END'
        'END' 'ELSE' D[I] := G[I] := 0
      'END';
```

P.NO.5) Contd.

```
  S53: 'IF' K=1 'THEN'  'GOTO' S13;
       L := 1;
      'FOR' K := 1 'STEP' 1 'UNTIL'  N 'DO'
      'BEGIN'
         J := LISTV[K];
         LISTV[K]:= 0;
        'IF' J=0 'THEN' 'GOTO' S33;
        'FOR' I := 1 'STEP' 1 'UNTIL' M 'DO'
        'BEGIN'
          'IF' XB[I,J] 'AND' X[I,J] > 0 'AND' G[I]=0 'THEN'
          'BEGIN'
             D[I] := 'IF' X[I,J] 'LE' S[J] 'THEN' X[I,J] 'ELSE' S[J];
             G[I] := J;
             LISTU[L] := I;
             L := L+1
          'END'
        'END'
      'END';
 S33: 'IF' L =1 'THEN' 'GOTO' S13;
       K := 1;
      'FOR' L := 1 'STEP' 1 'UNTIL' M 'DO'
      'BEGIN'
        I := LISTU[L];
         LISTU[L] := 0;
        'IF' I=0 'THEN' 'GOTO' S43;
        'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
        'BEGIN'
          'IF' XB[I,J] 'AND' R[J] = 0 'THEN'
          'BEGIN'
             S[J] := D[I];
             R[J] := I;
             LISTV[K] := J;
             K := K+1;
            'IF' XSJ[J] > H 'THEN'
            'BEGIN'
               H := XSJ[J];
               P := J
            'END'
          'END'
        'END'
      'END';
S43:  'GOTO' S53;
S13:  'IF' H>0 'THEN' 'GOTO' S4 'ELSE' 'IF' SUM(J,1,N,XSJ[J])=0
                                  'THEN' 'GOTO' S6 'ELSE' 'GOTO' S5;
 S4: K := P;
     H := 'IF' S [K]< XSJ[K] 'THEN' S[K] 'ELSE' XSJ[K];
S41: Y := R[K];
     X[Y,K] := X[Y,K] + H;
     XIS[Y] := XIS[Y]-H;
     XSJ[K] := XSJ[K]-H;
     T := G[Y];
    'IF' T=2*N 'THEN' 'GOTO' S03;
     X[Y,T] := X[Y,T] -H;
     XIS[Y] := XIS[Y] +H;
     XSJ[T] := XSJ[T] +H;
```

P.NO.5) Contd.

```
     K := T;
     'GOTO' S41;
   S5:H := INF;
     'FOR' I := 1 'STEP' 1 'UNTIL' M 'DO'
     'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
     'BEGIN'
       'IF' G[I] 'NE' O 'AND' R[J]=O 'THEN'
       'BEGIN'
          P := C[I,J] - U[I] - V[J];
         'IF' P < H 'THEN' H:= P
       'END'
     'END';
     'FOR' I := 1 'STEP' 1 'UNTIL' M 'DO'
     'BEGIN'
       'IF' G[I] 'NE' O 'THEN'  U[I]:=U[I] + H
     'END';
     'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
     'BEGIN'
       'IF' R[J] 'NE' O 'THEN' V[J]:=V[J] - H
     'END';
     'FOR' I := 1 'STEP' 1 'UNTIL' M 'DO'
     'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
      XB[I,J] := C[I,J] = U[I] + V[J];
     'GOTO' SO3;
   S6:COST := SUM(I,1,M,A[I]*U[I]) + SUM(J,1,N,B[J]* V[J]);
   'END';
     COPYTEXT('(')%')');
    INF : = READ;
   'IF' INF<0.0 'THEN' 'GOTO' L2 'ELSE'
    N := READ;
    M := READ;
   'BEGIN'
     'INTEGER' 'ARRAY' A[1:M],B[1:N],C[1:M,1:N],X[1:M,1:N];
     'FOR' I := 1 'STEP' 1 'UNTIL' M 'DO'
     'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
      C[I,J] := READ;
     'FOR' I := 1 'STEP' 1 'UNTIL' M 'DO'
      A[I] := READ;
     'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
      B[J] := READ;
      TRANSPORT(C,X,A,B,M,N,INF,COST);
      NEWLINE(3);
     'FOR' I := 1 'STEP' 1 'UNTIL' M 'DO'
     'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
     'BEGIN'
        NEWLINE(1);
        PRINT(C[I,J],10,2);
        PRINT(A[I],10,0);
        PRINT(B[J],10,0);
        PRINT(X[I,J],10,0);
        PRINT(COST,10,2);
     'END';
   'END';
   L2:;
'END';
****
```

PROGRAM NO.6

```
'BEGIN'        'COMMENT'  INTEGER PROGRAMMING;
  'INTEGER' I,J,M,N,COUNT;
   'INTEGER'  INF;
  'PROCEDURE' IMPLEN(M,N,A,X,API,NOSOLN,COUNT,INF);
  'VALUE' M,N,INF;
  'INTEGER' M,N,COUNT,INF;
  'BOOLEAN' API, NOSOLN;
  'REAL' 'ARRAY' A;
  'INTEGER' 'ARRAY' X;
     'BEGIN'
       'INTEGER' I,J,K,IA,E,D,MAX,LC;
       'REAL' Z,R;
       'BOOLEAN' NULL;
       'INTEGER' 'ARRAY' S,V[1:N];
       'INTEGER'  'ARRAY' Q[1:N];
       'IF' API 'THEN'
         'BEGIN'
             E := O;
            'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
            'IF' X[J] = O 'THEN'
             V[J] := O   'ELSE'
               'BEGIN'
                 E :=E + 1 ;
                 S[E] := J;
                 V[J] := 3;
               'FOR' I := 1 'STEP' 1 'UNTIL' M 'DO'
               A[I,O] := A[I,O] + A[I,J];
               'END';
               Z := A[O,O]; 'GOTO' LO;
       'END';
       'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
        S[J] := V[J] := O;
        Z := O.O;
        E := O;
  LO:   NOSOLN := 'TRUE';
        COUNT :=O;
        A[O,O] := INF;
  START: COUNT := COUNT + 1;
        'FOR' I:=1 'STEP' 1 'UNTIL' M 'DO'
        'IF' A[I,O] < O.O 'THEN' 'GOTO' FORMAT;
        'GOTO' INCUMBENT;
  FORMAT: NULL :='TRUE' ;
        'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
           'BEGIN'
              'IF' 'NOT' (V[J] = O 'AND' A[O,J] + Z < A[O,O])
              'THEN' 'GOTO' L1;
              'FOR' K := I 'STEP' 1 'UNTIL' M 'DO'
              'IF' A[K,O] < O.O  'AND' A[K,J] > O.O 'THEN'
                   'BEGIN' NULL := 'FALSE';
                      V[J] := 1;
                      'GOTO' L1
                      'END';
 L1:          'END';
          'IF' NULL 'THEN' 'GOTO' NEWS;
          'FOR' K:= I 'STEP' 1 'UNTIL' M 'DO'
```

P.NO.6) Contd.

```
            'BEGIN'
              'IF' A[K,O] 'GE' 0.0 'THEN' 'GOTO' L2;
               Q[1] := A[K,O];
              'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
              'IF' V[J] = 1 'AND' A[K,J] > 0.0        'THEN'
                Q[1] := Q[1] + A[K,J];
              'IF' Q[1] < 0 'THEN'
              'GOTO' NEWS;
 L2:        'END';
           MAX := -INF;
          'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
            'BEGIN'
              'IF' V[J] 'NE' 1 'THEN'
              'GOTO' L3;
                 Q[J] := 0.0;
              'FOR' I := 1 'STEP' 1 'UNTIL' M 'DO'
                 'BEGIN'
                    R := A[I,O] + A[I,J];
                   'IF' R < 0.0 'THEN'
               Q[J] := Q[J] + R;
                  'END';
                  'IF' MAX  'LE' Q[J]  'THEN'
                     'BEGIN'
                        MAX := Q[J];
                        D := J;
                     'END';
  L3:       'END';
             LC := INF;
              'FOR' J:= 1 'STEP' 1 'UNTIL' N 'DO'
             'IF'  MAX = Q[J] 'THEN'
               'BEGIN'
                 'IF' LC > A[O,J] 'THEN';
                   'BEGIN'
                     LC := A[O,J];
               'END';
                   'END'  ;
             E := E + 1;
             S[E] := D;
             V[D] := 3;
             IA := 1;
 RESET:     'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
            'IF' V[J] = 1 'THEN'
             V[J] := 0;
            'FOR' I := 1 'STEP' 1 'UNTIL'  M 'DO'
             A[I,O] := A[I,O] + IA * A[I,D];
             Z := Z + IA * A[O,D];
            'GOTO'  START;
 INCUMBENT:  NOSOLN := 'FALSE';
            'IF' Z 'GE' A[O,O] 'THEN'
            'GOTO' NEWS;
             A[O,O] := Z;
               'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
                  X[J] := 'IF' V[J] = 3 'THEN' 1 'ELSE' O;
 NEWS:         'IF' E = O 'THEN'
```

P.NO.6) Contd.

```
                'GOTO' RESULT;
 I4:             D := S[E];
                'IF' D > O 'THEN'
                'GOTO' UNDERLINE;
                 V[-D] := O;
                 E := E - 1;
                'GOTO' NEWS;
 UNDERLINE:      S[E] := -D;
                 V[D] := 2;
                 IA := -1;
                'GOTO' RESET;
 RESULT:
              'END';
          'BOOLEAN' API, NOSOLN;
              'BEGIN'
                COPY TEXT ('(')');
                NEW LINE (2);
                  INF := READ;
                  M := READ;
                  N := READ;
                  API := 'FALSE';
                    'BEGIN'
                      'ARRAY' A[O:M,O:N];
                      'INTEGER'  'ARRAY' X[1:N];
                      'FOR' I := O 'STEP' 1 'UNTIL' M 'DO'
                      'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
                       A[I,J] := READ;
                      'FOR' I := 1 'STEP' 1 'UNTIL' M 'DO'
                       A[I,O] := READ;
                       IMPLEN(M,N,A,X,API,NOSOLN,COUNT,INF);
                      'IF' NOSOLN 'THEN'
                        'BEGIN'
                           WRITETEXT('('NOLOLN')');
                        'END';
                        'FOR' J := 1 'STEP' 1 'UNTIL' N 'DO'
                         PRINT(X[J],3,O);
                         NEWLINE (5);
                        'FOR' I := 1 'STEP' 1 'UNTIL' M 'DO'
                         PRINT(A[I,O],3,O);
                         NEWLINE (5);
                         PRINT(A[O,O],3,2) ;
                         NEWLINE (5);
                         PRINT(COUNT,10,O);
                    'END';
            'END';
    'END';
****
```

PROGRAM NO. 7

```
     LIST (LP)
     PROGRAM  (ASSIGNMENT)
     INPUT 1=CRO
     OUTPUT 2=LPO
     TRACE 2
     END


     MASTER SEGMENT
     INTEGER X
     DIMENSION  D(20,20), X(20)
     COMMON N,D,X
     READ (1,102)M
     DO 40 II=1,M
     READ(1,103)CHAR
103  FORMAT(A8)
     WRITE(2,104)CHAR
104  FORMAT(1H ,A8)
1    WRITE(2,101)
101  FORMAT(///)
     READ(1,102)N
 102 FORMAT (I0)
     IF (N .LT. 1) STOP
     CALL MAINLOOP
 40  CONTINUE
      STOP
     END


     SUBROUTINE MAINLOOP
     INTEGER X
     DIMENSION  D(20,20), X(20)
     COMMON N,D,X
     WRITE(2,102)N
     READ (1,101) ((D(I,J), J=1,N) I=1,N)
101  FORMAT(10000F0.0)
     CALL ASSIGNMENT
     WRITE(2,102)X
 102 FORMAT(1H ,I5)
     WRITE(2,103)
103  FORMAT(/)
     RETURN
     END
```

P.NO. 7) Contd.

```
      SUBROUTINE ASSIGNMENT
      INTEGER C,CB,R,Y,FLAG,CBL,CL,CLO,RL,RS,SW,X
      REAL MIN
      DIMENSION D(20,20),X(20),C(20),CB(20),LAMBDA(20),MU(20),R(20),
     1Y(20),A(20,20)
      COMMON N,D,X
      DO 5 I=1,N
      MIN=D(I,1)
      DO 1 J=2,N
 1    IF (D(I,J) .LT. MIN) MIN=D(I,J)
      DO 2 J=1,N
 2    A(I,J)=D(I,J)-MIN
 5    CONTINUE
      DO 10 J=1,N
      MIN=A(1,J)
      DO 6 I=2,N
 6    IF (A(I,J) .LT. MIN) MIN=A(I,J)
      DO 7 I=1,N
 7    A(I,J)=A(I,J)-MIN
 10   CONTINUE
      DO 11 I=1,N
      X(I)=0
      Y(I)=0
 11   CONTINUE
      DO 13 I=1,N
      DO 12 J=1,N
      IF (A(I,J) .NE. 0.0 .OR. X(I) .NE. 0 .OR. Y(J) .NE. 0 ) GO TO 12
      X(I)=J
      Y(J)=I
 12   CONTINUE
 13   CONTINUE
 15   RL=0
      CL=0
      RS=1
      FLAG=N
      DO 16 I=1,N
      MU(I)=0
      LAMBDA(I)=0
      IF (X(I) .NE. 0) GO TO 16
      RL=RL+1
      R(RL)=I
      MU(I)=-1
      FLAG=FLAG-1
 16   CONTINUE
      IF (FLAG .EQ. N) RETURN
 17   I=R(RS)
      RS=RS+1
      DO 18 J=1,N
      IF (A(I,J) .NE. 0.0 .OR. LAMBDA(J) .NE. 0) GO TO 18
      LAMBDA(J)=I
      CL=CL+1
      C(CL)=J
      IF (Y(J) .EQ. 0) GO TO 30
      RL=RL+1
      R(RL)=Y(J)
      MU(Y(J))=I

 18   CONTINUE
```

P.NO. 7) Contd.

```
      IF (RS .LE. RL) GO TO 17
      SW=1
      CLO=CL
      CBL=0
      DO 19 J=1,N
      IF (LAMBDA(J) .NE. 0) GO TO 19
      CBL=CBL+1
      CB(CBL)=J
 19   CONTINUE
      MIN=A(R(1),CB(1))
      DO 21 K=1,RL
      DO 20 L=1,CBL
      MIN=MINO(MIN,A(R(K),CB(L)))
 20   CONTINUE
 21   CONTINUE
      DO 27 I=1,N
      IF (MU(I) .NE. 0) GO TO 23
      DO 22 L=1,CLO
 22   A(I,C(L))=A(I,C(L))+MIN
      GO TO 27
 23   DO 26 L=1,CBL
      A(I,CB(L))=A(I,CB(L))-MIN
      GO TO (24,26,28,30)SW
 24   IF (A(I,CB(L)) .NE. 0 .OR. LAMBDA(CB(L)) .NE. 0) GO TO 26
      LAMBDA(CB(L))=I
      IF (Y(CB(L)) .NE. 0) GO TO 25
      J=CB(L)
      SW=2
      GO TO 26
 25   CL=CL+1
      C(CL)=CB(L)
      RL=RL+1
 26   CONTINUE
 27   CONTINUE
      GO TO (28,30)SW
 28   IF (CLO .EQ. CL) GO TO 17
      L=CLO+1
      DO 29 I=L,CL
 29   MU(Y(C(I)))=C(I)
      GO TO 17
 30   I=LAMBDA(J)
      Y(J)=I
      IF (X(I) .NE. 0) GO TO 32
      X(I)=J
      DO 31 I=1,N
 31   IF (X(I) .EQ. 0) GO TO 15
      RETURN
 32   K=J
      J=X(I)
      X(I)=K
      GO TO 30
      RETURN
      END
      FINISH
****
```

PROGRAM NO. 8

```
'BEGIN'    'COMMENT'  SUM OF SQUARES AND PRODUCTS
  'REAL' A,B,C,D,E,Y,X1,X2,X3,X4, SIGMAX1X2, SIGMAX1X3, SIGMAX1X4,
                                  SIGMAX2X3, SIGMAX2X4, SIGMAX3X4,
                                  SIGMAYX1,SIGMAYX2,SIGMAYX3,
                                  SIGMAYX4;
  'INTEGER' I,N;
     SELECTOUTPUT(0);
     A:=B:=C:=D:=E:=SIGMAX1X2:=SIGMAX1X3:=SIGMAX1X4:=SIGMAX2X3
         :=SIGMAX2X4:=SIGMAX3X4:=SIGMAYX1 :=SIGMAYX2:=SIGMAYX3
                                :=SIGMAYX4:=0;
     N:=READ;
   'FOR' I:= 1 'STEP' 1 'UNTIL' N 'DO'
       'BEGIN'
                 Y:=READ;
                 X1:=READ;
                 X2:=READ;
                 X3:=READ;
                 X4:=READ;
                A:= A+Y*Y;
                B:= B+X1*X1;
                C:= C+X2*X2;
                D:= D+X3*X3;
                E:= E+X4*X4;
                SIGMA X1X2 := SIGMAX1X2 + X1*X2;
                SIGMA X1X3 := SIGMAX1X3 + X1*X3;
                SIGMA X1X4 := SIGMAX1X4 + X1*X4;
                SIGMA X2X3 := SIGMAX2X3 + X2*X3;
                SIGMA X2X4 := SIGMAX2X4 + X2*X4;
                SIGMA X3X4 := SIGMAX3X4 + X3*X4;
                SIGMAYX1:=SIGMAYX1+Y*X1;
                SIGMAYX2:=SIGMAYX2+Y*X2;
                SIGMAYX3:=SIGMAYX3+Y*X3;
                SIGMAYX4:=SIGMAYX4+Y*X4;
       'END';

        PRINT (A,4,4);
        WRITETEXT('('=SIGMAT2')');
        NEWLINE(1);
        PRINT(B,4,4);
        WRITETEXT('('=SIGMA X12')');
        NEWLINE(1);
        PRINT(C,4,4);
        WRITETEXT('('=SIGMAX2 2')');
        NEWLINE(1);
        PRINT(D,4,4);
        WRITETEXT('('=SIGMAX3 2')');
        NEWLINE(1);
        PRINT(E,4,4);
        WRITETEXT('('=SIGMAX4 2')');
```

P.NO. 8) Contd.

```
            NEWLINE(1);
            PRINT(SIGMAX1X2,4,4);
            WRITETEXT('('=SIGMAX1X2')');
            NEWLINE(1);
            PRINT(SIGMAX1X3,4,4);
            WRITETEXT('('=SIGMAX1X3')');
            NEWLINE(1);
            PRINT(SIGMAX1X4,4,4);
            WRITETEXT('('=SIGMAX1X4')');
            NEWLINE(1);
            PRINT(SIGMAX2X3,4,4);
            WRITETEXT('('=SIGMAX2X3')');
            NEWLINE(1);
            PRINT(SIGMAX2X4,4,4);
            WRITETEXT('('=SIGMAX2X4')');
            NEWLINE(1);
            PRINT(SIGMAX3X4,4,4);
            WRITETEXT('('=SIGMAX3X4')');
            NEWLINE(1);
            PRINT(SIGMAYX1,4,4);
            WRITETEXT('('=SIGMAYX1')');
            NEWLINE(1);
            PRINT(SIGMAYX2,4,4);
            WRITETEXT('('=SIGMAYX2')');
            NEWLINE(1);
            PRINT(SIGMAYX3,4,4);
            WRITETEXT('('=SIGMAYX3')');
            NEWLINE(1);
            PRINT(SIGMAYX4,4,4);
            WRITETEXT('('=SIGMAYX4')');
'END';

****
```

PROGRAM NO. 9

```
'BEGIN'      'COMMENT' INVERT PROGRAM;
  'INTEGER' N,I,J;
  'PROCEDURE'  INVERT(A,N,FAIL);
  'VALUE' N;
  'ARRAY'A;
     'INTEGER' N;
  'LABEL' FAIL;
    'BEGIN'
       'REAL' BIGAJJ,TRUE;
       'INTEGER' I,J,K;
      'REAL' 'ARRAY' P,Q[1:N];
      'BOOLEAN''ARRAY' R[1:N];
      'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'  R[I]:= 'TRUE';
    GRAND LOOP:
      'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
      'BEGIN'
     SEARCH  FOR  PIVOT:
           BIGAJJ:=0;
          'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
          'BEGIN'
          'IF'  R[J] 'AND' ABS(A[J,J]) > BIGAJJ 'THEN'
             'BEGIN'
                BIGAJJ := ABS(A[J,J]);
                K := J
             'END';
          'END';
          'IF' BIGAJJ =0 'THEN' 'GOTO' FAIL;
    PREPARATION OF  ELIMINATION  STEP I:
           R[K] := 'FALSE';
           Q[K] := 1/A[K,K];
           P[K] := 1;
           A[K,K] := 0;
          'FOR' J:=1 'STEP' 1 'UNTIL' K-1 'DO'
          'BEGIN'
             P[J] := A[J,K];
             Q[J] := ('IF' R[J] 'THEN' -A[J,K] 'ELSE' A[J,K]) *
                     Q[K];
             A[J,K] := 0
          'END';
          'FOR' J:= K+1 'STEP' 1 'UNTIL' N 'DO'
          'BEGIN'
             P[J] := 'IF' R[J] 'THEN' A[K,J] 'ELSE' -A[K,J];
             Q[J] :=  -A[K,J] * Q[K];
             A[K,J] := 0
          'END';
    ELIMINATION PROPER:
          'FOR' J:= 1 'STEP' 1 'UNTIL' N 'DO'
          'FOR' K:= J 'STEP' 1 'UNTIL' N 'DO'
                A[J,K] := A[J,K] + P[J] * Q[K]
      'END'  GRAND LOOP
    'END' INVERT(A,N,FAIL);
```

P.NO.9) Contd.

```
        N:=READ;
         'BEGIN' 'ARRAY' A[1:N ,1:N];
             'FOR' I:=1 'STEP' 1 'UNTIL' N  'DO'
             'FOR' J:=1 'STEP' 1 'UNTIL' N  'DO'
              A[I,J] := READ;
              INVERT(A,N,FAIL);
              'FOR' I:=1 'STEP' 1 'UNTIL' N-1  'DO'
              'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
               NEWLINE (2);
              'FOR' I:=1 'STEP' 1 'UNTIL' N 'DO'
              'FOR' J:=1 'STEP' 1 'UNTIL' N 'DO'
                  'BEGIN'
                     PRINT(A[I,J],2,4);
                  'END';
                   NEWLINE(1);
              'GOTO'  L1;
           FAIL:    WRITETEXT ('('SYSTEM INSOLUBLE')');
L1:     'END';
'END';
```

PROGRAM NO. 10

```
'BEGIN'   'COMMENT'      MEAN;
   'INTEGER' I,N;
       SELECT OUTPUT(0);
     NEWLINE(10);
      N:=READ;
      'BEGIN'
          'REAL' SUM, MEAN;
          'ARRAY' A[1:N];
L1 :       SUM:=0;
          'FOR' I:= 1 'STEP' 1 'UNTIL' N 'DO'
               'BEGIN'
                    A[I]:= READ;
                    SUM := SUM + A[I];
               'END';
           MEAN := SUM/N;
           PRINT (MEAN,2,3);
           NEWLINE(2);
         'GOTO' L1;
         'END';
'END';
****
```

P.I.A. TIME TABLE

WESTERN SECTOR

EAST BOUND FLIGHTS

ALL LOCAL TIMES ARE TAKEN

TIME FROM GMT			MON	TUE	WED		THU	FRI		SAT		SUN	
			PK 701	PK 703	PK 705	PK 707	PK 709	PK 711	PK 713	PK 715	PK 717	PK 719	PK 72
+5	KARACHI	dep	7-30	7-30	00-45	7-30	7-30	00-45	7-30	00-45	7-30	00-45	7-30
+3 1/2	TEHERAN	arr		8-45					8-45				8-45
		dep		9-50					9-50				9-50
+3	DHAHRAN	arr	7-55			7-55							
		dep	8-35			8-35							
+3	KUWAIT	arr					9-25						
		dep					10-15						
+3	BAGHDAD	arr			2-05					2-05			
		dep			3-00					3-00			
+3	CAIRO	arr						3-55				3-55	
		dep						4-35				4-35	
+2	BEIRUT	arr	9-55	10-40		9-00			10-15				10-15
		dep	10-40	11-55		9-45			11-05				11-05
+2	ISTANBUL	arr			4-25		12-25		12-40	4-25			12-40
		dep			5-25		13-20		13-35	5-25			13-35
+2	ROME	arr						6-45				6-45	
		dep						7-25				7-25	
+1	GENEVA	arr							15-35				15-35
		dep							16-15				16-15
+3	MOSCOW	arr									11-35		
		dep									12-35		
+1	PARIS	arr			7-55			8-10		7-55		8-10	
		dep			8-40			8-55		8-40		8-55	
+1	FRANKFURT	arr	13-40	14-55		12-45	15-15				13-45		
		dep	14-30	15-45		13-40	16-05				14-35		
+1	LONDON	arr	15-50	17-05	9-40	15-00	17-25	9-55	17-35	9-40	15-55	9-55	17-35

WEST BOUND FLIGHTS ALL LOCAL TIMES ARE TAKEN

TIME FROM GMT		MON	TUE	WED		THU	FRI		SAT		SUN	
		PK 722	PK 702	PK 704	PK 706	PK 708	PK 710	PK 712	PK 714	PK 716	PK 718	PK 720
+1	LONDON dep	12-45	12-45	9-00	12-45	12-45	9-00	12-45	9-00	12-45	9-00	12-45
+1	FRANKFURT arr		14-05			14-05		14-05	10-20			14-05
	dep		14-45			14-45		14-45	11-10			14-05
+1	PARIS arr	13-45		10-00			10-00			13-45		
	dep	14-30		10-45			10-45			14-30		
+3	MOSCOW arr											20-00
	dep											21-00
+1	GENVA arr				14-05						10-25	
	dep				14-45						11-05	
+2	ROME arr			13-25			13-25					
	ded			14-10			14-10					
+2	ISTANBUL arr	18-40			18-30			18-40		18-40	14-50	
	dep	19-45			19-20			19-35		19-35	15-35	
+2	BEIRUT arr		19-30		20-50	19-40			15-55		17-05	
	dep		20-20		21-40	20-30			16-45		17-55	
+3	CAIRO arr			17-55			17-55					
	dep			18-45			18-45					
+3	BAGHDAD arr	22-55								22-55		
	dep	23-35								23-35		
+3	KUWAIT arr							23-30				
	dep							00-20				
+3	DHAHRAN arr		23-35					1-10				
	dep		00-15					1-50				
+3 1/2	TEHERAN arr				1-15	00-05					21-20	
	dep				1-55	00-45					22-00	
+5	KARACHI arr	TUE 4-50	WED 4-45	THU 1-30	5-55	4-45	SAT 1-30	6-10	23-50	SUN 4-50	MON 2-00	MON 4-35

Persion Gulf - Karachi Sector

TIME FROM GMT			WED-SUN	THU	MON
			PK 745	PK 207	PK 209
+5	KARACHI	dep	9-15	9-15	9-15
+3	JEDDAH	arr	11-15		
		dep			
+4	DUBAI	arr		10-00	
		dep		10-40	
+4	DOHA	arr		11-25	
		dep		12-05	
+4	BAHRAIN	arr		12-35	10-40
		dep			11-20
+3	KUWAIT	arr			11-15

TIME FROM GMT			WED- SUN	THURSDAY	MONDAY
			PK 744	PK 206	PK 208
+3	KUWAIT	dep			12-00
+4	DOHA	arr			
		dep		12-05	
+4	BAHRAIN	arr		12-35	13-50
		dep		13-25	14-30
+4	DUBAI	arr		14-20	
		dep		15-00	
+3	JEDDAH	arr			
		dep	12-05		
+5	KARACHI	arr	17-45	17-45	17-45

Dacca = Bongkok - China - Sector

TIME FROM GMT			THURSDAY		SATURDAY	
			PK 750	PK 706-A	PK 752	PK 712-A
+6	DACCA	dep	8-30	13-30	8-30	13-30
+7	BONGKOK	arr		16-45		16-45
		dep				
+8	CANTON	arr	13-45			
		dep	14-45			
+8	SHNGHAI	arr	16-30		14-45	

TIME FROM GMT			THURSDAY		SATURDAY	
			PK 711-A	PK 751	PK 719-A	PK 753
+8	SHANGHAI	dep		18-00	-	16-20
+8	CANTON	arr				18-10
		dep				19-10
+7	BONGKOK	arr				
		dep	18-00		18-00	
+6	DACCA	arr	19-45	20-25	19-15	20-25

Karachi - Dacca Sector

TIME FROM GMT		MONDAY			TUESDAY			WEDNESDAY			THURSDAY		FRIDAY			SATURDAY			SUNDAY		
		PK 724	PK 730	PK 738	PK 724	PK 730	PK 736	PK 724	PK 730	PK 738	PK 704	PK 706	PK 724	PK 730	PK 736	PK 710	PK 712	PK 736	PK 724	PK 730	PK 736
+5	KARACHI dep	6-30	8-30	18-00	6-30	8-30	11-15	6-3[illegible]	8-30	18-00	3-30	8-30	6-30	8-30	11-15	3-30	8-30	11-15	6-30	8-30	11-15
+6	DACCA arr	10-40	12-40	22-10	10-40	12-40	15-25	10-40	12-40	22-10	7-40	12-40	10-40	12-40	15-25	7-40	12-40	15-25	10-40	12-40	15-25

TIME FROM GMT		MONDAY			TUESDAY			WEDNESDAY			THURSDAY		FRIDAY			SATURDAY			SUNDAY		
		PK 731	PK 723	PK 737	PK 731	PK 735	PK 723	PK 731	PK723	PK 737	PK 735	PK 711	PK 731	PK 735	PK 723	PK 735	PK 735	PK 719	PK 731	PK 735	PK 723
+6	DACCA dep	13-45	19-15	23-10	13-45	16-30	19-15	13-45	19-15	23-10	20-10	21-15	13-45	16-30	19-15	16-30	20-10	21-15	13-45	16-30	19-15
+5	KARACHI arr	16-00	21-30	TUE 1-25	16-00	18-45	21-30	16-00	21-30	THU 1-25	22-25	23-30	16-00	18-45	21-30	18-45	22-25	23-20	16-00	18-45	21-30

DACCA - LAHORE SECTOR

TIME FROM GMT		MON	TUE	WED	FRI	SUN
		PK 725	PK 725	PK 725	PK 725	PK 725
+6	DACCA dep	11-50	11-50	11-50	11-50	11-50
+5	LAHORE arr	13-25	13-25	13-25	13-25	13-25

TIME FROM GMT		MON	TUE	WED	FRI	SUN
		PK 726	PK 726	PK 726	PK 726	PK 726
+5	LAHORE dep	14-30	14-30	14-30	14-30	14-30
+6	DACCA arr	18-00	18-00	18-00	18-00	18-00