COPY 1 of 1 COPIES

AD603478

A HEURISTIC APPROACH TO PILOT SCHEDULING

94 p $3.00 hc
$0.75 mf

by

Robert Anthony Green

Captain, United States Air Force

B.S., United States Naval Academy, 1958

Submitted to the Graduate Faculty of the Schools of Engineering and Mines in partial fulfillment of the requirements for the degree of

Master of Science

in

Industrial Engineering

University of Pittsburgh

1964

ACKNOWLEDGEMENTS

The author feels indebted to many people for their encouragement in writing this thesis. The guidance offered by Dr. Albert G. Holzman and Mr. Robert E. Harding of the Engineering Department, University of Pittsburgh proved valuable indeed.

For the active interest and assistance given me by Mr. Henry R. Rogers, scheduler at the Greater Pittsburgh Airport, a special thank you.

The author proffers his thanks to these three men and those others who shall be nameless here but know how valued was their assistance.

The computer time for this research was made available to the author under Grant Number 11309 from the National Science Foundation.

TABLE OF CONTENTS

I. INTRODUCTION

Because of the availability of large computers in the United States Air Force, the repetitive nature of scheduling, and the time, effort and difficulties involved in scheduling by hand, it is felt that a great need for computer scheduling exists. Much research is being applied to the problem of school scheduling at the University of Pittsburgh. The author believes he can make a contribution in the area of the computer scheduling of aircrew personnel. Being a rated pilot in the United States Air Force as well as a student at the University of Pittsburgh, he has chosen to apply his efforts where he believes both organizations will derive mutual benefit. The problem of aircrew scheduling is such an area, he believes. He has aimed to advance the research in computer scheduling and yet slant it toward an Air Force application.

It seems apropos at this time to define a heuristic approach. The connotation as used herein is, a systematized set of rules, without a mathematical foundation, which enables the computer to derive a feasible schedule. These rules approximate the decisions normally made by the manual scheduler.

The thesis presented here is technique, rather than problem oriented. For this reason some of the statements may suggest areas that seem to require further explanation. In many of these areas the explanation has intentionally been omitted when it offers no real contribution to the technique explained or the specific problem considered.

For a more knowledgeable background of the problem, beyond that presented in the thesis, the author suggests that the reader familiarize himself with the related Air Force policies and procedures.

II. STATEMENT OF THE PROBLEM

A large number of pilots in the Air Force are being utilized today in fields not involving primary aircrew positions. While in these fields of endeavor pilots are required by the Air Force to maintain flying proficiency. Most Air Force bases maintain personnel and aircraft specifically to fulfill this requirement and much time is expended in the scheduling of these attached pilots. The aircraft and pilot availabilities must be considered, of course, in this scheduling process. Air Force restrictions and local flight policies add to the complexity of the problem. The pilot scheduling problem is further confounded by errors introduced by the human scheduler. It is the purpose of this thesis to develop a technique, employing a computer, to produce a feasible flight schedule.

The specific problem under consideration to illustrate the technique is the scheduling of Air Force pilots attending the University of Pittsburgh. These pilots are attached to the 911th Troop Carrier Group at the Greater Pittsburgh Airport for flying. The aircraft provided is the C-47 (DC-3, Douglas "Skytrain"). The computer program is written in the Fortran language for the IBM 7090 computer system utilizing the Michigan Executive System.

III. RESTRICTIONS

The major restrictions of this scheduling problem and their origins are outlined in this section of the thesis. Those which come from Air Force regulations will be identified.

A. Number of Pilots Assigned to Each Period

This restriction stems from the type aircraft flown, Air Force regulations which define when and how a rated pilot can be accredited with flying time and the utilization of <u>one</u> aircraft during each flight period. In the specific problem under consideration the aircraft is a C-47. According to the <u>Flight Handbook</u>[1]*, the minimum crew for a flight is a pilot and a co-pilot. In special cases, however, more than two pilots may be accredited with flying time on a single flight. At Greater Pittsburgh Airport, the average pilot does not fit into these special categories.[2] Because of the availability of rated pilots in the scheduling branch of the operations section of an Air Base, this restriction will be relaxed to allow for an occasional period when a single pilot may be scheduled by the computer. The other pilots required for these periods will be supplied by the scheduling branch.

*Parenthetical references placed superior to the line of text refer to the bibliography.

B. Total Number of Flights During Scheduling Interval

The basic information for deriving this number is found in an Air Force Manual.[2] The value of this parameter will be determined by the scheduler and used as an input to the computer program. Since the University of Pittsburgh operates under a trimester system, each pilot knows his flying availability for a four month period. This is the basis of the four month scheduling interval.

The annual flying requirement for attached pilots is 100 to 110 hours. The trimester schedule and the standard Air Force four-hour flight period for multi-engine aircraft determine that on the average eight or nine flights per trimester satisfy this annual requirement. This number establishes a guide for the scheduling officer. The program allows for variation in this parameter, making it flexible for general situations.

C. The Minimum Number of Flights Per Month

The Air Force Manual setting flight requirements[2] does not specify any required minimum number of monthly flights. This restriction stems rather, from the fact that flight pay is a function of monthly flights. To qualify for flight pay for a month, a pilot must fly at least four hours during that calendar month. While this requirement (a minimum number of flights per month) is not a regulation, it is a practical restriction and must be included in any operational schedule.

D. The Minimum Number of Night Flights

The pertinent manual[2] stipulates the annual hours of night flying time required for each rated pilot. This time requirement is then changed to the number of flights by the method explained in restriction B. This constraint also introduces other problems which must be considered. In most cases, night flights should be scheduled in weekly blocks because of limitations imposed by maintenance work scheduling and man-hour allocation. When the scheduler determines that a period is available for flying, he has been assured by Maintenance that an aircraft will be supplied. Hence, once a night flight is made available it should be flown.

E. Integer Solutions

Inherent in any scheduling or assignment problem , is the need for integer solutions. In the pilot scheduling problem the integer solution restriction must be considered. When a pilot is assigned to fly during any given period, he must be considered as an entity. The pilot must be either scheduled or not scheduled; he cannot be partially scheduled. The concept of completeness or entirety must also be followed in the utilization of a flying period. The flying period, like the pilot, must be considered as one unit and either scheduled or not scheduled. Thus, a definite restriction of the pilot scheduling problem is the need for solutions of integer values so as to eliminate any confusion which may arise.

F. Maximum Number of Daily Flights

Air Force regulations allow a crew-day which would permit a pilot to fly as many as three, four-hour flights. This, while possible, is completely impractical. Local policies dictate that pilots be limited to one flight per day. Should a pilot be scheduled for a night flight, he would not be required to fly the next morning, as illustrated in restriction G, below.

G. Minimum Time Between Flights

The flying of an aircraft is a very exacting operation. For this reason, crew rest is of major importance. The schedule generator must be one which considers these crew rest requirements and spaces flying assignments in order that adequate time is allotted for rest between flights.

IV. GENERAL EXPLANATION OF THE HEURISTIC APPROACH

The method of scheduling selected is one that would basically fulfill the needs for simple input, rapid schedule generation, and readily usable output. These were the basic criteria considered in the formulation of the program. The program was also constructed to satisfy the general restrictions which have been previously mentioned.

It should be noted that the technique is operational as evidenced by the fact that the flight schedule utilized for July 1964 by Mr. Henry R. Rogers, the Scheduling Officer for the 911th Troop Carrier Group, is the actual output of this program.

A major assumption of the method under discussion is that the number of periods made available by the scheduler is greater than the number of periods required for utilization in the schedule. This assumption seems intuitively true and can be verified as mathematically sound as shown below.

In the two sets of test data used the number of periods available were 300 and 231, respectively. Superficially the periods available seem more than ample, especially since two pilots are assigned to fly together. These figures are misleading, however. Many periods cannot be utilized for one reason or another; e.g., the day may be a holiday, the scheduler may prefer not to utilize certain periods because of unavailability of maintenance support, etc., previous commitments may leave the aircraft unavailable, etc. Again, using the available test data, there were 191 and 155 usable periods, respectively. This still seems ample since the periods required by the same test data were 146 and 128. Thus, the assumption was made that a sufficient number of periods were available to satisfy the requirements.

A second major assumption was that for every pilot, his availability and the periods made available by the scheduler were compatible. In other words, it was possible to schedule every pilot and fulfill his requirements under the availability restriction data he gave the scheduler. This seemed to be a valid assumption since it has previously been satisfied by manual scheduling methods although, at times, with great difficulty.

The third major assumption made was that the computer could be utilized to schedule pilots by fulfilling the flight requirements within the restrictions imposed by the problem. To accomplish this, a method was devised using a utility function as the primary criterion in scheduling. By use of this method, feasible schedules were not found in all cases. However, the output of the program utilizing this method is extremely useful in that it did accomplish better than 90 per cent of the scheduling, the remainder to be completed manually.

Utility as used in this thesis is not meant to have the same connotation as when used in an economic application. As used here, "utility" implies preference and/or availability, depending on the specific values used. The measures are provided subjectively by the people involved. In the explanation of the input and formulation of the initial assignment matrix (to be explained), "utility" refers to the preferences and availabilities of the periods as stipulated by the scheduler and the individual pilots. In the generation of the pilot schedule, the word utility represents a combination of the two types just discussed.

The method devised can be divided into three phases: preparatory, schedule generation and implementation.

A. Preparatory Phase

The preparatory phase is performed manually by the scheduling officer, and includes the work preliminary to the two succeeding stages. The general flight requirements are considered, total scheduling interval reviewed, pilots' availabilities determined, and any special cases noted. This area of preliminary investigation in further discussed in Section V (Input) and also in Appendices A (User's Memo) and D (Input).

B. Schedule Generation Phase

The schedule generation phase is accomplished by the computer program developed. This program uses a heuristic approach to schedule generation which is accomplished in three steps,as discussed below.

1. Formulation of the Assignment Matrix

The assignment matrix is a two-dimensional array with the elements representing utility values. This concept of utilities is expanded and discussed in detail in Section V (Input). Briefly, the scheduler assigns a utility value to the flight periods available for each day within the scheduling interval. The pilot assigns a utility value to the flight periods each day for a week, his weekly availability. The program then combines, by multiplication (because of the values used), the utilities assigned by the scheduler and the pilots. In this manner there is a value for each pilot for every period. This is maintained in computer memory as a two-dimensional array with each element a utility value, representing a specific pilot and period. This matrix is constantly up-dated as scheduling progresses. The initial assignment matrix is printed out in the first section of the output for information and verification.

2. Assignment

Assignments are made based upon a method of maximum utilities. The algorithm used follows. The program cycles through the pilots each in turn until the scheduling is completed. When a pilot is considered, he is assigned to the period of his first maximum utility. A system of checks exists within the program which insures the feasibility of this assignment. The assignment matrix and counters are up-dated and the assignment stored. The computer then searches for the maximum utility of the remaining pilots for that period in question, for the companion pilot. The companion pilot is then assigned, the matrix and counter up-dated and the assignment stored. The computer then repeats this scheduling cycle. This step is explained in detail in Section VI (Scheduling).

3. Output

In the assignment step of the program, the individual assignments are recorded as single numbers and the total schedule as an array. The translation of this array into two different usable forms is undertaken in the output step. This array is manipulated and translated into usable output as shown in Appendix E (Output).

C. Implementation Phase

The last phase of this approach to pilot scheduling is again manual. The scheduler must review the output of the program for exceptional cases. These will be listed for him as names of pilots not completely scheduled by the program, or as a list of zeros representing a flight where only one pilot was scheduled. In the tests made, over 90 per cent of the scheduling was completed by the machine and the only restriction causing most of the exceptional cases was the total number of flights required per pilot per scheduling interval. This does leave some scheduling to be accomplished manually. The responsibility for notification of scheduled pilots, the last step in the scheduling process, remains with the scheduling officer.

V. INPUT

The program input is best considered in four parts; a) general information; b) daily information; c) pilot information, and d) and specific changes to the general information. The format and further explanation of the program input is presented in Appendix A (User's Guide).

A. General Information

The general information is presented on two cards with the first card containing 1) the number of pilots to be scheduled, 2) the total number of flights per pilot for the scheduling interval, 3) the maximum number of flights per pilot that can be scheduled each month, and 4) the maximum number of night flights to be scheduled for each pilot over the scheduling interval. The 2) total, 3) monthly and 4) night restrictions, as given in the input are to represent the norm. Pilots with special requirements will be treated as another part of the input.

By careful consideration of the number of 2) total and 3) monthly pilot flights stipulated, the minimum monthly flight restriction may be satisfied. The scheduler <u>must</u> insure that the total number of flights allotted per pilot, for the scheduling interval, is greater than the sum of the flights allotted per pilot per month should the allocation of any one month be eliminated. For example, if a four month interval is being utilized and the scheduler would like no more than three flights to be scheduled in any month, he must insure that the total flights stipulated for the time interval be greater than nine.

The second of the two input cards allocates 15 spaces for identifying information to be printed during output. This information may be dates, code names, etc.

B. Daily Information

The second part of the input is a set of daily cards. Each day of the scheduling interval, with the exception of Sundays, is represented by a separate card. These cards contain the utilities the scheduler has assigned to each of the daily flights. The first of the daily cards *must* represent a Monday with all cards in order of increasing dates. If the scheduling interval does not start on a Monday, blank cards must be inserted. These requirements exist to insure compatability between the pilot and the daily cards. In all cases input data values not punched in the spaces provided will be interpreted as zeros.

On these daily cards, the first set of spaces allotted serves a dual purpose. They are used to designate the first *day* of the month, and the last *card* in this particular set. The months must be noted because *dates* are *not* associated with the *periods* during the assignment phase of this program. At this point, however, groups of consecutive periods *are* associated with the different months to control the monthly flight requirements. The last card is recorded to do away with the need for counting the daily cards and, to serve as a means of internal control.

Next on the daily card the date is recorded, followed by three numbers. These numbers are the utility values assigned by the scheduler to the three flight periods of the day. Air Force policy normally specifies three flight periods per day. These values vary from zero to nine, a scale determined arbitrarily. A value of zero assures that no flight will be scheduled for a stated period; three and four are used to denote day flights which the scheduler prefers to utilize; and all night flights are assigned a value of nine.

C. Pilot Information

The third set of input data is used to supply the pilots' utilities and availabilities. The words utility and availability are almost synonymous as used in this thesis. Utility is used to indicate a preference. Because of the values used, utility also determines availability. There is one card for each pilot with all cards arranged in order of the pilot's degree of availability (minimum availabilities last). The first 15 spaces on each card are available for the pilot's name and other identifying information, which will be used to identify pilot assignments in the program output. Six data groups of three spaces each follow this. These data groups represent the six days of the week and the three spaces, the three flights in each day. The number contained in these spaces represents the pilots' utility for these periods.

The utility values punched on the pilot cards are similar to those explained above for the daily cards. In this case, however, the values range from zero to two. If a zero is assigned or the space left blank, the pilot is considered unavailable during that period. The pilot assigns a value of one to all periods in which he is available with the exception of one day and one night period to which he assigns a value to two to indicate preferred flying times.

D. Specific Changes to General Information

The last set of input cards allows for flexibility in scheduling. While the first set of cards establishes maximum number of flights for the *average* pilot, *this* set provides opportunity for variation and changes for specific individuals.

The first card in this group states the number of pilot change cards which will follow. On the pilot change cards seven items of information are contained; the first being a number indicating the location of the referenced pilot card in the pilot deck. Thus, if the pilot card referenced is the *fifth* card in the pilot deck, the first number on the pilot change card is *five*. The next item on the card is the maximum number of flights per scheduling interval for *this* pilot.

The next four items are the desired maximum number of flights for each month of the four in the scheduling interval. Finally, the last item designates the total number of night flights specified for the scheduling interval. Care must be exercised to insure that values are assigned for every item, with special attention directed to items three through six.

VI. SCHEDULING

A general discussion of the scheduling method is presented in this section. A description of the methods for satisfying the restrictions introduced in Section III, and a verbal presentation of the scheduling method, is tendered. Only the areas of major significance will be covered since a more detailed description is available in Appendices B (Flow Diagrams) and C (Program Listings).

A. Restrictions

1. Number of Pilots Assigned to Each Period

The requirement of two pilots for each flight was one point considered in the formulation of the approach. After a pilot is scheduled for a particular period, the computer attempts to schedule a companion pilot. Whether there is one available or not, the assignment matrix is up-dated and all the pilot utilities for that period are set to zero. When this occurs, the program no longer has any utilities for the period, and no more assignments can be made there. This limits the number of pilots per period to no more than two. If only one is available, this is an exceptional case which the scheduler must handle.

2. Total Number of Flights During the Scheduling Interval

Control of total flights per scheduling interval using the heuristic approach is maintained by a running count of the scheduled flights for each pilot. The running count is tested, after up-dating, against the number assigned to this control parameter. When this count equals the total flights to be flown by a pilot, the pilot is eliminated from further scheduling consideration. This is accomplished by setting all the period utilities for that pilot to zero and setting a pilot index. Before a pilot is considered for an assignment this index is checked. If the pilot is completely scheduled his utilities are not even viewed (thus saving computer time) and the next pilot in line is considered.

3. Maximum Number of Flights Per Month

In controlling the number of monthly flights both a maximum and a minimum number must be considered. The maximum is controlled by a running count for each pilot's monthly schedule. When any month's scheduled flights equal the maximum, as stated in the program input, the pilot is no longer considered for scheduling that month. This control is maintained by a check, for the month, of every flight scheduled. A running count of this is maintained for each pilot. Whenever this count equals the maximum flights allowed by input, the utilities for that pilot in that month are set to zero, thus eliminating him from scheduling consideration for that month.

The method of maintaining the minimum monthly requirement has already been discussed in Section V (Input). Briefly recall that this is accomplished by values of the input parameters used, as indicated.

4. The Minimum Number of Night Flights

The technique employed in measurement and control of the number of night flights is intriguing. The first problem encountered was identification of these night flights. One solution is to have a separate system to solve this aspect of the scheduling problem. However, a unique method was selected which eliminates the necessity of a separate system.

If a night flight requirement for the pilot being assigned exists, the period number is divided by three and tested for a remainder. It is a mathematical truth that given all consecutive integers as a set, only every third integer starting with three can be divided by three without leaving a remainder. Because of the input format, every third period is a night flight. This method of evaluation is feasible and conveniently attainable as both fixed and floating point[1] modes are available on the computer utilized. A running count of the night flights

---

[1] A fixed point number is an integer (whole) number; where as a floating point number may be an integer or may have a fractional part.

scheduled is stored and tested in a manner similar to that used for total flights. When a pilot is completely scheduled, the assignment matrix is up-dated by making his every third utility equal to zero. At the same time, the pilot requirement for night flights is set to zero, thus eliminating any further night flight checks for this pilot.

5. Integer Solutions

The integer solution restriction of this program seems almost superficial. This was not always the case, however. In the method finally developed, the heuristic approach, the inputs are integer variables. Up to and including the assignment stage of the program the only arithmetic operations used are multiplication, addition and subtraction. These conditions, integer input, with no division operations insure that integer values are preserved during the assignment phase.

During the output phase of the program, division is introduced and hence the possibility of non-integer values. To satisfy the integer requirement in this phase, the numbers are maintained as fixed point numbers which are either printed out or used as subscripts. A more detailed explanation is forthcoming in Section VII (Output).

6. Maximum Number of Daily Flights

This requirement exists in order to limit pilots to one flight during the scheduling day. This is accomplished by setting the pilot's utility value to zero for the period preceding and following the assignment. In this manner, the computer can no longer consider these periods for scheduling. This method also insures that when a night flight is scheduled for a pilot, his first period the next morning cannot be assigned The only times this method is ineffective are when a night flight is scheduled and the morning period for that same day remains available; and when a morning flight is scheduled and the evening period of the same day remains available for assignment.

These deficiencies are removed by the high utility values given to the night flights, which eliminate the possibility of a pilot having a morning flight scheduled before an evening flight of the same day. The possibility of a flight in the morning, when a night flight is already scheduled, is eliminated by giving the period two time periods before the night flight a utility of zero for the pilots scheduled for the night flight.

To summarize, any time a period (t) is scheduled, the utility value of periods (t - 1) and (t + 1) are set to zero. If the flight is a night flight, the utility value for period (t - 2) is also set to zero. In this manner, flights per pilot are limited to one a day.

7. Minimum Time Between Flights

As introduced in Section III, this restriction exists to provide adequate crew rest for pilots between flights. The normal time required to satisfy the crew rest requirement is 15 hours.

This restriction is satisfied by limiting flights to one per day, and not permitting a morning flight to follow a flight the previous night. This is accomplished in the process of satisfying the previous restriction (6).

B. Scheduling Procedure

The explanation of the heuristic approach under consideration will be directed here toward the two basic steps; 1) the formation of the initial assignment matrix, which also includes a discussion of the input, and 2) the process up to, but not including, the final manipulation for output of the assignments.

During the explanations, an attempt will be made to avoid unnecessary details, i.e., individual variables. Formats and minor points will also be eliminated from this discussion. If questions arise, the reader is directed to Appendix C (Program Listing).

1. Formation of the Initial Assignment Matrix

In the IBM 7090 computer, under the Michigan Executive System, the locations in memory are loaded with a core constant.[3] For this reason, one of the first problems was that of the removal of this core constant. A subroutine (see Appendix C) was written to zero the location of every dimensioned variable. In many cases this action was not required, but the program debugging was greatly simplified. In addition to this initialization, other variables are initialized or zeroed as required.

The first data card read contains the number of pilots, and the number of total, monthly, and night flights per pilot. These numbers are then assigned as the limits of each pilot's restrictions. As the individual pilots are scheduled, a count of then assigned flights is checked against these limits. Of course, these individual limits could then be changed, if needed, by an additional input. The information card is then read and stored in alphameric notation to be printed in output format when required.

In the next step the daily cards are read into the computer of the daily cards. This information is stored in an array whose indices are the same consecutive periods used in the assignment matrix. The dates and flights on the daily cards are associated with these index numbers. Thus, the dates and flights are available when required by the program. The elements of this array are the utilities of the periods as assigned by the scheduling officer.

As the daily cards are read in, they are checked for a number in the first set of spaces. If a negative number is found there the program then records the period as the first period of the month. In this way the periods are associated with months. To test for the month is any period, one needs only to know if it lies between two values -- the first day of one month and the first day of the following month. If instead of a negative number, a positive number is found, the program views this as the last of the daily cards and no more cards will be read at this time. By utilizing this system, a method of recording the months is available and also the need for counting daily cards is eliminated.

After the daily cards are read and the program up-dated as required, the pilot cards are read and utilized sequentially. As a pilot card is read it is assigned a number which corresponds to its position in the pilot deck and the pilot identification is stored in alphameric form to be recalled as needed.

The utility values read from the pilot card are then multiplied by the utilities from the daily array in groups of 18, and are stored in the row of the assignment matrix corresponding to that pilot. In this way, all utilities for said pilot are the results of a multiplication of the daily and the pilot utility values associated with each period. This step is repeated until all pilots have been considered and the initial assignment matrix completed. The initial assignment matrix is printed out at this time for purposes of information and verification.

The next data card is read. This card indicates the number of pilot change cards which will follow with each pilot change card referring to a single pilot. The cards are read into the computer making it possible for the input parameters from the first card to be revised for individual pilots.

By this time, the assignment matrix has been formed and the pilot restrictions recorded and up-dated. The program is ready to start the second step.

2. Generation of a Feasible Schedule

In the second step, the program cycles through the pilots, considering each in turn, for assignment. As each pilot is considered, he is assigned the first preferred period as indicated by the values in the assignment matrix. A system of checks and balances exists within the program which insures the feasibility of this assignment. The assignment matrix and the counters are upgraded and the assignment stored as a number in an internal array. The computer then searches the utilities of the period assigned for a companion pilot. This is accomplished by searching the assignment matrix to locate the maximum utility value for this period. The pilot for whom this value exists is now assigned as the companion pilot. The matrix and counters are again upgraded and the assignment stored. The computer then repeats this scheduling cycle, until all pilots' utility values have been exhausted. A detailed explanation of how the computer completes this cycle follows.

In selection of the pilot's maximum utility, the first one considered becomes the tentative maximum. The program compares this tentative maximum with all of the remaining utilities in turn. When one is found which is larger than the tentative maximum, this one becomes the new tentative maximum and the period number is stored. The result of this search identifies the first maximum utility for the pilot and the period with which it is associated. This value is then examined to insure that it is greater than zero. If it is equal to zero, this pilot has no more usable periods available, although his flight requirements are still not fulfilled. This fact is noted in the output and this pilot is no longer considered for scheduling.

Once the maximum utility for a pilot is determined, he is assigned to the pertinent period. The program then up-dates the pilot's counters and tests them against the requirement as given in the input. The first counter is for total flights for the scheduling interval and is augmented by one for each assignment made. When this accumulated total equals the required total, as specified by the input, this fact is recorded. This record is used to indicate future revisions necessary to the assignment matrix which consist of setting all this pilot's utility values to zero so that he can be scheduled no longer, and up-dating an index which will eliminate him from scheduling by the computer. This serves to save computer time.

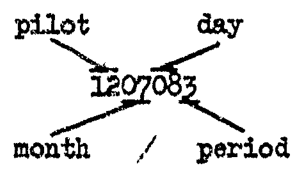
If the accumulated total is less than the requirement specified by the input, the computer then checks to see in which month the period is located. This is accomplished by subtracting the number of the period scheduled from the number assigned to the first period of the second month. The result of this subtraction is then tested. If the number is positive then the period is not in the first month. This test is repeated for the following months until the month containing the period is determined. When the month is located, a value of one is then added to the pilot's pertinent monthly counter. If the counter then equals the maximum number of flights for the month as stipulated by the input, the assignment matrix is revised. This revision is accomplished by setting all the pilot utilities for the month equal to zero. This eliminates the pilot from further assignment in that month.

If the monthly requirements have not been satisfied, the program considers night flights. The computer checks the night flight requirement for the pilot. The night flight requirement can be eliminated in two ways, 1) through the input when the scheduler does not establish a requirement, or 2) by the program. If a pilot's requirement is satisfied or the period tested is not a night flight, the night flight requirement is removed. The justification for the removal of the requi ment stems from the fact that high utilities are given to the night fli periods to insure they will be utilized first. The test for a night flight is accomplished through the computer's ability to use both fixed and floating point numbers. If a night flight requirement exists, the number of the period in question is divided by three. The computer the tests for a remainder by use of fixed and floating point modes. If a remainder exists, this indicates the period is not a night flight peric and no more night flight periods are available for the pilot, the nigh flight requirement is then removed and the assignment stored. If no remainder exists, the period in question is a night flight period and a value of one is added to the pilot's night flight counter. At this point, the utility value of the period two flight periods prior to the one under consideration is set to zero. In this way, the program eliminates the possibility of assigning the pilot a morning flight period o the same day. The value of the night flight counter is then compared with the night flight requirement. If no further night flight requireme exists, the assignment matrix is again revised. This revision is accomplished by setting every third period utility for the pilot to zero and eliminating the requirement. The assignment matrix has no more positive night flight utilities for the pilot and so no more checks for night flights will be made.

The program up-dates the assignment matrix and stores the pertinent information. The assignment matrix is updated by setting the utility of the period scheduled, along with the preceding and following periods, to zero. This action, in conjunction with that previously taken in regard to night flights, limits the pilot to one flight per day. This also insures that the pilot scheduled for a night flight will not be scheduled for a flight the following morning, satisfying the crew rest restriction.

Assignments are stored as seven digit numbers with the first two digits as pilot identifiers and the following five as period identifiers.

Example:



Because of the size of this number and the characteristic of the IBM 7090 computer, storage is in floating point mode.[3]

An attempt is next made to schedule a companion pilot for this period, in a manner similar to testing for the pilot's maximum utility except that here it is testing for the period's maximum utility. Unlike the aforementioned procedure, which takes the first maximum utility for the pilot, the program now takes the last maximum utility for the period. If a maximum utility other than a zero exists, a companion pilot is available and will be scheduled. The computer then sets all pilot utility values for this period to zero. The companion pilot's counters are up-dated and checked, the assignment matrix revised, and the assignment stored, thus completing one cycle.

If no companion pilot is available, a fictitious companion pilot is assigned, thus completing the cycle. These cycles are repeated for all the pilots in turn until all are completely scheduled or only zero values exist in the assignment matrix.

During the assignment stage of the computer scheduling process there are two cases where incomplete scheduling may occur. The first case is when a pilot's turn in the scheduling cycle arrives and he lacl positive utility values. When the pilot to be assigned has a maximum utility equal to zero, his name is printed on a list denoting that his schedule is not complete. At this time, the pilot's index is revised eliminating him from future consideration. Another area where this information can be obtained is from the first schedule display. In this case, the number of individual pilot's scheduled flights will not equal the number of total flights requested by the scheduler.

The second type of exception exists when no companion pilot is available to fly during a period. As explained in the restriction portion of this section, this condition was considered and allowed to exist. This information will be noted on the second assignment display. In this case, the companion pilot assigned to the period is a fictitous one and is denoted by a series of zeros instead of a name.

VII. OUTPUT

An example of the output of this heuristic approach to computer scheduling of attached pilots is shown in Appendix E. It is presented in three parts; 1) the initial and assignment matrix, 2) the pilot problem area, and 3) the pilot and daily completed flight schedules. This discussion will follow the same order presented in Appendix E.

The first part, printing of the initial assignment matrix, was discussed in the explanation of the computer program. It is printed before any scheduling action in undertaken by the computer. The initial assignment matrix presents the utilities of every pilot and the period and date which they represent and can be used as a check on the input. It was designed, however, primarily to accomodate possible changes in the future. If for some reason, i.e., sickness, a change is necessary for a certain flight period, the scheduler has ready reference to the availabilities and utility values for the other pilots for that period.

The second part of the output is a list of the pilots whose requirements could not be completely satisfied by the computer, a list of exceptions. As indicated above, the names on this list are printed as they are encountered in the scheduling phase.

The third part of the output is the print out of the flight schedule and does not commence until the scheduling phase is completed and the schedule is internally available to the computer as a list of assignment numbers.

In this phase the list of assignments is manipulated to achieve the results desired. The preferred output of the schedule is in two forms. The first form is a list of pilots and the dates each is schedu to fly, i.e.,:

ADAMS

5	21	2
5	28	2
6	3	3
6	10	3

The pilot's name and the dates scheduled by the computer are presented in chronological order. This presents a usable display of each pilot's assignments. The second form of display lists the dates in chronologic order followed by the pilots scheduled, i.e.,:

6	10	3	CAREY	ADAMS
6	13	1	FAY	DAVIS

The pilot order has no bearing other than the fact that in the method used, the pilot whose name appears first on the input will appear first on the output.

As explained previously, the assignments are stored in the locations AM(i) i = 1, 2, ..., N as seven digit numbers, i.e.,:

AM (1) = 01 07 08 3. AM (4) = 01 07 10 1.
AM (2) = 08 07 10 1. AM (5) = 08 07 12 2.
AM (3) = 06 07 08 3. AM (6) = 06 07 12 2.

This list is used to explain the manipulation to obtain desired output.

After the headings are printed for the first output form, the assignment list is ordered. The following result is the above list ordered with identifier eliminated for simplicity:

0107083
0107101
0607083
0607122
0807101
0807122

The computer proceeds through this list manipulating each number in turn. A number is divided by 10,000 and the result changed to fixed point mode giving the pilot identification number, i.e., (01). The number is then converted to floating point mode (01.), multiplied by 10,000 and subtracted from the original assignment number, i.e., 0107083 - 0100000 = 07083. The resulting difference represents the date and period of the flight assigned.

By testing the pilot number, all dates scheduled are associated with individual pilots. The date-period-numbers, since they are now small enough to be manipulated, are maintained in fixed point mode and further subdivided into month, day and flight period. When all the pilot's assignments are collected and ordered, they are printed in the following format:

ADAMS	5	21	2
	5	28	2

The second format presents the date and period scheduled in chronological order. To do this the assignment numbers are altered so that date-period group now precedes the pilot identifier. This is accomplished by means of a method similar to that used previously in separating the pilot identifiers. Instead of multiplying the pilot number by 10,000, the date-period group is now multiplied by 100 and added to the pilot number. This transforms the example list:

FROM	TO
0107083	0708301
0107101	0710101
0607083	0708306
0607122	0712206
0807101	0710108
0807122	0712208

The transformed list is then reordered:

0708301
0708306
0710101
0710108
0712206
0712208

The second form of output is obtained from this list by dividi the assignment number into its two basic parts, date-period group and pilot number, and listing the pilots in date-period group order.

VIII. AREAS OF FUTURE RESEARCH

A fruitful area for research may lie in the expansion of the maximum utility method of assignment to include a weekly availability matrix, a method which would consider the number of pilots available for each period. A large number of pilots may be available for certain periods of the week while only a few are available for others. If a pilot were available for both, a method should be devised to further increase his utility for the periods when minimum pilots are available. This improvement would be incorporated in the generation of a feasible schedule and not effect either the formation of the initial assignment matrix or the output of the program. This insertion would eliminate some of the exceptional cases which now appear in the current system.

Another area which might produce improvement is an input for aircraft availability. This input could be of the same form as that of the daily utilities except, in this case, the utility values would represent preference and/or availability of the aircraft usage supplied by the maintenance section, further simplifying the scheduler's preliminary research. Using the same arbitrary scale, these values could be multiplied by the corresponding values from the daily cards and stored in the same array as used in the present program, as explained in Section VI (Scheduling).

At this stage of development of the pilot scheduling problem, pilot availabilities are presented on a weekly basis. No opportunity is given to the pilot to either request specific dates to fly or to eliminate specific dates he knows he will not be available. An input in this area offering the pilots this opportunity would extend the flexibility of the program, however, no method for its application will be offered.

The application of linear programming to this type scheduling problem is another area available for further research. Some preliminary research has already been conducted by the author and is presented in Appendix F, of this thesis.

IX. PROGRAM POTENTIAL

The potential of computer scheduling in this particular area seems almost unlimited. Intuitively, these possibilities fall into two basic areas; 1) man-power saving, and 2) aircraft allocation. This area is subject to speculation. In its present form, the program developed offers substantial potential financial savings in both these areas, as indicated below.

The scheduler now spends approximately one week of every month on schedule formulation. There is also the need to monitor the progress of the schedule. In the area of schedule formulation this program can make real contributions, in the opinion of the author. The program could be used on a four-month basis. Normally, as scheduled by the month, a four-month schedule would take four weeks to prepare. By use of the scheduling method presented here, this is reduced to approximately two days of scheduler effort and less than one minute of computer time. One day's labor could produce the daily period utilities and pilot requirements as needed by the program input. Upon receipt of the pilot availabilities, no more than half-a-day's work would be required to punch and check the data cards. This would leave another half-day to be spent in reviewing the schedule and preparing forms used for pilot notification.

In the area of aircraft allocation, an improvement could be realized by better utilization of the aircraft available. At the time of pilot notification, maintenance personnel can be presented with the aircraft requirement for the four-month scheduling interval. With the advent of the known aircraft requirements for future dates, better preventive maintenance can be practiced and a more stable work force maintained.

This requirement to supply attached flying personnel with the necessary flight time is general throughout the Air Force. The saving realized at an individual base could be increased substantially with general application of the computer scheduling program throughout the Air Force. Because of the number of large computers available and the FORTRAN language, the program devised would offer ready adaptability.

The program, as designed, was to be used for a four-month scheduling interval, six-day scheduling week, and three flights per scheduling day. This need not always be the case. As shown in the example of the specific time interval. Any time interval may be considered, the limiting factor being that the total number of flight periods must not exceed 330. The program was based on a six-day scheduling week, a parameter subject to variation. The important facet to consider is the compatability of the daily utilities and pilot availabilities. If a five-day scheduling week, for example, is used, the daily utilities need only have a blank card to represent the missing day. The pilot cards then need present only the utilities for the days to be utilized. In this case, the easiest solution would be to make every sixth daily card a blank and utilize only the first five days location of the pilot card.

This same basic approach can be used if the number of daily flight periods is different. If a two-period system is used, one of them being a night period, one needs only to eliminate the period from the data. Some care must be exercised, again for compatability, to eliminate the same periods on both daily and pilot cards. If no night flights are utilized, one needs only to omit all reference to night flights in the input.

Other minor variations similar to these mentioned may arise. Further elaboration in this area would serve no purpose; each case must be considered specifically.

X. SUMMARY

The result of this thesis is a practical computer scheduling program for scheduling attached flying personnel to flights. The input shown in Appendix D is actual data as supplied by Mr. Rogers, the scheduler for Air Force personnel attached to the 911th Troop Carrier Group. The output shown in Appendix E is the schedule generated by this program which is being utilized at present for July and August, 1964.

In discussion with Mr. Rogers, it was stated that he believes with more experience in the program's use he would find it even more useful. As the result of its practical application, demonstrated during the trial period (July - August), plans are now being formulat for its continued use after the author departs from Pittsburgh.

As a further aid to the utilization of the scheduling program, a User's Guide was developed. This User's Guide, as contained in Appendix A, will be reduced as a separate document for wider distributio A flow diagram and the program listing of the program is also presented

The use of the daily input information implies that the schedul consider long range objectives and plans by being required to supply utilities for every period during the time interval. This implication could be viewed at the extremes. It would be very desirable, conceivably, to be forced to do long range planning, even without the benefit this scheduling program produces. On the other extreme, a complete la of long range planning, the program is limited to a single month. This would still present a feasible schedule but the time expanded to gather and punch the information would render its use questionable.

The use of a single number to represent an assignment is a novel concept in the area of computer scheduling. By its use, a separate location for the utilities and assignments is feasible in large problems of this type. In the case at hand, the heuristic computer scheduling program, 11,550 internal locations have been dimensioned for the assignment matrix, and only 500 locations to store the assignments made. This offers substantial savings on internal storage locations.

Another advancement has been made in night flight scheduling. By assigning <u>first</u> priority to night flights for a four-month interval, a better utilization of these flights by the computer can reduce the number of night flights required. Again, here exists the necessity for long range planning. The amount of time and effort expanded in long range planning by the scheduler is a major determinant of the benefits reaped by using computer programming.

The range of possible application of this heuristic scheduling computer program is wide enough to make the applications of the program an area worth of investigation. This program offers potential users the opportunity to consider <u>any</u> time interval up to four months. It further offers a range in the number of pilots to be scheduled up to 35. Some variation can also be made in the scheduling day and week, i.e., the number of periods per day, and the number of days per week.

Man-power and time savings is also an important contribution of this computer scheduling program. Savings are realized for both the scheduler and the pilots concerned. The scheduler's time, freed by the use of the program, can be allocated to other phases of his required duties. The pilot saving is accomplished by more effective utilization of his time.

The author feels that his thesis is a significant first step toward mechanizing and standardizing attached flying personnel scheduling procedures at the numerous Air Force installations across the nation.

APPENDIX A

USER'S GUIDE

to the

C.R.T. Computer Scheduling Program

July 1964

TABLE OF CONTENTS

I. Introduction

The purpose of this manual is to acquaint interested personnel and potential users with a computer approach to Combat Readiness Training (C.R.T.) scheduling of air crews. The computer approach used is a heuristic program written in Fortran which generates a feasible schedule. The program was written for use on an IBM 7090 computer operating under the University of Michigan Executive System. The program must be adapted for use under any other monitoring system.

II. Restrictions

The program was written to handle a number of basic restrictions. Other restrictions could be added but would require revision of portions of the program. The basic restrictions considered are listed below as applicable to an individual pilot.

1. No more than two pilots per flight period.
2. Total number of flights for time interval scheduled.
3. Maximum number of flights per month.
4. Minimum number of flights per month.
5. Maximum number of night flights per time interval.
6. Maximum daily flights.
7. Adequate crew rest.

III. Program Requirements

The program requirements considered is a list of information to be used as input. No attempt will be made to suggest methods of obtaining this information.

1. Total number of pilots.
2. Total number of flights per pilot for the time interval of the schedule.
3. Number of flights per month per pilot.
4. Total number of night flights per pilot for time interval.
5. Date and utilities or preference values for the three possible flights in each day of scheduling interval (daily cards).
6. Weekly availability for each pilot -- will be same for every week of the time interval (punched on pilot cards).
7. The total number of pilots with <u>special</u> flight requirements.
8. Each pilot whose requisites differ from those indicated in numbers two through four, and the specific nature of these differences.

Above requirements must be punched on standard 80 column IBM computer cards. The format used to record the required information is explained in Section IV (Input Format).

IV. Input Format

General

The input of the computer program is a collection of cards (dat deck) which presents required information in a form usable by the progr The information is to be punched on a card in columns as stipulated be with all numbers right justified. For example, if two columns are allc ted for a number; e.g., column four and column five; and the number i single digit -- the right column (five) will be used.

The utility value assigned to both the daily cards and the pilo cards are subjective values determined by the scheduler. If the utili value is a zero the flight period is not considered for assignment. Th larger the utility value allotted the greater the chance for assignment

The recommended utility values for the daily cards are:

- 0 - period not to be assigned
- 1 to 4 - values for daily flights
- 9 - value for night flights

The recommended utility values for the pilot cards are:

- 0 - pilot not available
- 1 - pilot available
- 2 - assigned for one day and one night period for which pilot is available to indicate priority

IV. Input Format

Data Deck

1. First Card

Column	4-5	Number of pilots.
	9-10	Total number of flights per pilot for time interval scheduled.
	14-15	Maximum number of flights per month per pilot.
	19-20	Maximum number of night flights per pilot for time interval

2. Second Card

Column	1-25	Any information (dates) - This will be printed on output.

3. Daily Cards

One card for every day in order, Monday through Saturday (six per week). Set must start with a Monday card (when not practical substitute blank).

Column	4-5	Punch a -1 on first day of month (except first month which remains blank.) Punch a +1 on last day to be scheduled (last daily card). All other times leave blank.
	6-8	Number of the month - numerical position one through twelve from calendar.
	9-11	Day of the month - date.
	12-14	Utility number (0-5) for first flight period of day.

15-17 Utility number (0-5) for second flight period of day.

18-20 Utility number (0-9) for third flight period of day. Night flights

4. Pilot Cards

One card for every pilot to be scheduled. Must have same number of cards as the total number of pilots indicated on card one. For maximum efficiency place cards in decreasing order of pilot availability.

Column 1-15 Pilot designation (will appear in output).

16-18 Pilot utility (0-2) for the three periods on Monday.

20-22 Pilot utility (0-2) for the three periods on Tuesday.

24-26 Pilot utility (0-2) for the three periods on Wednesday.

28-30 Pilot utility (0-2) for the three periods on Thursday.

32-34 Pilot utility (0-2) for the three periods on Friday.

36-38 Pilot utility (0-2) for the three periods on Saturday.

5. Change Card

Column 1-5 The number of special pilot cards to follow. If no special pilot cards follow - leave blank.

6. Pilot Change Cards

One card for each pilot whose requisites differ from those indicated on card one. (See following page for columns.)

6. Pilot Change Cards (cont.)

Column 4-5 The pilot's number - his numerical position in the Pilot Cards.

9-10 Total number of flights for the time interval.

14-15 Maximum number of flights to be scheduled for first month.

19-20 Maximum number of flights to be scheduled for second month.

24-25 Maximum number of flights to be scheduled for third month.

29-30 Maximum number of flights to be scheduled for fourth month.

34-35 Total number of night flights to be scheduled.

V. Check List

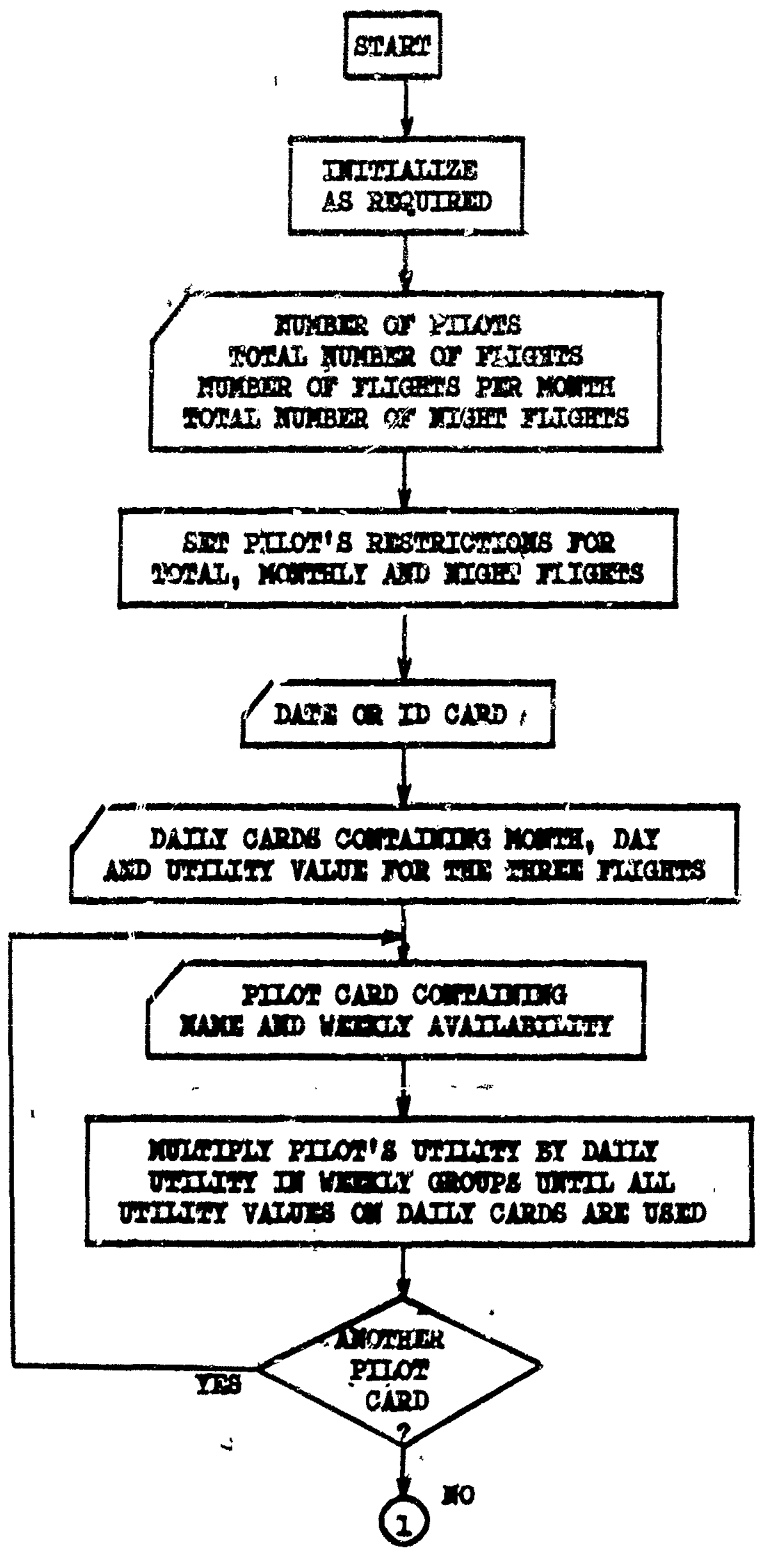
The following check list should be used after the input data is punched on cards. It indicates errors which may be overlooked when compiling the data cards.

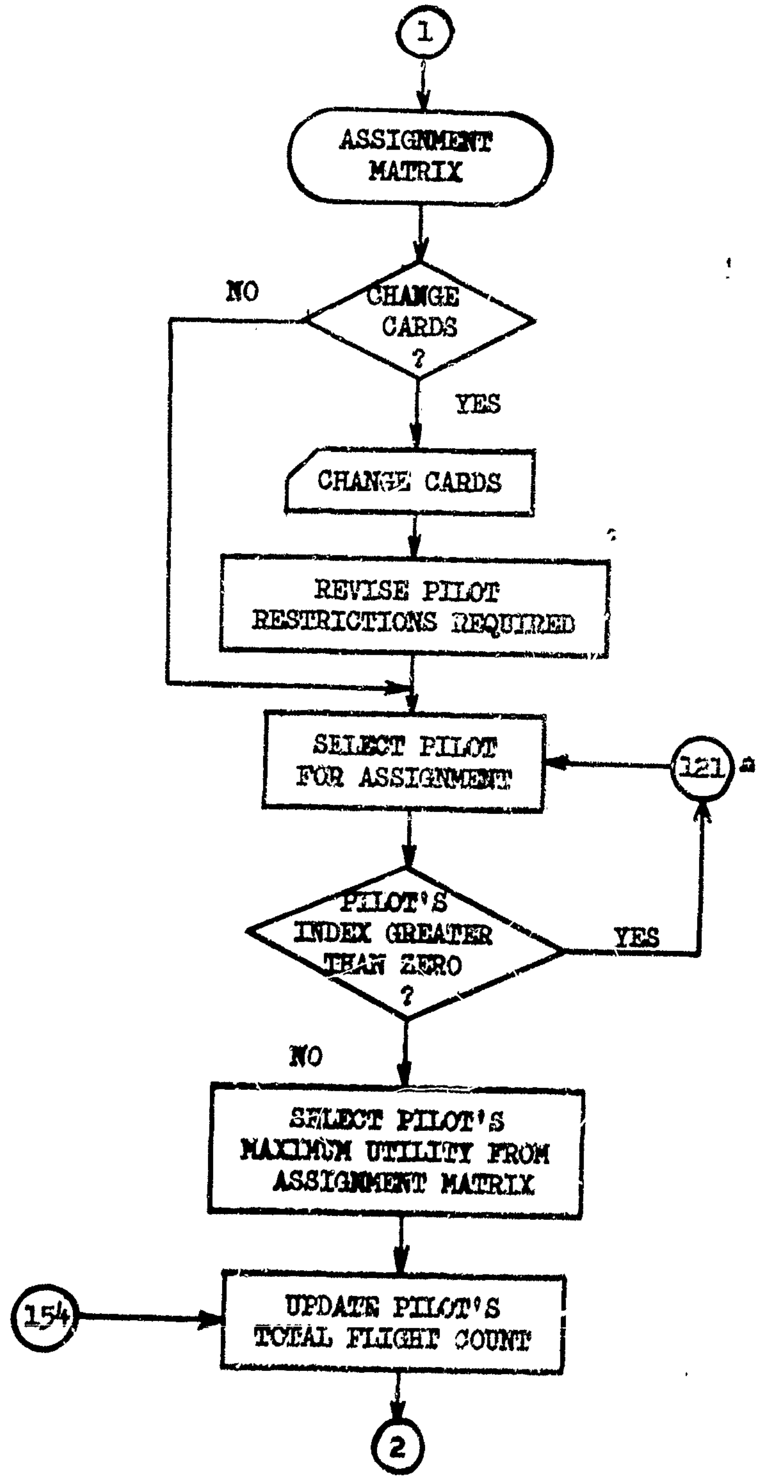
1. Does the number of pilot cards equal the number punched on the first card?
2. Do the daily cards begin with a Monday and are there six cards per week?
3. Are any numbers punched in columns one through five in the daily cards except the first day of each month (exclude first month) and on the last card in set?
4. Are the pilot cards in decreasing order of availability?
5. Is there a utility of two given for one day period and one night period of those available on each pilot card?
6. Does the number punched on the change card equal the number of pilot change cards used?
7. Is the pilot number of each pilot the same as his location in the pilot card set?
8. Are there four monthly maxima punched on the pilot change cards? If fewer months are utilized - punch zeros.

APPENDIX B

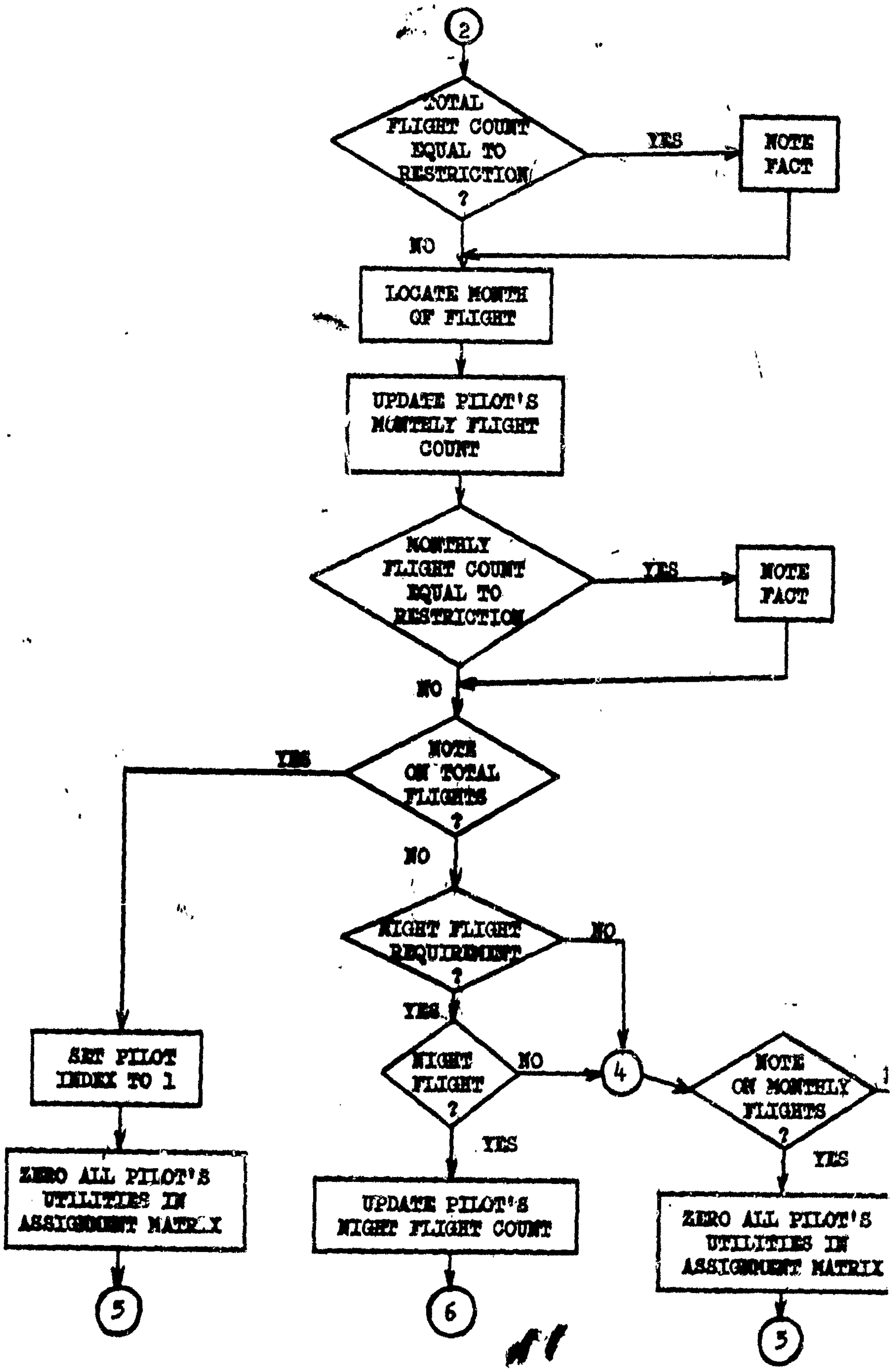
SUMMARY FLOW DIAGRAMS

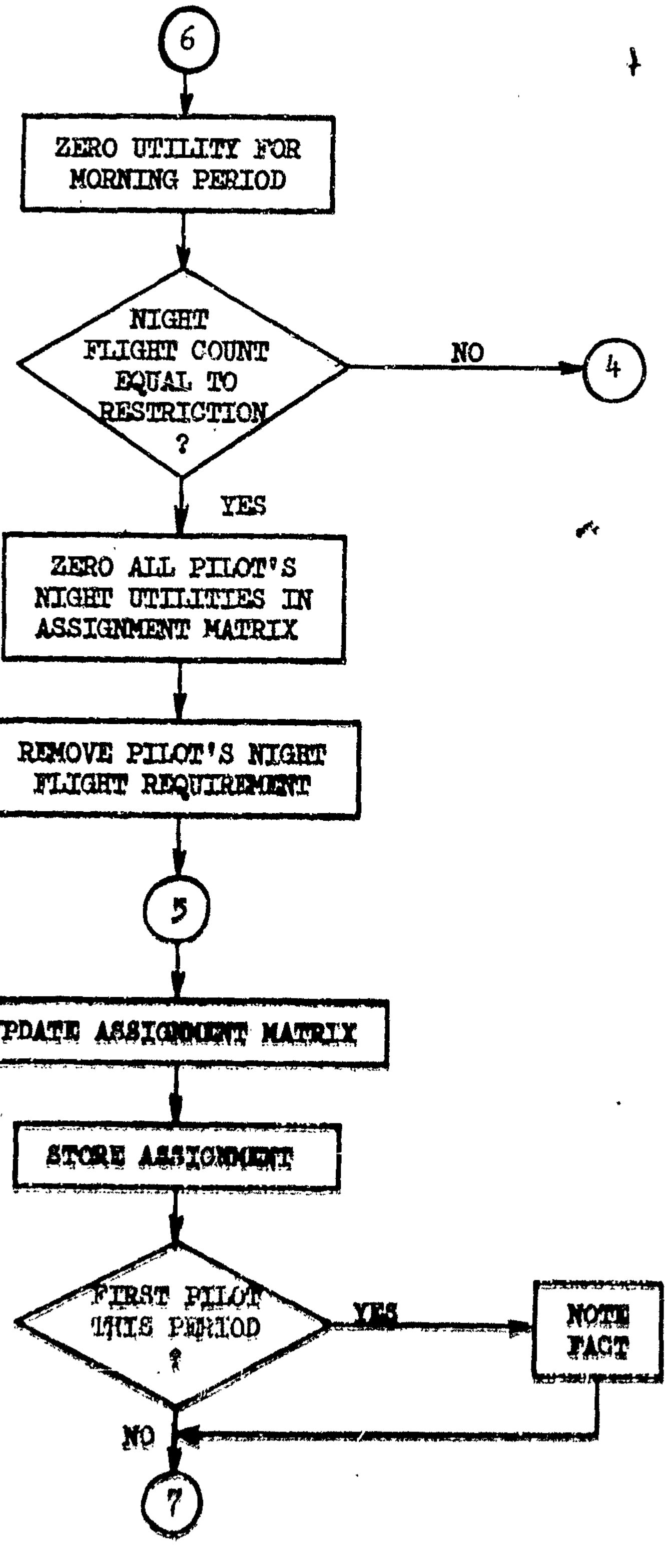
MAIN PROGRAM

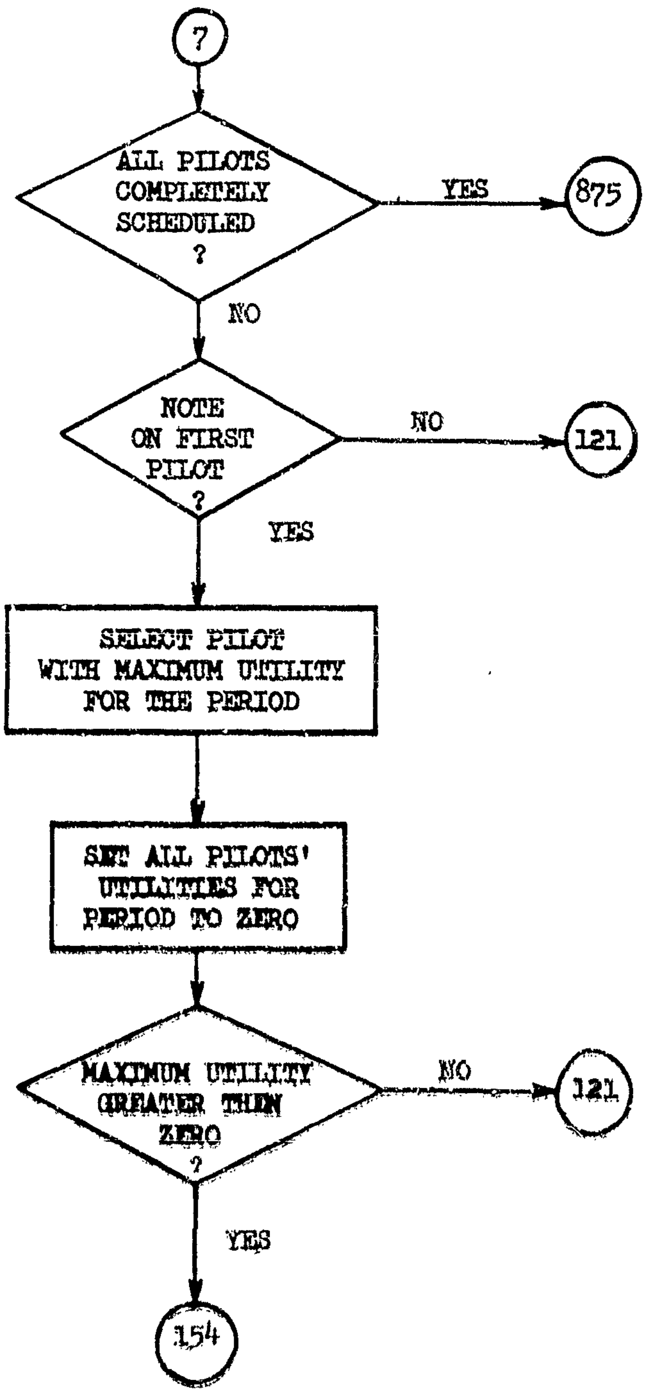


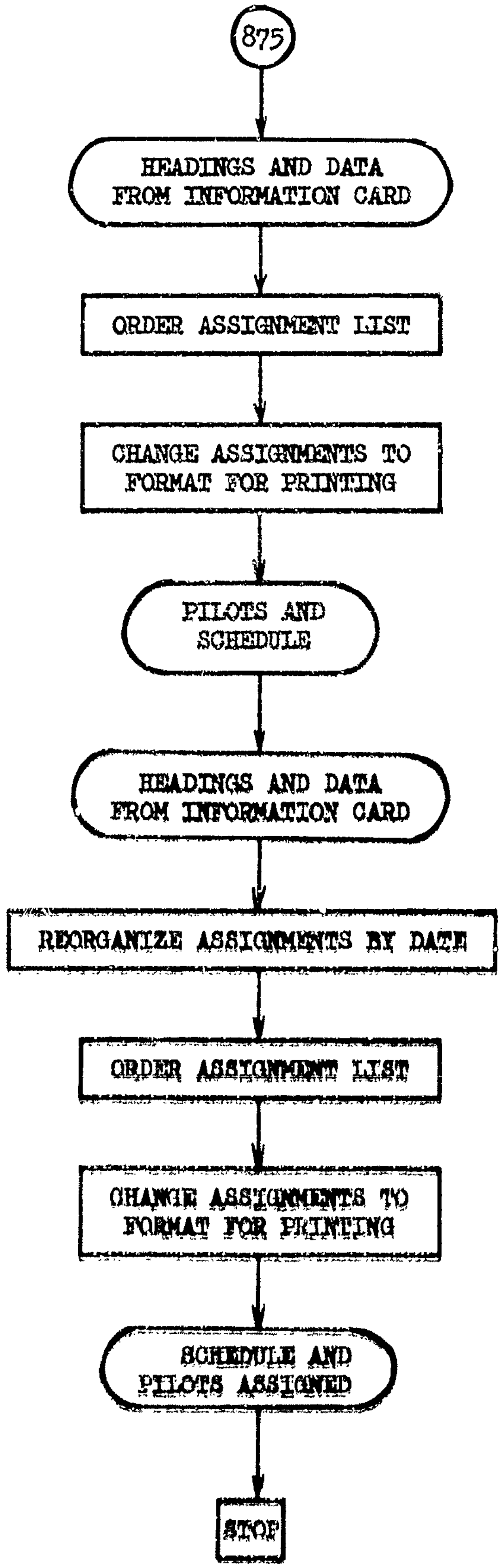


* ALL THREE DIGIT NUMBERS ARE COMPATIBLE WITH PROGRAM LISTING

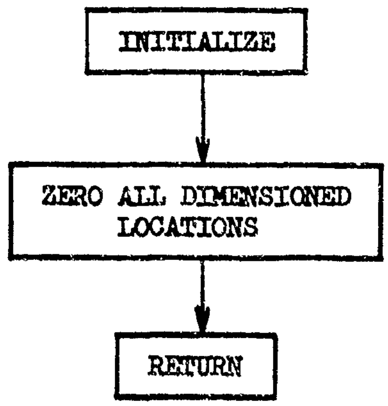

2
TOTAL FLIGHT COUNT EQUAL TO RESTRICTION ?
YES
NOTE FACT
NO
LOCATE MONTH OF FLIGHT
UPDATE PILOT'S MONTHLY FLIGHT COUNT
MONTHLY FLIGHT COUNT EQUAL TO RESTRICTION
YES
NOTE FACT
NO
NOTE ON TOTAL FLIGHTS ?
YES
NO
NIGHT FLIGHT REQUIREMENT ?
NO
YES
NIGHT FLIGHT ?
NO
4
NOTE ON MONTHLY FLIGHTS ?
YES
SET PILOT INDEX TO 1
ZERO ALL PILOT'S UTILITIES IN ASSIGNMENT MATRIX
5
YES
UPDATE PILOT'S NIGHT FLIGHT COUNT
6
YES
ZERO ALL PILOT'S UTILITIES IN ASSIGNMENT MATRIX
5

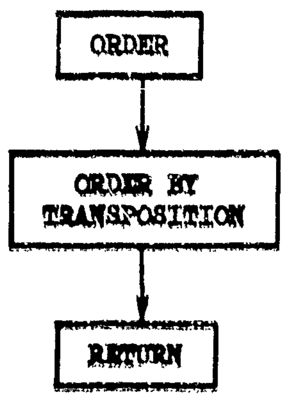

6
ZERO UTILITY FOR MORNING PERIOD
NIGHT FLIGHT COUNT EQUAL TO RESTRICTION ?
NO
4
YES
ZERO ALL PILOT'S NIGHT UTILITIES IN ASSIGNMENT MATRIX
REMOVE PILOT'S NIGHT FLIGHT REQUIREMENT
5
UPDATE ASSIGNMENT MATRIX
STORE ASSIGNMENT
FIRST PILOT THIS PERIOD ?
YES
NOTE FACT
NO
7


7
ALL PILOTS COMPLETELY SCHEDULED ?
YES
875
NO
NOTE ON FIRST PILOT ?
NO
121
YES
SELECT PILOT WITH MAXIMUM UTILITY FOR THE PERIOD
SET ALL PILOTS' UTILITIES FOR PERIOD TO ZERO
MAXIMUM UTILITY GREATER THEN ZERO ?
NO
121
YES
154


875
HEADINGS AND DATA FROM INFORMATION CARD
ORDER ASSIGNMENT LIST
CHANGE ASSIGNMENTS TO FORMAT FOR PRINTING
PILOTS AND SCHEDULE
HEADINGS AND DATA FROM INFORMATION CARD
REORGANIZE ASSIGNMENTS BY DATE
ORDER ASSIGNMENT LIST
CHANGE ASSIGNMENTS TO FORMAT FOR PRINTING
SCHEDULE AND PILOTS ASSIGNED
STOP

SUBROUTINES





APPENDIX C PROGRAM LISTINGS

INITIALIZING SUBROUTINE

```
      SUBROUTINE INIT
      DIMENSION  IC(330),PER(330), ITM(18),PILOT(35), A(35), B(35),
     1M(35,330), JC(35),JJ(35),MAX(35), NF(35), NTF(35),MM(5),
     2MF(35,5) ,MMF(35,5), RM(5),  AM(500), NM(500),
     3NITE(35),MNITE(35)
      COMMON IC, PER, ITM, PILOT, A, B, M, JC, JJ, MAX, NF, NTF, MM, M
     1MMF, RM,  AM, NM, NITE, MNITE
C     THIS SUBROUTINE IS USED TO REMOVE THE CORE CONSTANT FROM THE
C     DIMENSIONED LOCATIONS IN THE MAIN PROGRAM
C     THIS SUBROUTINE IS NOT NECESSARY IF THIS FUCTION IS AVAILABLE
C     IN THE MONITOR SYSTEM UTILIZED
C     THIS FUCTION IS PERFORMED BY A SERIES OF DO LOOP IN WHICH EACH
C     VARIABLE LOCATION DIMENSIONED IS SET TO ZERO
      DO 1  I =1,330
      IC(I) =0
      PER(I) =0
      DO 2  J =1,35
      M(J,I) =0
    2 CONTINUE
    1 CONTINUE
      DO 3  I=1,35
      PILOT(I) =0
      A(I) =0
      B(I) =0
      JC(I) =0
      JJ(I) =0
      MAX(I) =0
      NF(I) =0
      NTF(I) =0
      NITE(I) =0
      MNITE(I) =0
      DO 4  J =1,5
      MF(I,J) =0
      MMF(I,J) =0
    4 CONTINUE
    3 CONTINUE
      DO 5  I=1,500
      AM(I) = 0.
      NM(I) =0
    5 CONTINUE
      DO 6  I=1,18
      ITM(I) =0
    6 CONTINUE
      DO 7  I=1,5
      RM(I) =0
    7 CONTINUE
      RETURN
      END
```

```
      DIMENSION  IC(330),PER(330), ITM(18),PILOT(35), A(35), B(35
     1M(35,330), JC(35),JJ(35),MAX(35), NF(35), NTF(35),MM(5),
     2MF(35,5) ,MMF(35,5), RM(5),  AM(500), NM(500),
     3NITE(35),MNITE(35)
      COMMON IC, PER, ITM, PILOT, A, B, M, JC, JJ, MAX, NF, NTF.
     1MMF, RM,  AM, NM, NITE, MNITE
C     FOR BEST RESULTS THE OBJECT DECK OF THIS PROGRAM SHOULD BE
C     ON THE IBM 7090 THE COMPILATION TAKES APPROXIMATELY TWO MI
C     TWO SUBROUTINES ARE USED
C     THE PROGRAM IS DIVIDED INTO THREE BASIC SECTIONS, 1. FORMI
C     INITIAL ASSIGNMENT MATRIX, 2.  THE SCHEDULING AND 3. PRINT
C     THE OUTPUT
C     THESESECTIONS ARE FURTHER DIVIDED INTO MORE BASIC UNITS WH
C     ARE IDENTIFIED BY COMMENTS
C     PERIODS WHICH WOULDBE APPROXIMATELY 13 WEEKS
C     THE PROGRAM IS DESIGNED TO HANDLE A MAXIMUM OF 35 PILOTS A
C     THE MAJOR VARIABLES
C     M(I,J) = ELEMENTS OF THE ASSIGNMENT MATRIX
C              I = PILOT NUMBER
C              J = THE PERIOD NUMBER
C     IJJ    = NUMBER OF TOTAL PERIODS
C     M(II) = FIRST PERIOD OF THE MONTH
C              II = THE NUMBER OF THE MONTH
C     MPIL   = THE PILOT BEING ASSIGNED
C     MP     = THE PERIOD BEING ASSIGNED
C     AM(N) = THE LOCATION OF THE ASSIGNMENT RECORDED AS A NUMBER
      CALL INIT
      JUMP = 1
      JOE = 1
      RT = 0
      NN = 1
      READ 215,NPIL,MAXTF,MAXMF, NMNIT
      LST=NPIL
      DO 32 I=1,NPIL
      NTF(I)=MAXTF
      MNITE(I) = NMNIT
      DO 36 II=1,5
      MMF(I,II)=MAXMF
   36 CONTINUE
   32 CONTINUE
      READ 218,C,D,E,G,H
C     INPUT TO FORM ASSIGNMENT MATRIX
C     PERIOD AVAILABILITY -MAX SEMIANNUAL, OMITTING SUNDAYS
      J=0
      II = 1
      MM(II) = 0
      WRITE OUTPUT TAPE 6,718,II,MM(II)
      IEND = 0
      DO 21 N=1,350
      READ 200, IEND,DATE,DAY, FP, SP,TP
      IC(J+1)=FP
      IC(J+2)=SP
      IC(J+3)=TP
      PER(J+1)=1000.0*DATE +10.0*DAY+1.0
```

```
      PER(J+2)=1000.0*DATE +10.0*DAY+2.0
      PER(J+3)=1000.0*DATE+10.0*DAY+3.0
      J = J + 3
      IF(IEND)23,21,22
   23 II=II+1
      MM(II)=J-3
      WRITE OUTPUT TAPE 6,718,II,MM(II)
      IEND = 0
   21 CONTINUE
C     PILOT AVAILABILITY - WEEKLY
   22 IJJ=J
      II=II+1
      MM(II)=IJJ
C     IJJ IS NUMBER OF PERIODS
      DO 31 I=1,NPIL
C     NPIL IS NUMBER OF PILOTS (INPUT)
      J=1
      READ 210,PILOT(I),A(I),B(I),( ITM(NN),NN=1,18)
   88 DO 41 NN=1,18
      M(I,J)=IC(J)*ITM(NN)
      J=J+1
   41 CONTINUE
      IF(J-IJJ)88,88,31
   31 CONTINUE
C     OUTPUT OF ASSIGNMENT MATRIX
      WRITE OUTPUT TAPE 6,275 ,JOE
      WRITE OUTPUT TAPE 6,887
      WRITE OUTPUT TAPE 6,618
      WRITE OUTPUT TAPE 6, 996
      DO 285 J=1,IJJ
      WRITE OUTPUT TAPE 6,666,J,PER(J),(M(I,J),I=1,NPIL)
  285 CONTINUE
C     TO CHANGE PILOTS FLIGHT IF REQUIRED
      WRITE OUTPUT TAPE 6,275 ,JOE
      WRITE OUTPUT TAPE 6,888
      READ 228, NDD
      IF(NDD)153,153,381
  381 DO 90 I=1,NDD
      READ 237, JPIL,JTF,JFMF,JSMF,JTMF,JQMF,NMNIT
      MTF(JPIL) = JTF
      MMF(JPIL,1) =JFMF
      MMF(JPIL,2)=JSMF
      MMF(JPIL,3)=JTMF
      MMF(JPIL,4)=JQMF
      MNITE(JPIL) = NMNIT
   90 CONTINUE
  153 II=1
      PILOT(NPIL+1)=0.0
      A(NPIL+1)=0.0
      B(NPIL+1)=0.0
C     SELECTION OF PILOTS MAXIMUM UTILITY
  121 DO 40 I=1,NPIL
      IF(JC(I))140,110,140
  14  NY = NY + 1
      GO TO 40
  110 JJ(I)=1
      MAX(I)=M(I,1)
      DO 50 J=2,IJJ
      IF(MAX(I)-M(I,J))113,50,50
  113 MAX(I)=M(I,J)
```

```
      JJ(I)=J
   50 CONTINUE
      IF(MAX(I))144, 51,144
   51 LST = LST - 1
      JC(I) = 1
      WRITE OUTPUT TAPE 6,201,PILOT(I),[illegible](I),[illegible](I)
      IF(LST)252,875,40
  144 NS=[illegible]
      MP = JJ(I)
      MPIL = I
C     TO TEST FEASIBILITY OF TENTATIVE ASSIGNMENT
  154 NF(MPIL)=NF(MPIL)+1
      IF(NF(MPIL)-NTF(MPIL))217,221,221
  221 RT=1.0
  217 DO 75 II=1,4
      IF(MP-MM(II+1))227,75,75
   75 CONTINUE
  227 MF(MPIL,II)=MF(MPIL,II)+1
      IF(MF(MPIL,II)-MMF(MPIL,II))207,222,222
  222 RM(II)=1.
C     TO UP GRADE ASSIGNMEN MATRIX
  207 IF(RT)252,637,287
C     TO KEEP TRACK OF THE NIGHT FLIGHT.
  637 IF(MNITE(MPIL))252,247,636
  636 BMP = MP/3
      AMP = MP
      DIFT = AMP/3.0 - BMP
      IF(DIFT)247,387,247
  387 NITE(MPIL) = NITE(MPIL) + 1
      M(MPIL,MP-2)=0
      IF(NITE(MPIL)-MNITE(MPIL))247,322,322
C     WHEN JUMP =3 CLEAR NIGHT PERIODS, WHEN JUMP=1 CLEARS ALL P
  322 JUMP = 3
      LS = 3
      MNITE(MPIL) = 0
      GO TO 286
  287 LTS = LTS - 1
      JC(MPIL) =1
      MAX(MPIL) = 0
      LS=1
  286 LF=IJJ
      RT = 0.0
      GO TO 176
  247 IF(RM(II))252,292,249
  249 LS=MM(II) +1
      LF=MM(II+1)
      RM(II) = 0.0
  176 DO 82 LL=LS,LF,JUMP
      M(MPIL,LL)=0
   82 CONTINUE
      RM(II) = 0.0
      JUMP = 1
  292 M(MPIL,MP)=0
      M(MPIL,MP-1)=0
      M(MPIL,MP+1)=0
      MAX(MPIL) =0
C     STORE ASSIGNMENTS
  190 PIL = MPIL
      AM(NN) = PIL*100000. + PER(MP)
      NN=NN+1
```

```
      IF(LST)875,875,216
  216 NS=NS+1
      IF(NS-1)213,213,179
  213 MPIL=1
      MMXX = M(1,MP)
      DO 85 L=2,NPIL
      IF(M(L,MP)-MMXX) 85,163,163
  163 MPIL=L
      MMXX = M(MPIL,MP)
   85 CONTINUE
      IF(M(MPIL,MP))192,192,154
  192 MPIL= NPIL+1
      GO TO 19?
  179 DO 93 L=1,NPIL
   93 M(L,MP)=0
   4? CONTINUE
      IF(NY-NPIL)340,875,340
  34? NY = ?
      IF(LST)252,875,121
C     TO PRINT PILOTS AND SCHEDULE
  875 WRITE OUTPUT TAPE 6,275 ,JOE
      WRITE OUTPUT TAPE 6,889
      WRITE OUTPUT TAPE 6,236,C,D,E,G,H
      WRITE OUTPUT TAPE 6,251
      NN = NN-1
      NNP=1
      CALL INORD(NN,AM)
      J = 1
      DO 73 N=1,NN
      PP = AM(N)/100000.
      NPP = PP
      APP = NPP
      ANI = AM(N)-APP*100000.
      IF( NNP-NPP)158,131,158
  131 NM(J) = ANI
      J=J+1
      GO TO 73
  158 WRITE OUTPUT TAPE 6, 253,PILOT(NNP),A(NNP),B(NNP)
  150 IJ=0
      JET = J-1
      DO 71 I=2,JET
      IF(NM(I-1)-NM(I)) 71, 71,160
  160 ITEMP=NM(I)
      NM(I)=NM(I-1)
      NM(I-1)=ITEMP
      IJ=99
   71 CONTINUE
      IF(IJ)182,182,150
  182 DO 67 I=1,JET
      NNM=NM(I)/1000
      NND=NM(I)-NNM*1000
      ND=NND/10
      NP=NND-ND*10
      WRITE OUTPUT TAPE 6,255,NNM,ND,NP
   67 CONTINUE
      NNP=NPP
      NM(1)=ANI
      J=2
   73 CONTINUE
C     TO PRINT DATE AND PILOTS SCHEDULED
```

```
      WRITE OUTPUT TAPE 6,275 ,JOE
      WRITE OUTPUT TAPE 6,890
      WRITE OUTPUT TAPE 6,236,C,D,E,G,H
      WRITE OUTPUT TAPE 6,270
      DO 86 N=1,NN
      PP = AM(N)/100000.
      NPP = PP
      APP = NPP
      ANI = AM(N)-APP*100000.
      AM(N) = ANI*100. + APP
   86 CONTINUE
      CALL INORD(NN,AM)
      DO 68 N=1,NN
      AMM = AM(N)/100.
      MM = AMM
      AMMM = AM(N+1)/100.
      MMM = AMMM
      AMM = MM
      IF(MM-MMM)151,173,151
  173 JIG=1
      IPIL = AM(N)-AMM*100.
      GO TO 68
  151 IF(JIG)321,396,321
  396 IPIL=NPIL+1
      GO TO 966
  321 JIG=0
      LPIL = AM(N)-AMM*100.
      MON=MM/1000
      MD=MM-MON*1000
      MDAY=MD/10
      MPER=MD-MDAY*10
  966 WRITE OUTPUT TAPE 6,277,MON,MDAY,MPER,PILOT(IPIL),A(IPIL),
     1PILOT(LPIL),A(LPIL),B(LPIL)
   68 CONTINUE
  201 FORMAT(9X,3A5,33HHAS NOT BEEN COMPLETELY SCHEDULED   )
  888 FORMAT (9X,6HPART 2/9X,18HPILOT PROBLEM AREA//)
  887 FORMAT (9X,26HAPPENDIX E  PROGRAM OUTPUT   ///9X,6HPART 1/)
  889 FORMAT (9X,6HPART 3/9X,5HPILOT)
  890 FORMAT (9X,6HPART 4/9X,5HDAILY)
  666 FORMAT (9X,I3,5X,F5.0,5X,32(I2,1X))
  996 FORMAT (9X,12HPER      DATE,30X,6HPILOTS/)
  618 FORMAT (9X,25HINITIAL ASSIGNMENT MATRIX        )
  718 FORMAT (9X,21HFIRST PERIOD IN MONTH,I5,2HIS,I5)
  270 FORMAT(14X,4HDATE/9X,24HMONTH    DAY     PERIOD    ,5X,5HPILO
     15HPILOT//)
  215 FORMAT    (4I5)
  237 FORMAT    (7I5)
  228 FORMAT    (I5)
  218 FORMAT    (5A5)
  236 FORMAT(9X,15HFLIGHT SCHEDULE///9X,5A5//)
  251 FORMAT(14X,5HPILOT,5X,5H DATE, 8H  PERIOD/)
  200 FORMAT    (I5,5F3.0)
  253 FORMAT (9X,3A5)
  255 FORMAT (24X,2I3,3X,I2)
  275 FORMAT (I1)
  277 FORMAT (10X,3(I3,5X),2(3A5,5X))
  210 FORMAT    (3A5,6(3I1,1X))
  252 CALL SYSTEM
      END
```

APPENDIX D INPUT

```
$DATA
   25    06     3     1
CKT JULY AUGUST 64
           7  6  0  4
           7  7  4  4
           7  8  4  4
           7  9  4  4
           7 10  4  4
           7 11  3  0  0
           7 13  0  4  9
           7 14  4  4  9
           7 15  4  4  9
           7 16  4  4  9
           7 17  4  4  9
           7 18  3  0  0
           7 20  0  4
           7 21  4  4
           7 22  4  4
           7 23  4  4
           7 24  4  4
           7 25  3  0  0
           7 27  0  4  0
           7 28  4  4  0
           7 29  4  4  0
           7 30  4  4  0
           7 31  4  4  0
      -1   8  1  3  0  0
           8  3  0  4
           8  4  4  4
           8  5  4  4
           8  6  4  4
           8  7  4  4
           8  8  3  0  0
           8 10  0  4  0
           8 11  4  4  0
           8 12  4  4  0
           8 13  4  4  0
           8 14  4  4  0
           8 15  3  0  0
           8 17  0  4  0
           8 18  4  4  0
           8 19  4  4  0
           8 20  4  4  0
           8 21  4  4  0
           8 22  3  0  0
           8 24  0  4  9
           8 25  4  4  9
           8 26  4  4  9
           8 27  4  4  9
           8 28  4  4  9
           8 29  3  0  0
       1   8 31  0  4  9
 WASSELL             111 122 111 111 111 000
 CAREY               002 020 111 000 111 111
 BENNETT             000 000 020 10[illegible] 01[illegible] 100
```

CIOIA	002	121	001	111	001	000
GREEN	010	000	210	000	110	000
WEISBARTH	120	000	110	000	110	000
PETERSON	000	000	001	121	012	000
MITTENDORF	002	200	001	011	001	000
MAYO	000	200	001	112	001	100
HOLLEY	000	020	000	010	000	000
TRACY	000	200	100	101	102	000
POWELL	000	000	000	110	120	000
PENNINGTON	000	000	000	110	120	000
MAHONEY	000	000	000	210	000	100
FAY	002	000	001	020	001	100
KEITH	000	000	000	211	012	000
KEATING	000	002	001	001	001	200
BELL	000	000	002	210	001	100
KUROWSKI	000	000	000	020	000	000
NICOLOAN	000	000	002	210	000	000
DUQUETTE	000	000	000	020	002	100
ADAMS	000	000	002	210	000	000
DAVIS	000	000	000	000	000	200
ARMSTRONG	000	000	002	200	000	000
WILSON	000	020	000	000	000	000

10

3	04	3	1	1
4	04	3	1	
5	04	3	1	
6	03	0	3	
10	04	3	1	
12	04	3	1	
13	03	3	0	
14	04	3	1	
19	04	3	1	
23	04	3	1	

1b

APPENDIX E PROGRAM OUTPUT

PART 1

INITIAT ASSIGNMENT MATRIX

PER	DATE								
1	7061	0	0	0	0	0	0	0	0
2	7062	4	0	0	0	4	8	0	0
3	7063	-0	-0	-0	-0	-0	-0	-0	-0
4	7071	4	0	0	4	0	0	0	8
5	7072	8	8	0	8	0	0	0	0
6	7073	-0	-0	-0	-0	-0	-0	-0	-0
7	7081	4	4	0	0	8	4	0	0
8	7082	4	4	8	0	4	4	0	0
9	7083	-0	-0	-0	-0	-0	-0	-0	-0
10	7091	4	0	4	4	0	0	4	0
11	7092	4	0	0	4	0	0	8	4
12	7093	-0	-0	-0	-0	-0	-0	-0	-0
13	7101	4	4	0	0	4	4	0	0
14	7102	4	4	4	0	4	4	4	0
15	7103	-0	-0	-0	-0	-0	-0	-0	-0
16	7111	0	3	3	0	0	0	0	0
17	7112	0	0	0	0	0	0	0	0
18	7113	0	0	0	0	0	0	0	0
19	7131	0	0	0	0	0	0	0	0
20	7132	4	0	0	0	4	8	0	0
21	7133	9	18	0	18	0	0	0	18
22	7141	4	0	0	4	0	0	0	8
23	7142	8	8	0	8	0	0	0	0
24	7143	18	0	0	9	0	0	0	0
25	7151	4	4	0	0	8	4	0	0
26	7152	4	4	8	0	4	4	0	0
27	7153	9	9	0	9	0	0	9	9
28	7161	4	0	4	4	0	0	4	0
29	7162	4	0	0	4	0	0	8	4
30	7163	9	0	0	9	0	0	9	9
31	7171	4	4	0	0	4	4	0	0
32	7172	4	4	4	0	4	4	4	0
33	7173	9	9	0	9	0	0	18	9
34	7181	0	3	3	0	0	0	0	0
35	7182	0	0	0	0	0	0	0	0
36	7183	0	0	0	0	0	0	0	0
37	7201	0	0	0	0	0	0	0	0
38	7202	4	0	0	0	4	8	0	0
39	7203	-0	-0	-0	-0	-0	-0	-0	-0
40	7211	4	0	0	4	0	0	0	8
41	7212	8	8	0	8	0	0	0	0
42	7213	-0	-0	-0	-0	-0	-0	-0	-0
43	7221	4	4	0	0	8	4	0	0
44	7222	4	4	8	0	4	4	0	0
45	7223	-0	-0	-0	-0	-0	-0	-0	-0
46	7231	4	0	4	4	0	0	4	0
47	7232	4	0	0	4	0	0	8	4
48	7233	-0	-0	-0	-0	-0	-0	-0	-0
49	7241	4	4	0	0	4	4	0	0
50	7242	4	4	4	0	4	4	4	0
51	7243	-0	-0	-0	-0	-0	-0	-0	-0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
4	4	4	8	0	8	0	8	0	8	0	8	0	8	0
0	4	4	4	8	4	0	4	8	4	8	4	0	0	0
-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
4	4	4	0	0	0	0	0	0	0	0	0	0	0	0
0	8	8	0	0	4	0	0	0	0	0	0	0	0	0
-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
0	0	0	3	3	0	6	3	0	0	3	0	6	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	18	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	18	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	9	0	9	18	0	18	0	18	0	18	0
4	4	4	8	0	8	0	8	0	8	0	8	0	8	0
0	4	4	4	8	4	0	4	8	4	8	4	0	0	0
9	0	0	0	0	9	9	0	0	0	0	0	0	0	0
4	4	4	0	0	0	0	0	0	0	0	0	0	0	0
0	8	8	0	0	4	0	0	0	0	0	0	0	0	0
8	0	0	0	9	18	9	9	0	0	18	0	0	0	0
0	0	0	3	3	0	6	3	0	0	3	0	6	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
[illegible]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[illegible]	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[illegible]	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
[illegible]	4	4	8	0	8	0	8	0	8	0	8	0	8	0
[illegible]	4	4	4	8	4	0	4	8	4	8	4	0	0	0
[illegible]	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
[illegible]	4	4	0	0	0	0	0	0	0	0	0	0	0	0
[illegible]	8	8	0	0	4	0	0	0	0	0	0	0	0	0
[illegible]	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0

16

52	7251	0	3	3	0	0	0	0	0
53	7252	0	0	0	0	0	0	0	0
54	7253	0	0	0	0	0	0	0	0
55	7271	0	0	0	0	0	0	0	0
56	7272	4	0	0	0	4	8	0	0
57	7273	0	0	0	0	0	0	0	0
58	7281	4	0	0	4	0	0	0	[illegible]
59	7282	8	8	0	8	0	0	0	0
60	7283	0	0	0	0	0	0	0	0
61	7291	4	4	0	0	8	4	0	0
62	7292	4	4	8	0	4	4	0	[illegible]
63	7293	0	0	0	0	0	0	0	[illegible]
64	7301	4	0	4	4	0	0	4	[illegible]
65	7302	4	0	0	4	0	0	8	4
66	7303	0	0	0	0	0	0	0	[illegible]
67	7311	4	4	0	0	4	4	0	[illegible]
68	7312	4	4	4	0	4	4	4	[illegible]
69	7313	0	0	0	0	0	0	0	[illegible]
70	8011	0	3	3	0	0	0	0	[illegible]
71	8012	0	0	0	0	0	0	0	0
72	8013	0	0	0	0	0	0	0	[illegible]
73	8031	0	0	0	0	0	0	0	[illegible]
74	8032	4	0	0	0	4	8	0	[illegible]
75	8033	-0	-0	-0	-0	-0	-0	-0	[illegible]
76	8041	4	0	0	4	0	0	0	[illegible]
77	8042	8	8	0	8	0	0	0	[illegible]
78	8043	-0	-0	-0	-0	-0	-0	-0	[illegible]
79	8051	4	4	0	0	8	4	0	[illegible]
80	8052	4	4	8	0	4	4	0	[illegible]
81	8053	-0	-0	-0	-0	-0	-0	-0	[illegible]
82	8061	4	0	4	4	0	0	4	[illegible]
83	8062	4	0	0	4	0	0	8	[illegible]
84	8063	-0	-0	-0	-0	-0	-0	-0	[illegible]
85	8071	4	4	0	0	4	4	0	[illegible]
86	8072	4	4	4	0	4	4	4	[illegible]
87	8073	-0	-0	-0	-0	-0	-0	-0	[illegible]
88	8081	0	3	3	0	0	0	0	[illegible]
89	8082	0	0	0	0	0	0	0	[illegible]
90	8083	0	0	0	0	0	0	0	[illegible]
91	8101	0	0	0	0	0	0	0	[illegible]
92	8102	4	0	0	0	4	8	0	
93	8103	0	0	0	0	0	0	0	
94	8111	4	0	0	4	0	0	0	
95	8112	8	8	0	8	0	0	0	
96	8113	0	0	0	0	0	0	0	
97	8121	4	4	0	0	8	4	0	
98	8122	4	4	8	0	4	4	0	
99	8123	0	0	0	0	0	0	0	
100	8131	4	0	4	4	0	0	4	
101	8132	4	0	0	4	0	0	8	
102	8133	0	0	0	0	0	0	0	
103	8141	4	4	0	0	4	4	0	
104	8142	4	4	4	0	4	4	4	
105	8143	0	0	0	0	0	0	0	
106	8151	0	3	3	0	0	0	0	
107	8152	0	0	0	0	0	0	0	
108	8153	0	0	0	0	0	0	0	
109	8171	0	0	0	0	0	0	0	
110	8172	4	0	0	0	4	8	0	
111	8173	0	0	0	0	0	0	0	

0	0	0	3	3	0	6	3	0	0	3	0	6	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	4	4	8	0	8	0	8	0	8	0	8	0	8	0
0	4	4	4	8	4	0	4	8	4	8	4	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	4	4	0	0	0	0	0	0	0	0	0	0	0	0
0	8	8	0	0	4	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	3	3	0	6	3	0	0	3	0	6	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
4	4	4	8	0	8	0	8	0	8	0	8	0	8	0
0	4	4	4	8	4	0	4	8	4	8	4	0	0	0
-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
4	4	4	0	0	0	0	0	0	0	0	0	0	0	0
0	8	8	0	0	4	0	0	0	0	0	0	0	0	0
-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
0	0	0	3	3	0	6	3	0	0	3	0	6	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	4	4	8	0	8	0	8	0	8	0	8	0	8	0
0	4	4	4	8	4	0	4	8	4	8	4	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	4	4	0	0	0	0	0	0	0	0	0	0	0	0
0	8	8	0	0	4	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	3	3	0	6	3	0	0	3	0	6	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

112	8181	4	0	0	4	0	0	0	8	8
113	8182	8	8	0	8		0	0	0	0
114	8183	0	0	0	0	0	0	0	0	0
115	8191	4	4	0	0	8	4	0	0	0
116	8192	4	4	8	0	4	4	0	0	0
117	8193	0	0	0	0	0	0	0	0	0
118	8201	4	0	4	4	0	0	4	0	4
119	8202	4	0	0	4	0	0	8	4	4
120	8203	0	0	0	0	0	0	0	0	0
121	8211	4	4	0	0	4	4	0	0	0
122	8212	4	4	4	0	4	4	4	0	0
123	8213	0	0	0	0	0	0	0	0	0
124	8221	0	3	3	0	0	0	0	0	3
125	8222	0	0	0	0	0	0	0	0	0
126	8223	0	0	0	0	0	0	0	0	0
127	8241	0	0	0	0	0	0	0	0	0
128	8242	4	0	0	0	4	8	0	0	0
129	8243	9	18	0	18	0	0	0	18	0
130	8251	4	0	0	4	0	0	0	8	8
131	8252	8	8	0	8	0	0	0	0	0
132	8253	18	0	0	9	0	0	0	0	[illegible]
133	8261	4	4	0	0	8	4	0	0	0
134	8262	4	4	8	0	4	4	0	0	0
135	8263	9	9	0	9	0	0	9	9	9
136	8271	4	0	4	4	0	0	4	0	4
137	8272	4	0	0	4	0	0	8	4	4
138	8273	9	0	0	9	0	0	9	9	18
139	8281	4	4	0	0	4	4	0	0	0
140	8282	4	4	4	0	4	4	4	0	0
141	8283	9	9	0	9	0	0	18	9	9
142	8291	0	3	3	0	0	0	0	0	3
143	8292	0	0	0	0	0	0	0	0	0
144	8293	0	0	0	0	0	0	0	0	0
145	8311	0	0	0	0	0	0	0	0	0
146	8312	4	0	0	0	4	8	0	0	0
147	8313	9	18	0	18	0	0	0	18	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	8
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	4	8	0	8	0	8	0	8	0	8	0	8	0
4	4	4	8	4	0	4	8	4	8	4	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	4	0	0	0	0	0	0	0	0	0	0	0	0
8	8	0	0	4	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	3	3	0	6	3	0	0	3	0	6	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	18	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	8
0	0	0	0	0	18	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	9	0	9	18	0	18	0	18	0	18	0
4	4	8	0	8	0	8	0	8	0	8	0	8	0
4	4	4	8	4	0	4	8	4	8	4	0	0	0
0	0	0	0	9	9	0	0	0	0	0	0	0	0
4	4	0	0	0	0	0	0	0	0	0	0	0	0
8	8	0	0	4	0	0	0	0	0	0	0	0	0
0	0	0	9	18	9	9	0	0	18	0	0	0	0
0	0	3	3	0	6	3	0	0	3	0	6	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	18	0	0	0	0	0	0	0	0	0	0

PART 2
PILOT PROBLEM AREA

ADAMS	HAS NOT BEEN COMPLETELY SCHEDULED
CIOIA	HAS NOT BEEN COMPLETELY SCHEDULED
MAHONEY	HAS NOT BEEN COMPLETELY SCHEDULED
BELL	HAS NOT BEEN COMPLETELY SCHEDULED
NICOLOAN	HAS NOT BEEN COMPLETELY SCHEDULED

PART 3
PILOT
FLIGHT SCHEDULE

CRT JULY AUGUST 64

PILOT	DATE	PERIOD
WASSELL		
	7 14	3
	7 21	2
	7 22	2
	8 10	2
	8 18	2
	8 25	2
CAREY		
	7 13	3
	7 22	1
	7 28	2
	8 5	1
	8 11	2
	8 25	2
BENNETT		
	7 8	2
	7 15	2
	7 22	2
	8 5	2
CIOIA		
	7 14	1
	8 24	3
GREEN		
	7 8	1
	7 15	2
	7 22	1
	8 3	2
WEISBARTH		
	7 8	2
	8 3	2
	8 10	2
PETERSON		
	7 17	3
	7 23	2
	7 31	2
	8 14	2
	8 20	2
	8 27	2
MITTENDORF		
	7 7	1
	7 21	1
	7 28	1
	8 11	1
	8 18	1
	8 24	3
MAYO		
	7 16	3
	7 21	1
	7 28	1

[redacted]

	8	4	1
	8	18	1
	8	25	1
HOLLEY			
	7	7	2
	7	14	2
	7	28	2
	8	4	2
TRACY			
	7	7	1
	7	8	1
	7	14	1
	8	4	1
	8	11	1
	8	28	3
POWELL			
	7	10	2
	7	17	2
	7	24	2
	8	7	2
PENNINGTON			
	7	10	2
	7	17	2
	7	24	2
MAHONEY			
	7	9	1
	8	6	1
FAY			
	7	9	2
	7	13	3
	7	30	2
	8	13	2
	8	15	1
	8	27	2
KEITH			
	7	16	3
	7	23	1
	7	31	2
	8	7	2
	8	13	1
	8	14	2
KEATING			
	7	11	1
	7	14	3
	7	18	1
	8	1	1
	8	8	1
	8	15	1
BELL			
	7	16	1
	7	25	1
	3	8	1
	8	20	1
	8	28	3
KUROWSKI			
	7	16	2
	7	23	2
	7	30	2
	8	6	2

	7	15	3
	7	30	1
	8	27	1
DUQUETTE			
	7	9	2
	7	16	2
	7	17	3
	8	6	2
	8	13	2
	8	20	2
ADAMS			
	7	15	3
	7	30	1
	8	20	1
	8	27	1
DAVIS			
	7	11	1
	7	18	1
	7	25	1
	8	1	1
ARMSTRONG			
	7	9	1
	7	16	1
	7	23	1
	8	6	1
	8	13	1
	8	26	3
WILSON			
	7	7	2
	7	14	2
	7	21	2
	8	4	2
	8	11	2
	8	18	2

PART 4
DAILY
FLIGHT SCHEDULE

CRT JULY AUGUST 64

DATE MONTH	DAY	PERIOD	PILOT	PILOT
7	7	1	MITTENDORF	TRACY
7	7	2	HOLLEY	WILSON
7	8	1	GREEN	TRACY
7	8	2	BENNETT	WEISBARTH
7	9	1	MAHONEY	ARMSTRONG
7	9	2	FAY	DUQUETTE
7	10	2	POWELL	PENNINGTON
7	11	1	KEATING	DAVIS
7	13	3	CAREY	FAY
7	14	1	CIJIA	TRACY
7	14	2	HOLLEY	WILSON
7	14	3	WASSELL	KEATING
7	15	2	BENNETT	GREEN
7	15	3	NICOLOAN	ADAMS
7	16	1	BELL	ARMSTRONG
7	16	2	KUROWSKI	DUQUETTE
7	16	3	MAYO	KEITH
7	17	2	POWELL	PENNINGTON
7	17	3	PETERSON	DUQUETTE
7	18	1	KEATING	DAVIS
7	21	1	MITTENDORF	MAYO
7	21	2	WASSELL	WILSON
7	22	1	CAREY	GREEN
7	22	2	WASSELL	BENNETT
7	23	1	KEITH	ARMSTRONG
7	23	2	PETERSON	KUROWSKI
7	24	2	POWELL	PENNINGTON
7	25	1	BELL	DAVIS
7	28	1	MITTENDORF	MAYO
7	28	2	CAREY	HOLLEY
7	30	1	NICOLOAN	ADAMS
7	30	2	FAY	KUROWSKI
7	31	2	PETERSON	KEITH
8	1	1	KEATING	DAVIS
8	3	2	GREEN	WEISBARTH
8	4	1	MAYO	TRACY
8	4	2	HOLLEY	WILSON
8	5	1	CAREY	000000000000000
8	5	2	BENNETT	000000000000000
8	6	1	MAHONEY	ARMSTRONG
8	6	2	KUROWSKI	DUQUETTE
8	7	2	POWELL	KEITH
8	8	1	KEATING	BELL
8	10	2	WASSELL	WEISBARTH
8	11	1	MITTENDORF	TRACY
8	11	2	CAREY	WILSON
8	12	1	KEITH	ARMSTRONG

8	13	2	FAY	DUQUETTE
8	14	2	PETERSON	KEITH
8	15	1	FAY	KEATING
8	18	1	MITTENDORF	MAYO
8	18	2	WASSELL	WILSON
8	20	1	BELL	ADAMS
8	20	2	PETERSON	DUQUETTE
8	24	3	CIOIA	MITTENDORF
8	25	1	MAYO	000000000000000
8	25	2	WASSELL	CAREY
8	26	3	ARMSTRONG	000000000000000
8	27	1	NICOLOAN	ADAMS
8	27	2	PETERSON	FAY
8	28	3	TRACY	BELL

APPENDIX F

PRELIMINARY RESEARCH IN THE APPLICATION OF LINEAR PROGRAMMING APPLIED TO ATTACHED PILOT SCHEDULING

Introduction

This appendix will present a brief review of the preliminary research undertaken with regard to applying linear programming to the problem of aircrew scheduling and the difficulties encountered. The specific problem is as stated under the Statement of the Problem presented in the text of this thesis. If questions arise as to the reasons for any course of action or definition of terms, the reader is referred again to the text.

Certain basic assumptions have been made as to the familiarity of the reader with the body of the text. It is further assumed that the reader has a working knowledge of linear programming and is familiar with the conversion of the transportation model to a standard linear programming format.

The text listed three basic criteria which the author believes any practical scheduling program must satisfy. To review, there is a definite need for simple input, rapid schedule generation and readily usable output. In the areas of simple input and readily usable output, there exists a definite need to devise computer programs which will convert any input to the linear program's input format and reduce the linear program's result to a readily usable format. The program's input and output should not differ greatly from those resulting from the heuristic program developed in the text. The author has done no research in the development of these programs, but rather, has limited his preliminary research to that of the generation of the linear programming problem. No attempt was made to solve the problem as defined in the thesis body by L/P due to time limitations and machine incapability. (Number of restrictions, etc.) These limitations will be further discussed in this appendix.

The restrictions to the problem which will be discussed here are those explained in the text.

A. Number of Pilots Assigned to Each Period.

B. Total Number of Flights During Scheduling Interval.

C. Minimum Number of Flights Per Month.

D. Minimum Number of Night Flights.

E. Integer Solution.

F. Maximum Number of Daily Flights.

G. Minimum Time Between Flights.

In the attempt to utilize linear programming to satisfy these restrictions, many additional assumptions were made. These assumptions will be identified as the restriction explanation progresses. The word restriction as used here has a two-fold meaning. (1) The problem restrictions are as used in the text of the thesis. (2) The program restriction, as the restrictions to the linear program. The type of restriction will be identified where confusion may arise. When considering the application of linear programming to a practical scheduling problem, size becomes a major factor.[4]

In conjunction with the large size of this type problem and the limited size of the linear programming capacity here at the University of Pittsburgh, a very small problem was undertaken. This was the scheduling of seven pilots for a 45-period month. The linear program used for this preliminary research was the simplex program developed by Mr. Wayne Baughman, University of Pittsburgh. This (50 restrictions, 99 variable) program was not readily adaptable and some revisions were made by the author to achieve problem solution. The simplex program, even after revisions, was still not of sufficient size to satisfy the problem requirement. The LP 90 (CEIR) program developed for use on the IBM 7090 (1024 restrictions, infinite variables) intuitively seemed to overcome the size problem[5] but due to its unavailability here at the University of Pittsburgh this line of research was discontinued and the heuristic approach was devised and developed as presented in the main body of this thesis.

Explanation of Restrictions as Applied to Linear Programming

The basic foundation for the program was a transportation tableau which was set up with the pilots as the origins and the available periods representing the destinations. The tableau was still much too large to be considered, therefore the periods where less than two pilots were available, were eliminated, thus reducing the dimensions of the transportation tableau to seven pilots and 19 periods. This tableau then became the basis for the formulation of the linear programming problem.

TRANSPORTATION TABLEAU

PERIODS

PILOTS		1	2	. . .	j	. . .	J	Total
	1	x_{11} c_{11}	x_{12} c_{12}	. . .	x_{1j} c_{1j}	. . .	x_{1J} c_{1J}	A_1
	2	x_{21} c_{21}	x_{22} c_{22}	. . .	x_{2j} c_{2j}	. . .	x_{2J} c_{2J}	A_2
	⋮	⋮	⋮		⋮		⋮	⋮
	i	x_{i1} c_{i1}	x_{i2} c_{i2}	. . .	x_{ij} c_{ij}	. . .	x_{iJ} c_{iJ}	A_i
	⋮	⋮	⋮		⋮		⋮	⋮
	I	x_{I1} c_{I1}	x_{I2} c_{I2}	. . .	x_{Ij} c_{Ij}	. . .	x_{IJ} c_{IJ}	A_I
	TOT	2	2		2		2	

The objective function used in the linear program can formally be stated as:

$$\text{(a)} \quad \sum_{i=1}^{I} \sum_{j=1}^{J} c_{ij} \; x_{ij}$$

Where c_{ij} is the measure of desirability assigned to the pilot 'i' in the period 'j'. The c_{ij} of the slack and artificial variables were supplied as required by the programmer.

In the restrictions which the author feels are obvious, minimum explanation will be presented.

A. Number of Pilots Assigned to Each Period.

There will be as many program restrictions as there are periods.

(b) $$\sum_{j=1}^{J} x_{ij} \leq 2 \qquad i = 1, 2, \ldots, I$$

Thus limiting the number of pilots per period to two or less.

B. Total Number of Flights During Scheduling Interval.

Two basic types of restrictions are used to satisfy this problem requirement. The first, sets the maximum number of periods to be scheduled during the time interval. The second, one restriction for each pilot, sets the minimum for the interval.

(c) $$\sum_{i=1}^{I} \sum_{j=1}^{J} x_{ij} \leq B$$

B = Maximum total flights to be scheduled x two (two pilots per period)

(d) $$\sum_{j=1}^{J} x_{ij} \geq A \qquad i = 1, 2, \ldots, I$$

A = Minimum number of flight periods per pilot

C. Minimum Number of Flights Per Month.

Since the time interval considered was one month this problem restriction was implicitly satisfied by restriction (d). For a larger problem, program restriction (d) above, could be revised to:

$$\text{(e)} \qquad \sum_{j=L}^{M} x_{ij} \geq Y \qquad i = 1, 2, \ldots, I$$

Where: L = First period of the month

M = Last period of the month

Y = Minimum number of flight periods per month

This inequality (e) would set limits for the <u>minimum</u> number of flights per month. The addition of another set of restrictions (f) below, would set the <u>maximum</u>.

$$\text{(f)} \qquad \sum_{j=L}^{M} x_{ij} \leq Z \qquad i = 1, 2, \ldots, I$$

Where: L = First period of the month

M = Last period of the month

Z = Maximum number of flights per month

A set of these for each month would satisfy this problem restriction.

D. Minimum Number of Night Flights.

The satisfying of this problem restriction could be accomplished by selection of high preference values for night periods in the objective function. This would assure night flight priority. The number available would set the <u>limit</u> for a maximum number utilized. This would not assure that every pilot would satisfy individual requirements, and thus, presents an area for further research.

E. Integer Solution.

This problem restriction was not completely satisfied in the author's preliminary research. Integer programming could be utilized, although, considering the astronomical storage problem introduced, this approach does not seem practical at this time. With the advent of the IBM 360 series,[6] this approach may be feasible.

F. Maximum Number of Daily Flights.

To satisfy this problem restriction, problem restrictions were added to insure that the summation of the daily flights be less than or equal to one.

$$(g) \quad \sum_{n=p}^{q} x_{in} \leq 1 \qquad i = 1, 2, \ldots, I$$

Where: p = First flight of the day

q = Last flight of the day

These restrictions need only be entered when the pilot has more than one flight period available. For small problems this could be done manually, but should be included in the input program.

G. Minimum Time Between Flights.

As stated in the text, this is required to insure proper crew rest. This is satisfied in conjunction with problem restriction (F). The only area where restriction (F) does not satisfy this contrition is when a night period is available for utilization.

SUMMARY

A preliminary investigation into the use of linear programming in the scheduling of attached pilots has shown the validity of the use of the linear programming approach. However, because of the size of the linear programming routine required and its unavailability at the University of Pittsburgh, research was not continued in this area. The problem definitely requires an input routine to produce the linear program input restrictions and objective functions in appropriate format before this method of scheduling can be used operationally.

Possibly, with the advent of the IBM 360 series, and the realization of input and output programs, linear programming will be a fruitful area for further research in the general scheduling problem and the specific problem of attached pilot scheduling. In the process of any further research in the area the author highly recommends reference to the Bibliography -- Abstracts from Optimal Scheduling in Educational Institutions.[4]

BIBLIOGRAPHY

1. Flight Handbook, United States Air Force Technical Publication, T.O. 1C-47-1, Department of the Air Force, March 15, 1963.

2. Flight Management, United States Air Force Manual 60-1, Department of the Air Force, July 31, 1962.

3. University of Michigan Executive System, University of Michigan, September, 1963.

4. Holzman, A. G. and Turkes, W. R. Optimal Scheduling in Educational Institutions, Co-operative Research Project N-1325, University of Pittsburgh, 1964.

5. LP/90 Usage Manual, A Comprehensive Computing System for Linear Programming, 2nd ed., C-E-I-R, Inc., Washington, D.C., July, 1963.

6. IBM System/360 System Summary, New York: International Business Machines Corporation, 1964.

REFERENCES NOT CITED

Ackoff, R. Progress in Operations Research, Vol. I, New York: John Wiley and Sons, Inc., 1961.

Business System Research Memorandum, New York: International Business Machines Corporation, not dated.

Ford, L. R., Jr. and Fulkerson, D. R. Flows in Networks, Princeton: Princeton University Press, 1962.

Giffler, B. A Schedule Algebra For Deterministic Scheduling Problems, International Business Machines Corporation Research Center, April, 1959.

Holzman, A. G. and Turksen, I. B. Boolean Isomorphic Structures in an Educational Scheduling Model, Pittsburgh: University of Pittsburgh, not dated.

Pert Fundamentals, Washington, D. C.: Pert Orientation and Training Center, not dated.

Sasieni, M., Yaspan, A. and Friedman, L. Operations Research Methods and Problems, New York: John Wiley and Sons, Inc., 1961.

Vajda, S. Mathematical Programming, London: Addison-Wesley Publishing Company, Inc., 1961.