This dissertation has been 64-13,351
microfilmed exactly as received

LINDSEY, Joseph Ferguson, 1924-
CONTROL LIMITS FOR OPTIMUM JOB SHOP
SCHEDULING.

The University of Oklahoma, Ph.D., 1964
Engineering, general

University Microfilms, Inc., Ann Arbor, Michigan

THE UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

CONTROL LIMITS FOR OPTIMUM JOB SHOP SCHEDULING

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

DOCTOR OF PHILOSOPHY

BY

JOSEPH FERGUSON LINDSEY

Norman, Oklahoma

1964

CONTROL LIMITS FOR OPTIMUM JOB SHOP SCHEDULING

APPROVED BY

Charles L. Proctor

C. M. Sliepcevich

John E. Brixey

Joseph R. Assenzo

Richard A. Terry

DISSERTATION COMMITTEE

ACKNOWLEDGEMENT

The author takes this opportunity to express his appreciation to Dr. C. M. Sliepcevich, who has directed his course of study at the University of Oklahoma, for his guidance, encouragement and inspiration.

The author would like to thank all the members of his committee for their help and particularly, Dr. C. L. Proctor, who carried the major burden in directing this dissertation.

The author further wishes to thank his departmental Chairman, Professor H. L. Manning, Virginia Polytechnic Institute, for using departmental grants from the General Electric Corporation for the support of his graduate program.

Sincere thanks are due to Mr. Charles E. Maudlin, and the University of Oklahoma Computer Laboratory Staff for their assistance in "debugging" programs and in scheduling computer time.

Lastly, the author wishes to express his gratefulness to his wife, Clare, and their four sons; for without their cooperation and sacrifices, further graduate study would have been impossible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

Figure | Page

CONTROL LIMITS FOR OPTIMUM JOB SHOP SCHEDULING

CHAPTER I

INTRODUCTION

The development of the simplex method of linear programming and the introduction of digital computers after World War II have contributed immeasurably to the research that has been done on the 'scheduling problem.' The 'scheduling problem' refers to the determination of an optimum schedule for the job shop type of operation in which each part requires a number of operations to be performed by several machines with certain sequential limitations. Although the problem appears to be rather trivial, no reasonable, economical, exact method of solution has appeared at this time due to the combinatorial aspects of the problem.

Various assumptions have generally been made to facilitate mathematical development of solutions, and in most cases these restrictions require that a machine does only one operation at a time, that not more than one operation be performed on a specific part by the same machine, and that all parts require operations by all machines. With these restrictions the total number of possible schedules which would require calculation in order to find the minimum overall time (the normal optimizing objective) is $(n!)^m$, where n equals the number of parts and m equals the number of machines. For a simple 4 machine by 8 part schedule $(8!)^4$ yields more than 2.63×10^{18} combinations. This represents a considerable work load for the fastest of computers. If each schedule could be derived in a nano-second (10^{-9}), it would take one computer, working 24 hours per day, 85 years to complete the enumeration. By comparison, a set of four linear equations in eight unknowns has only $\frac{8!}{4!4!} = 70$ possible solutions, and it should be remembered that linear programming did not become of interest until Dantzig was able to find a method which would reduce the order of magnitude of this combinatorial problem.

In order to show the approaches that previous researchers have developed so that the magnitude of the problem may be appreciated, and so that indirect contributions of the past may be gratefully acknowledged, a chronological history is included in the next chapter. The approaches to solving this problem run the gauntlet from non-mathematical simulation to the development of new mathematics. Most of the researchers have approached the 'scheduling problem' from a mathematical point of view, and although the problem is much better formulated than before, it is still an unresolved mathematical problem. Industrial engineers and production control personnel have been empirically solving a similar problem for many years and may continue to do so for many more before mathematics develops a feasible, computable, exact solution. It is the writer's premise that we have enough mathematical techniques at this time to assist the personnel responsible for scheduling in optimizing their operations.

This dissertation is the result of research on the 'scheduling problem' in which "empirical" parameters have been established which can be used both as control limits and as additional constraints in linear programming solutions of related scheduling problems.

CHAPTER II

ORIGIN OF SCHEDULING ALGORITHMS

A simple example of 2 machines and 3 parts which gives rise to $(3!)^2 = 36$ possible schedules will be used to illustrate some of the previous researchers' approaches to solving the 'scheduling problem.'

Let the parts be designated A, B, and C and the machines 1 and 2. Job A_1 will mean that part A is being processed on machine 1 and so on. Since several algorithms are only applicable for the cases when all jobs are processed in the same order, the following matrix which consists of times to perform the operations represented by the row-column indices assumes that a sequential relationship X_1 precedes X_2 for all parts.

$$\begin{array}{lc} & \begin{array}{cc} \text{Mach. 1} & \text{Mach. 2} \end{array} \\ \begin{array}{l} \text{Part A} \\ \text{Part B} \\ \text{Part C} \end{array} & \begin{bmatrix} 4 & 6 \\ 5 & 3 \\ 7 & 6 \end{bmatrix} \end{array} = T_{ij}$$

First, all 36 schedules will be shown from a precedence relationship, and then each schedule will be evaluated to determine the total time of the minimum schedule. Many of the schedules are obviously non-optimum and any successful algorithm must be able to eliminate them. With the limitation of scheduling all parts in the same order, only schedules designated with * need to be calculated.

	(1)*	(2)	(3)	(4)	(5)	(6)	(7)	(8)*	(9)	(10)
Mach. 1	ABC	ACB	BAC	BCA	CAB	CBA	ABC	ACB	BAC	BCA
Mach. 2	ABC	ABC	ABC	ABC	ABC	ABC	ACB	ACB	ACB	ACB
	(11)	(12)	(13)	(14)	(15)*	(16)	(17)	(18)	(19)	(20)
Mach. 1	CAB	CBA	ABC	ACB	BAC	BCA	CAB	CBA	ABC	ACB
Mach. 2	ACB	ACB	BAC	BAC	BAC	BAC	BAC	BAC	BCA	BCA
	(21)	(22)*	(23)	(24)	(25)	(26)	(27)	(28)	(29)*	(30)
Mach. 1	BAC	BCA	CAB	CBA	ABC	ACB	BAC	BCA	CAB	CBA
Mach. 2	BCA	BCA	BCA	BCA	CAB	CAB	CAB	CAB	CAB	CAB
	(31)	(32)	(33)	(34)	(35)	(36)*				
Mach. 1	ABC	ACB	BAC	BCA	CAB	CBA				
Mach. 2	CBA	CBA	CBA	CBA	CBA	CBA				

By substituting in the time values from the T_{ij} matrix completion times for each schedule may be calculated as follows: For schedule (1)

Mach. 1 $\quad 4 + 5 + 7 = 16$

Mach. 2 $\quad *4 + 6 + 3 + 3* + 6 = 22$

Max = 22

* Designates idle time that Mach. 2 waits for Mach. 1 for its next part.

Values for all 36 schedules are given below:

Sched.	Time	Sched.	Time	Sched.	Time	Sched.	Time
1.	22	10.	31	19.	28	28.	27
2.	27	11.	26	20.	31	29.	22
3.	24	12.	31	21.	28	30.	22

Sched.	Time	Sched.	Time	Sched.	Time	Sched.	Time
4.	31	13.	24	22.	24	31.	31
5.	26	14.	31	23.	31	32.	26
6.	31	15.	22	24.	27	33.	31
7.	25	16.	28	25.	31	34.	27
8.	20	17.	31	26.	26	35.	25
9.	25	18.	28	27.	31	36.	22

A summary of the results reveals a point, which has been substantiated in large scale simulations of larger schedules by Heller (1), that there are considerably fewer different times than there are schedules.

Time	Frequency
20	1
22	5
24	3
25	3
26	4
27	4
28	4
31	12
	36

One of the earliest algorithms to find the schedule which takes the least time was developed by S. M. Johnson (2). The solution is very restrictive as it is limited to two machines, and all parts must be processed in the same order, the second operation on the part immediately following the first. The steps in the algorithm are:

1. Select the smallest time required to perform an operation.
2. If the smallest time is on machine 1, schedule it first; if the smallest time is on machine 2, schedule it last.
3. Delete jobs already scheduled and repeat steps 1 and 2 until all jobs are scheduled.

From the sample problem:

	Mach. 1	Mach. 2
Part A	4	6
Part B	5	3
Part C	7	6

Since 3 is the smallest time and is on machine 2, schedule Part B last. 4 is the next smallest and on machine 1, therefore schedule Part A first. Part C is then scheduled in between. Completion of the algorithm yields a schedule

Machine 1	ACB	$T_1 = 4 + 7 + 5 = 16$
Machine 2	ACB	$T_2 = {*4} + 6 + {*1} + 6 + 3 = 20$

The above solution agrees with the solution found by evaluating all possible schedules, and it certainly is much more easily performed. Although this method is a great improvement over trial and error methods, it is too restrictive, and this type of algorithm can not be extended to a more general case without making unrealistic assumptions.

In addition to Johnson, L. G. Mitten (3) has also developed the mathematics for sequencing n parts on two machines. While the results of these researchers are not very useful, directly, their beginning approaches may well have been the basis or impetus for the later work which has been done in this field.

The next technique to be presented was the processing of 2 jobs on N machines. Here also, each job was processed by each machine although the sequences for the products need not be the same. This method of Akers and Friedman (4) establishes a set of 2^N possible schedules. This set is without regard to the technological orderings of the sequences that each part must follow, and the second step eliminates these unsatisfactory schedules. From the remaining group there are certain schedules which are obviously non-optimum and they are eliminated by a series of rules. The final sub-set of schedules is evaluated by Gantt Chart methods to determine the shortest processing time. Again, this particular technique is limited, but it marks progress as it uses precedence relationships and inductive logic to reduce the combinatorial problem to a manageable set of schedules.

Since the combinatorial problem was so formidable, some researchers, notably A. J Rowe, (5), J. R. Jackson (6) and R. L. Sisson (7) developed simulation models of the production process. The objective of the simulation models was to gain insight of the effect of various rules for determining priority of scheduling each machine. Later, Conway, Johnson and Maxwell (8) made further investigations into priority dispatching rules. The results of these investigations have been helpful in identifying conditions which mathematical models must include and the results also provide a basis for decision in the later sampling experiments of Giffler and Thompson (9) and Heller (1). The simulation models are not applicable to operating conditions but are research tools which help to develop better heuristic scheduling rules.

While dynamic programming and integer linear programming methods have been applied to the scheduling problem by Bellman (10), Story and Wagner (11, Chapter 14), the larger problems are not computationally feasible. Wagner and Story are hopeful that the integer linear programming method can be modified to more rapidly approach the optimum solution, and when this is accomplished a possible solution to the problem may exist.

Several researchers have made unusual approaches to the 'scheduling problem,' but in spite of rather elegant mathematics they do not appear to offer workable algorithms. Salveson (12) has proposed a variation of the Gantt chart method and Reinitz (11, Chapter 5) has applied Markov theory and dynamic programming to this problem. Giffler (11, Chapters 3 and 4) has made two significant contributions in this area. The first is in the development of a mathematical theory of scheduling which should help the mathematicians working on the problem from a theoretical point of view. Secondly, using a Gan t chart approach, Giffler has developed an algorithm which will generate feasible schedules. By eliminating schedules which are known to be non-feasible, the combinatorial set of all possible schedules is significantly reduced. The algorithm in itself does not select an optimum schedule, but only a set of schedules which will contain the optimum. Sampling is then required to find a pseudo-optimum[1] schedule since it is still impossible to calculate all of the subset of schedules. Recently, (1964) Hardgrave and Nemhauser (13) developed a proof for a graphical method of determining the span of a schedule. They illustrate the method for two and three dimensions (one dimension for each part) and generalize their proof for n dimensional space. The two dimensional method was previously reported in Saiseni et al.(14) without proof. A computer method to solve the n dimensional case would be similar to Giffler's algorithm and since it also expands factorially it is not practical for calculating the minimum schedule.

Working from a different approach, Heller (15) has devised an algorithm which also generates schedules. By use of a sampling technique it is also possible to generate a pseudo-optimum schedule. Since the algorithm of Heller was easier to program and produced schedules faster than Gifflers, it is the method that this researcher has used in gathering data for the statistical parameters used as control limits. Although Giffler's algorithm could have been used, Heller indicated that his experiments were more economical of

---

[1]Pseudo-optimum is used to describe the schedule with minimum time found in a sampling of possible schedules. It may be the true minimum but the probability is small for any large problem.

computer time and for that reason were followed. A full discussion of Heller's method is described in the next chapter.

CHAPTER III

THE HELLER SCHEDULING ALGORITHM

The fact that a practical algorithm for solving sequencing problems had not been developed (1959) led Heller (16) to approach the problem from a combination of sampling and simulation techniques. The procedure to follow in determining pseudo-optimum schedules is to generate a number of possible schedules and then to take the one with the minimum time. After developing an algorithm to generate schedules, Heller made many experimental computer runs which led to a better understanding of the 'scheduling problem.' Those results which were useful in developing the computer program used to generate data for this dissertation will be covered at the end of this chapter.

The algorithm which was developed by Heller is a significant contribution to this area and it has its theoretical basis in linear graph theory. The explanation in this chapter is a simplified version of Heller's algorithm; for a complete development, see (16, 17).

The example of chapter 2 will be used to illustrate the algorithm. The modification of the algorithm does not require a fixed sequencing order for the parts and in the original development, a part may return to the same machine for subsequent operations. A three digit code is used to designate each operation. The first digit represents the machine, the second represents the part and the third designates the operation sequence number for the part. If A, B, C, of the example is replaced by 1,2,3, the number (112) represents the operation performed on part 1 by machine 1 and it is the second operation performed on the part. Figure 1 shows the three example jobs with times indicated next to the circles.

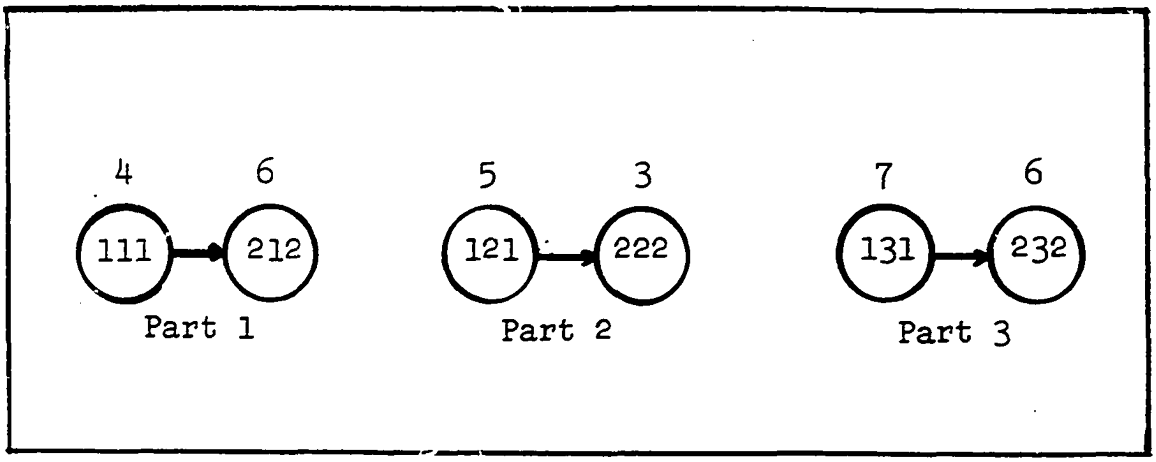


Fig. 1. Simple linear graphs

A more complex job is shown in Figure 2 and it represents the case where sequence is not completely restricted and where a machine can perform more than one operation on a part. This example is more general than the model used to generate data for this dissertation. In Figure 2 operations (111) and (212) are independent and can be performed at any time. Operation (313) cannot begin until both operation (212) and (111) are complete. Operation (214) can be done after

(111) is complete and (415) cannot begin until both (313) and (214) are finished. Operation (116) follows completion of (415). Note that operations 1 and 6 are on machine 1 while operations 2 and 4 are on machine 2.

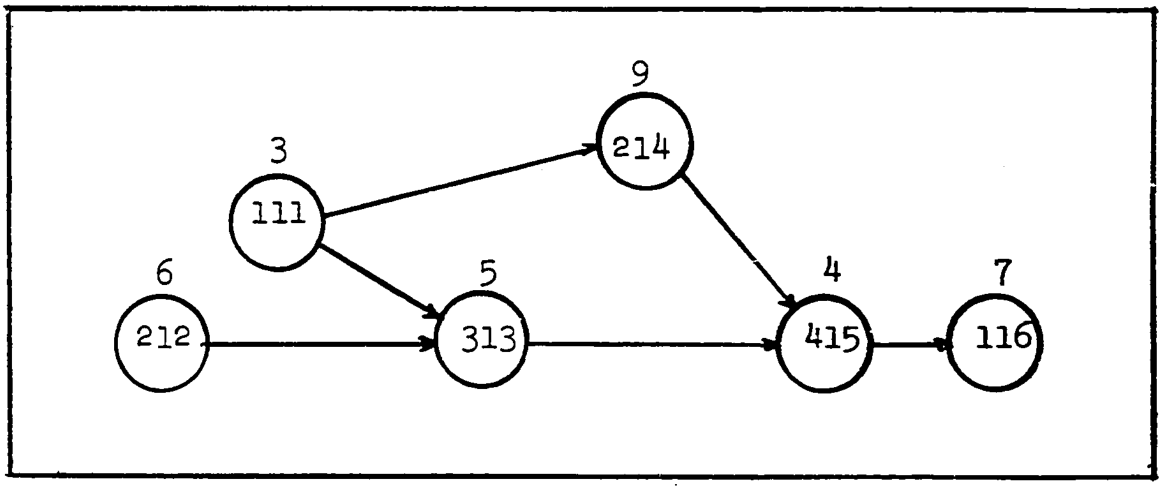


Fig. 2. General linear graph

In terms of graph theory, the directed branches of these linear graphs will indicate precedence relations of the jobs through the machines and the circled operations numbers will be designated as nodes. In figure 1, operation (111) is followed (covered in graph theory terminology) by node (212) or (111) $\longrightarrow$ (212). Figures 1 and 2 indicate the technological orderings of the various jobs (parts). The designation of an operation by Heller is slightly different than the one used in this paper and the apparent redundancy of including the operation number in the operation designation is compensated for by the facility of handling the data on the computer.

First, let us develop the method for determining the time required to produce a part such as the one shown in figure 2. If we call t_{mjo} the processing time of node (mjo), where m designates the machine number, j designates the job (part) number and o designates operation number, then the total time required up to and including node (mjo) is given by $T(mjo) = \max_{(m'j'o') \rightarrow (mjo)} T(m'j'o') + t_{mjo}$ where the primes indicate the nodes and time up to and including previous node (i.e. all nodes which cover node (mjo)). Unless otherwise stated, we will assume that all operations are processed as soon as ordering constraints are met. Since, in figure 2, operation (111) and (212) do not cover (follow) any other node, we will give them a starting time of zero. Consequently $T(111) = t_{111} = 3$ and $T(212) = t_{212} = 6$.

$$T(313) = \max \left\{T(111),\ T(212)\right\} + t_{313}$$
$$T(313) = T(212) + t_{313} = 6 + 5 = 11$$
$$T(214) = T(111) + t_{214} = 3 + 9 = 12$$
$$T(415) = \max \left\{T(214),\ T(313)\right\} + t_{415}$$
$$T(415) = T(313) + t_{415} = 12 + 4 = 16$$
$$T(116) = T(415) + t_{116} = 16 + 7 = 23$$

In the above calculations we only considered the technological orderings of each part, and it is possible that the schedule generated is not feasible due to machine interference. In figure 3 a schedule graph of the part in figure 2 is shown. While the time values generated for the schedule in figure 3 take into account the coverings of the nodes, they have not considered the availability of the machines and $T(214) \neq T(111) + t_{214}$ since machine 2 is still operating on (212) when (111) is completed. Hence, in order to include both the technological ordering and the elimination of simultaneous scheduling of more than one part on the same machine, an additional constraint is added to the solution.

$T(mjo) = \max_{(m'j'o') \longrightarrow (mjo)} \left\{ T(m'j'o'),\ T(\sum m_i') \right\} + t_{mjo}$ where $T(\sum m_i')$ equals the sum of all operating time previously assigned to machine m_i.

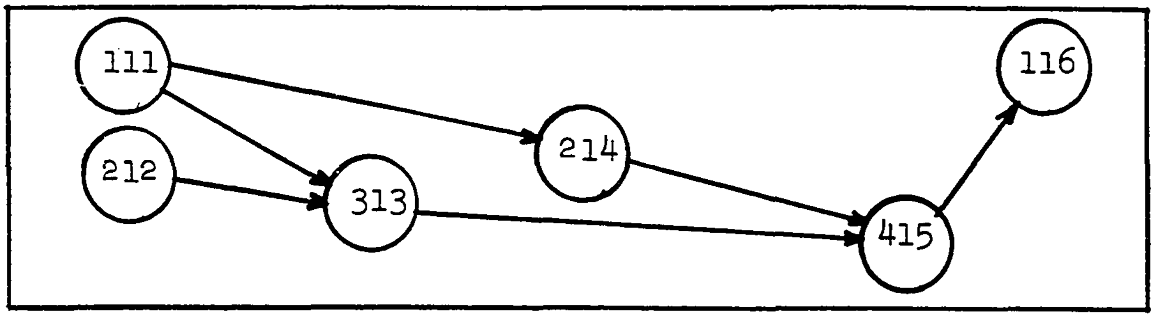


Fig. 3. Schedule graph of Fig. 2.

Since the example of Chapter 2 will be used to demonstrate the algorithm, one possible schedule is shown in figure 4.

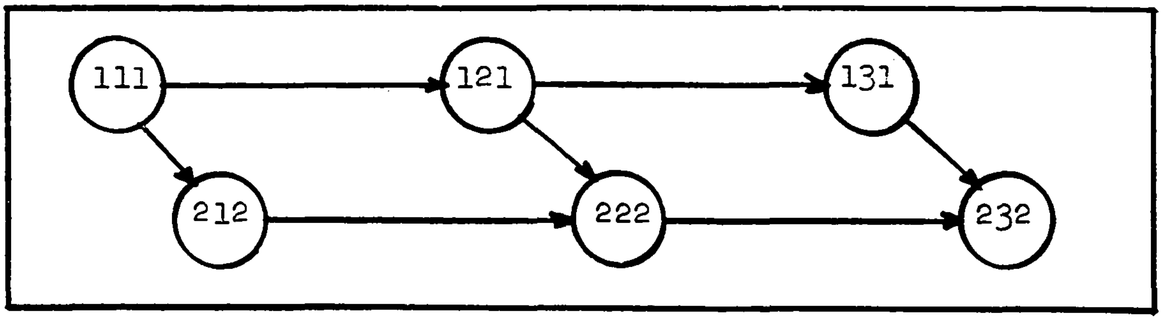


Fig. 4. Schedule graph of Fig. 1.

Neither Gannt charts nor linear graphs lend themselves to computational methods by a digital computer, therefore some method must be developed to transform the information on a graph into information usable by the digital computer. The algorithm developed by Heller makes use of the notion of a "table of coverings" which is a list that includes all the necessary precedence relations. Figure 5 is a "table of coverings" of the example problem. Each node requires one line in the table, and the columns are as follows:

Column 1:	node designation (mjo)	C_1
Column 2:	nodes covering given node in technological ordering ($m^o j o^o$)	C_2
Column 3:	number of nodes covered by (mjo) in item 2	C_3
Column 4:	processing time, t_{mjo}	C_4
Column 5:	starting time of node (mjo)	C_5
Column 6:	finish time of node (mjo)	C_6

Although the algorithm can be followed with each line in an unordered sequence, computer programming is greatly facilitated when the list is ordered on Column C_1 by machine number and by part number as indicated in figure 5. Note that the linear graphs of technological orderings, figure 1, can be reproduced from the list. For example: node (111) of C_1 is followed by (211) of C_2, and no operation precedes node (111) since $C_3 = 0$. Node (212) of C_1 is the last operation on part 1 since $C_2 = 0$, but one operation precedes (212) as indicated by $C_3 = 1$.

The first step in the algorithm is to select the operation which will be performed first and to designate the initial start time. Zero is a convenient start time and for a first operation one can choose any node which does not cover another. A zero in C_3 indicates a possible starting place. In the example there are three zeros in C_3 at time zero and any random method or any priority scheduling rule may be used to determine the start. In a more generalized problem, there may be at least one zero for most machines and some method of selecting order of machines should also be determined to allow for all possible feasible schedules to have equal probability of selection.

C_1	C_2	C_3	C_4	C_5	C_6
111	211	0	4		
121	222	0	5		
131	232	0	7		
212	0	1	6		
222	0	1	3		
232	0	1	6		

Fig. 5. Initial table of coverings

With the selection of the initial operation node, the start time (equals zero) is placed in C_5 and $T(mjo) = t_{mjo}$ is placed in column C_6. To denote that the node has been already scheduled, C_3 is coded, a -1 being one designation which is convenient for computer operations. Next, Column C_2 is checked for subsequent operation. If it is zero, there is no following operation. If there is a node designation, the line containing the node in C_1 is altered as follows: The number of preceding operations, which is the content of C_3, is reduced by 1 unit. (In this research project strict sequencing was followed and C_3 initially is either 1 or 0.) When the value of C_3 is changed to zero the node is ready for scheduling. Next the start column C_5 is checked for any technological ordering requirements. If the value of C_5 for (mjo) is less than the value of C_6 for (m'jo'), where the primes designate preceding nodes already scheduled, a tentative start time equal to the value in C_6 of (m'jo') is placed in C_5 of line (mjo).

The results of the above operations are indicated in Figure 6 where node (111) was chosen as the starting point of the schedule.

C_1	C_2	C_3	C_4	C_5	C_6
111	212	$\cancel{0}$ -1	4	0	4
121	222	0	5		
131	232	0	7		
212	0	$\cancel{1}$ 0	6	4	
222	0	1	3		
232	0	1	6		

Fig. 6. Table of coverings after first cycle

The scheduling algorithm only guarantees generating feasible schedules without regard to optimization of time. Therefore the second node to be scheduled may be any node which has $C_3 = 0$. Experience with this algorithm and with pure simulation indicate priority rules which should be followed if overall schedule time is to be minimized. In this simple example we required operation 2 to immediately follow operation 1, so we must find a node (mjo) where j = j' and o = o' + 1. (212) is such a node and C_3 = 0 so we will schedule it next. Since this is the first operation on machine 2, we need not be concerned with the machine load and the time for the finish of (212) will equal $C_5 + C_4$ or 6 + 4 = 10. The value in C_3 of line (212) is reduced by 1 to -1. Since C_2 equals zero, there is no operation covering (212). Figure 7 shows the results of scheduling node (212).

C_1	C_2	C_3	C_4	C_5	C_6
111	212	$\cancel{0}$ -1	4	0	4
121	222	0	5		
131	232	0	7		
212	0	$\cancel{1}$ $\cancel{0}$ -1	6	4	10
222	0	1	3		
232	0	1	6		

Fig. 7. Table of coverings after node (212) scheduled

For the next step we will choose node (121). Since machine 1 has been scheduled, it cannot be used for another operation until time equalling maximum T(mj'o'), C_6, has elapsed. This is 4 and therefore the start of (121) is 4 which is placed in C_5; T(mjo) = $C_4 + C_5$ = 9 which is placed in C_6. C_2 indicates that (222) follows and the 1 in C_3 must be reduced by 1. The finish time of (121) is placed in C_5 of node (222) if it is greater than the value already there. The value in C_3 is also reduced by 1. The start value of 9 indicates when the previous operations on part 2 will be complete. However, the 10 in C_6 of line (212) indicates that machine 2 will not be released until time 10, so the start time is set equal to 10, the maximum of the technological ordering and machine availability. $T(212) = C_4 + C_5 = 13$, which is placed in C_6. Figure 8 shows the result of the above scheduling and figure 9 shows a completed schedule.

C_1	C_2	C_3	C_4	C_5	C_6
111	212	~~0~~ -1	4	0	4
121	222	~~0~~ -1	5	4	9
131	232	0	7		
212	0	~~1~~ ~~0~~ -1	6	4	10
222	0	~~1~~ ~~0~~ -1	3	~~9~~ 10	13
232	0	1	6		

Fig. 8. Table of coverings after node (222) scheduled

The sample schedule generated by the algorithm is not the optimum as 20 was determined to be minimum time. With the restrictions placed on the sample, there are only six possible schedules and all six could be evaluated and the minimum selected. With large problems the number of possible schedules becomes too large for complete enumeration and a sampling scheme is required (18).

C_1	C_2	C_3	C_4	C_5	C_6
111	212	$\cancel{0}$ -1	4	0	4
121	222	$\cancel{0}$ -1	5	4	9
131	232	$\cancel{0}$ -1	7	9	16
212	0	$\cancel{1}$ $\cancel{0}$ -1	6	4	10
222	0	$\cancel{1}$ $\cancel{0}$ -1	3	$\cancel{9}$ 10	13
232	0	$\cancel{1}$ $\cancel{0}$ -1	6	16	22

Fig. 9. Table of coverings for completed schedule

A restatement of the steps in the algorithm is as follows:

1) Select one of the nodes $(m^o j^o o^o)$ which has $C_3 = 0$.
2) Let $C_5 = \max \left\{ \max_{j,o} C_6\ (m^o jo),\ C_5\ (m^o j^o o^o) \right\}$
3) Set $C_3(m^o j^o o^o) = -1$
4) Compute $C_6(m^o j^o o^o) = C_4(m^o j^o o^o) + C_5(m^o j^o o^o)$
5) If node $C_2(mj^o o) = 0$, replace $C_3(mj^o o)$ by $C_3(mj^o o)-1$ and replace $C_5(mj^o o)$ by $\max \left\{ C_5(mj^o o),\ C_6(m^o j^o o^o) \right\}$
6) Repeat steps 1 to 5 until all entries in C_3 are -1.

Heller has made a large number of computer runs with this algorithm and has found that best results are obtained when jobs are scheduled on a "first come first served basis." His results showed that sample minimum schedule time was less with this rule, was reached with a smaller sample and had the smallest variance. Therefore, in the computer program used to generate data for this report, parts were scheduled to machines in random order and the part which had arrived the earliest (smallest C_5 value), was scheduled first. In case of ties, a pseudo-random number generator routine was used to select which one of the smallest would go first. A flow chart and a Fortran listing of the program are shown in the Appendix.

CHAPTER IV

THE CONTROL LIMIT HYPOTHESIS

A review of the mathematical complexities involved in the 'scheduling problem' indicates that any eventual satisfactory solution will still be expensive to program on the computer. The question that immediately comes to mind is, how practical would such a method be in the normal industrial world? Does industrial engineering and production control require such an elaborate procedure, or how would they use it if they had it today? Of course, our industry is made up of many independent segments, and while some might use the algorithm, it would appear to be too refined for present day scheduling problems.

In a very humorous introductory chapter to Industrial Scheduling (11), William F. Pounds reveals the difficulties that he had in trying to discover the 'scheduling problem' in industry. The scheduling personnel knew of no problem!

There are several problems that play havoc with the scheduling procedures and which would certainly place so many restrictions on a perfected optimum scheduling algorithm that the refinements would be lost and the presumed additional expenses unjustified. Some of the items which cause most of the problems in a scheduling environment are: (1) lack of raw materials at time part should be scheduled, (2) lack of tooling, (3) machine breakdowns, (4) sales pressure with priority orders, (5) delaying engineering changes, and (6) lack of man power.

There are already companies that use computers for scheduling day to day production as well as for making general overall plans. These scheduling programs take into account the most recent information regarding raw materials, tooling, man power, sales priorities, engineering changes, and production assembly work load. (19, 20) These programs approach optimum for the conditions and time cycle under which they perform. They certainly cannot optimize to the degree that a deterministic optimized algorithm can, but the production process is dynamic and not deterministic, and adding a stochastic environment to the visualized optimizing algorithm again limits the feasibility of computability.

The question once more arises, what would management like and/or use along the line of optimum scheduling? There may be as many answers as there are managements, but this researcher has set out to establish an optimizing procedure which many managements could use, which does not require a large scale computer, and which offers flexibility. This procedure is based on the familiar control limit concepts which are used in a variety of applications by almost all forms of industrial enterprise.

The parameters developed for the control limits can be utilized for three purposes. First, they can be used in pre-planning since they can give the time interval for which various schedules should be completed if performed in a near-optimum time. Secondly, they can be used as a performance rating of completed schedules. Thirdly, they can be used as constraints in preliminary planning with linear programming optimization.

In the research conducted, the schedules were all based on products which had normally distributed operation times with mean of 50 and variance of 100 (21). In order to eliminate further bias in the selection of schedules to be used, a computer program was written to generate the input data for the scheduling runs. A pseudo-random number generator routine was used to assign operation numbers to each node and to assign times for each operation. A flow diagram and a Fortran listing of the program are included in the Appendix.

Unless further research and/or experience with control limits prove otherwise, each company making use of this technique will analyse its production mix to establish parameters. Once the company parameters have been established, the control limits may be readily calculated from the data normally available for preparing schedules, i.e., number of parts, number of operations, number of machines, and hours per part, per operation and per machine. Cost is excluded from this control limit since it is assumed that the initial assignment of machines for the various operations has been made by optimization of a cost function through linear programming techniques.

A common present day method of evaluating industrial performance is to calculate percent machine utilization and percent of standard performance. The calculations give management an indication of how well a production unit is producing without any indication of losses due to scheduling interferences. While most planning and some evaluation take sequencing delays into account in the form of loading percentages, there has been little work done to provide more exacting criteria for management evaluation of operations. With todays utilization of computers for production planning and control, more exacting methods are required for optimization.

The simulation experiments performed for this report indicate that strong linearity exists between normal scheduling input parameters and pseudo-minimum schedule time. It is anticipated that the industrial data will provide a similar relationship. By analysing the industrial data in the manner explained in Chapter VI, an estimating function can be developed which will establish a limit for the overall shop activity based on input parameters.

CHAPTER V

ANALYSIS OF EXPERIMENTAL DATA

The experiment consisted of using a modification of the Heller scheduling algorithm (see Appendix A) to determine the minimum schedule for a large number of m x n schedules. A formal statistical design was not utilized since the available computer was uneconomical for this type of simulation. As a consequence, the experiment could not be replicated, could not extend to as large schedules as desired and the number of samples to determine a pseudo-minimum was less than the optimum previously determined by both Heller (1) and Giffler, *et al*. (11, Chapter 3).

Figure 10 shows the number and size of the schedules which were generated. The initial plans to keep within a 10 x 10 were expanded so that more data would be available at the higher end of the scale. The experiment was designed to verify a hypothesis, namely that there is a pattern in the set of minimum schedules which could provide a method of estimation of the expected minimum schedule which would be simpler to calculate than using a scheduling algorithm. For this reason a series of 2 x n schedules were generated using the input data program (see Appendix A) and times from a normal distribution in hopes of getting a relationship dependent upon number of parts with number of machines fixed.

Number of Parts	Number of Machines								
	2	3	4	5	6	7	8	9	10
4	x								
5	x	x		x					
6	x								
7	x	x	x	x	x	x			
8	x	x	x				x		
9	x	x	x	x	x	x	x	x	
10	x	x	x	x	x	x	x	x	x
11						x	x	x	
12						x	x		

Fig. 10. Design of Experiment

Similarly, the schedules for m x 10 operations were generated to determine a relationship with number of parts fixed and number of machines varying. Other combinations were used within the 10 x 10 limitations but those cases for number of machines greater than number of operations were omitted as well as any combination giving the same number of total operations as previously generated starting with 2 machines and working upward. The purpose of getting the variety of operations was to have more data available in case number of operations was significant as an estimator.

There is little justification for simulating both m x n and n x m schedules since the algorithm uses accumulation of time by parts and by machines and the difference in the minimum time for the normalized data would be negligible. An attempt to prove that the expected minimum time would be the same was unsuccessful as several examples proved otherwise due to the first come first served priority rule which was incorporated in the algorithm.

The analysis of data program (see Appendix A) made several determinations concerning the input data. The maximum time that all parts were on one machine and the maximum time that all parts took on one machine were calculated for each schedule. In addition, a weighting factor which measured variation of operation numbers on each machine, the number of parts to number of machines ratio, the average machine time, the average part time and variances of machine and part time were also calculated.

Although maximum part time and maximum machine time versus minimum schedule time did not individually correlate, graphically, the sum of maximum part time and maximum machine time did show a strong linear relationship as demonstrated in Figure 11. Figures 12 and 13 show the strong linearity of minimum schedule times versus fixed number of machines and for a fixed number of parts, respectively. Minimum schedules versus ratio of parts to machines is shown in Figure 14 to possess linearity, also.

In addition to the linearity already demonstrated,

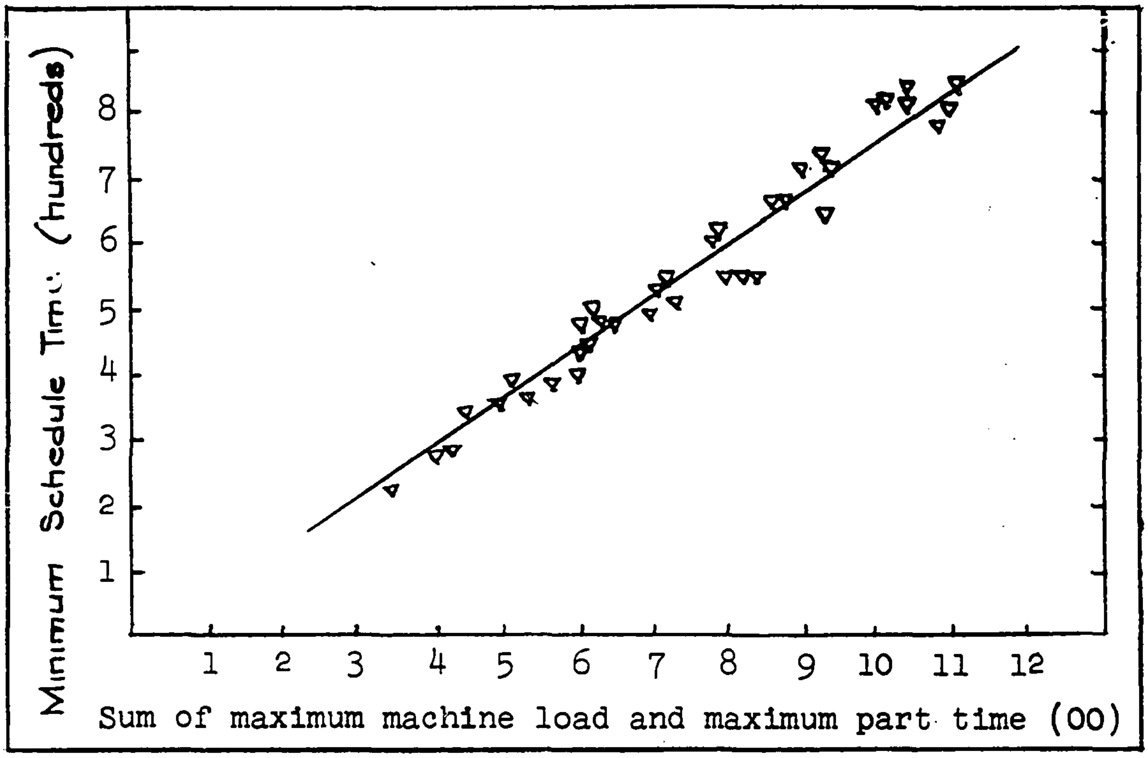


Fig. 11. Minimum schedule time versus sum of maximum machine load and maximum part time

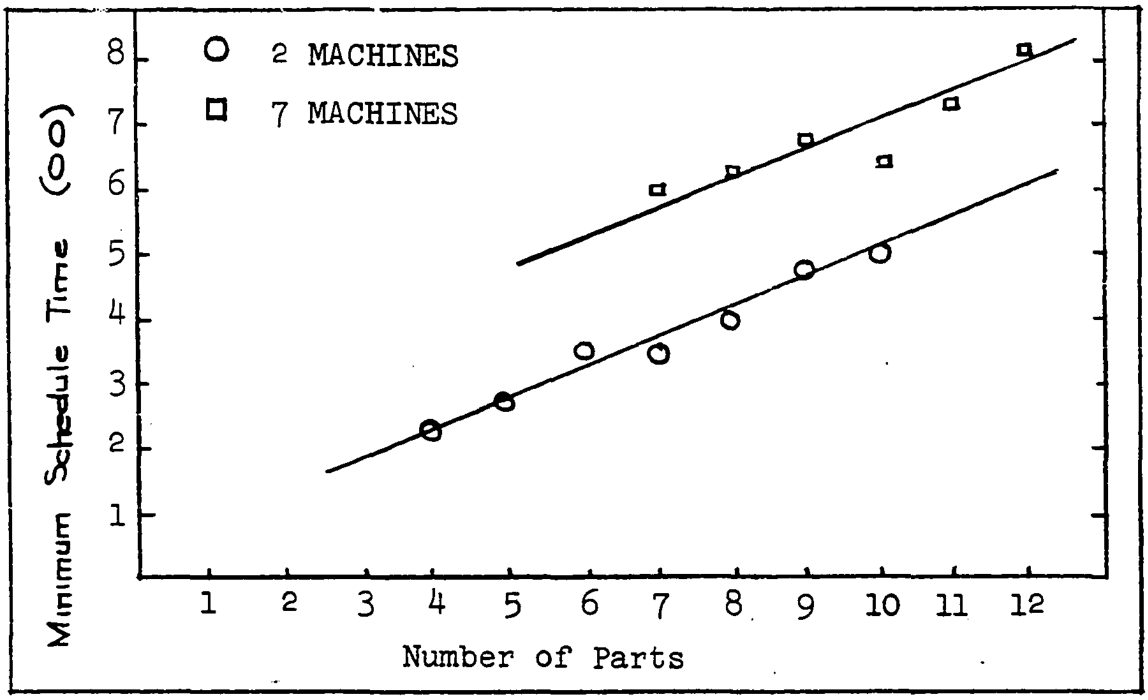


Fig. 12. Minimum schedule time versus number of parts for fixed number of machines

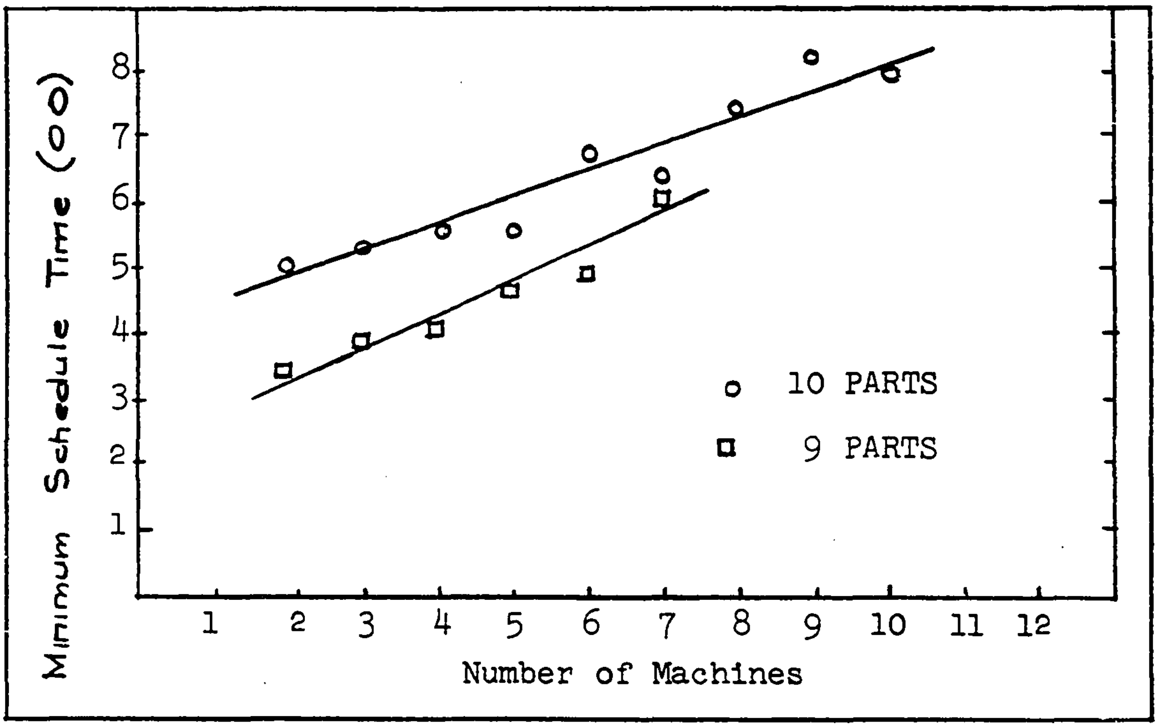


Fig. 13. Minimum schedule time versus number of machines for fixed number of parts

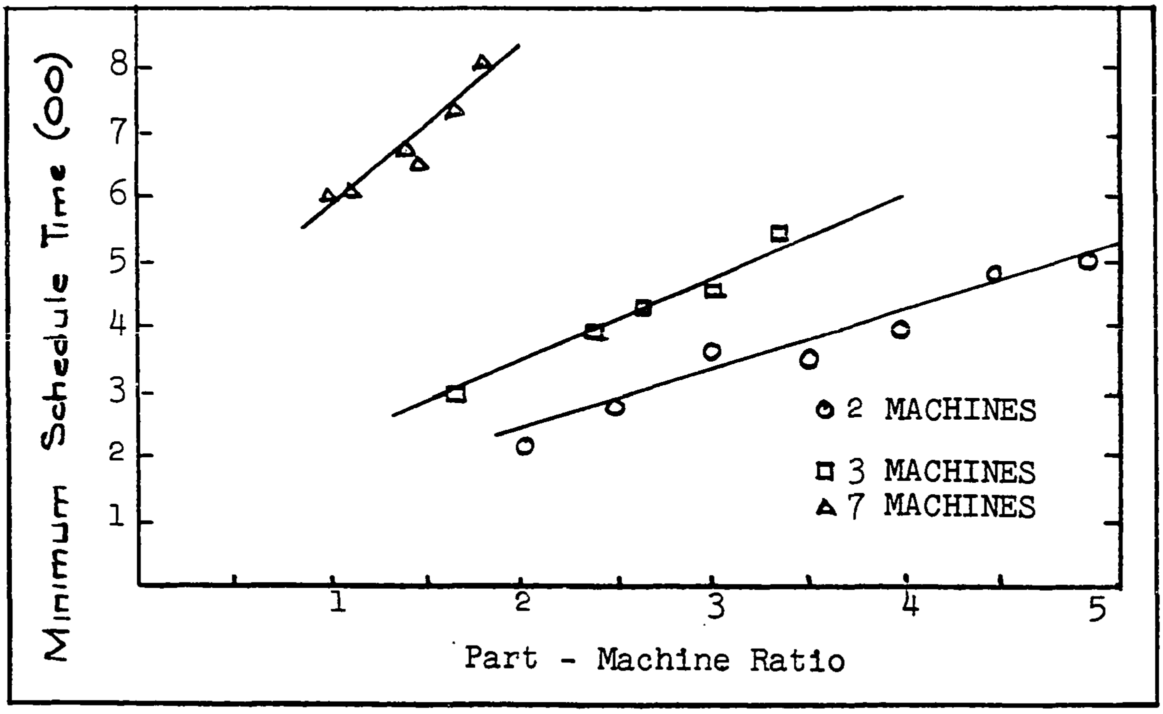


Fig. 14. Minimum Schedule Time versus Part - Machine Ratio for Various Number of Machines

there is also a strong linear relationship between the number of operations and the minimum schedule time, as shown in Figure 15. Although this relationship could probably have provided a statistically valid estimating equation, it can be readily deduced that it cannot be extended since a 3 x 10 schedule should give different results than a 5 x 6 schedule. The first schedule would be a function of 10 parts while the second would be a function of 6 and the expected sum of 10 random variables will seldom equal the sum of 6 random variables taken from the same distribution.

Since there were so many linear relationships, it was decided a good estimating equation could probably be obtained by making use of multiple linear regression.

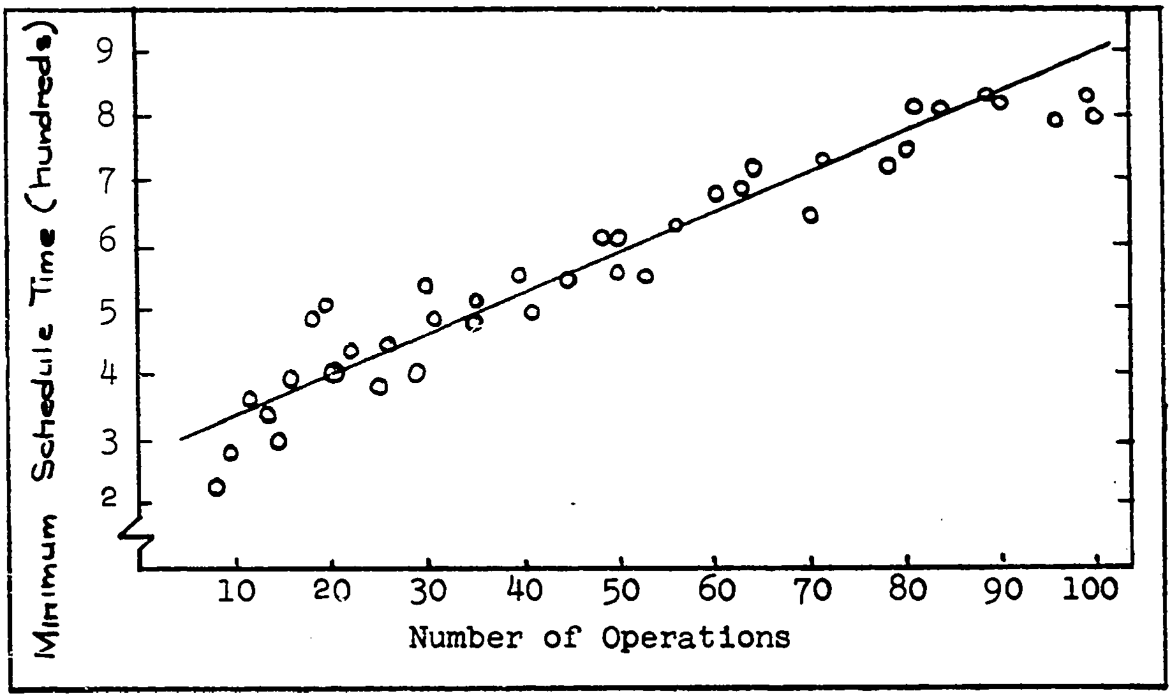


Fig. 15. Minimum Schedule Time Versus Number of Operations

Several computer runs were made with a Doolittle multiple linear regression program (Appendix A), using all 10 "independent" variables of Table 1. The initial runs indicated that several of the variables were not very important since the standardized beta was negligible and only two items show significance for the "t" test. (See Table 2) (Kramer, 23) An estimating function using only variables $\overline{P}$, P_{var}, and Ratio was developed and when P_{var} was rejected by the "t" test, a 2 variable program was run which gave the following estimating equation:

$$Y = -131.48 + 1.9426\, \overline{P} + 94.7096\, R \tag{5.1}$$

This regression line accounted for 93.6 percent of the variance.

Since the schedules used to obtain the samples utilized a part to machine ratio of 1:1 or greater, it is possible that the estimating function using R would not be applicable to the sample linear programming schedules which might have an R less than 1. Therefore, a series of programs were run in which essentially non-related independent variables were utilized. The first set, Table 3, reduced eventually to a three variable equation:

$$Y = -106.4120 + 0.5911\, \overline{M} + 1.3173\, \overline{P} + 36.9606\, R \tag{5.2}$$

The second set of data, Table 4, also reduced to a three variable equation:

$$Y = -106.9355 + 0.5964\, M_{max} + 1.1195\, P_{max} + 34.0844\, R \tag{5.3}$$

Both of the above equations removed 97 percent of the variance in the data. A single variable equation utilizing

TABLE 1

VARIABLES FOR REGRESSION ANALYSIS

Y	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
215	8	213.5	106.7	215	130	4.5	252.9	0.0	2.0	345
275	10	259.0	103.6	275	128	512.0	441.3	3.0	2.5	403
362	12	338.5	112.8	362	131	1104.5	247.7	2.0	3.0	493
345	14	340.0	97.1	345	110	50.0	98.1	4.0	3.5	455
398	16	391.5	97.8	398	114	84.5	144.6	0.0	4.0	512
480	18	455.0	101.1	480	127	1250.0	114.8	9.0	4.5	607
508	20	506.0	101.2	508	119	8.0	199.5	2.0	5.0	627
295	15	250.0	150.0	259	168	103.0	181.5	1.0	1.6	427
394	21	375.3	160.8	394	173	270.3	102.1	1.0	2.3	567
435	24	398.3	149.3	430	168	782.3	485.6	16.0	2.6	598
444	27	427.0	142.3	444	170	223.0	586.0	1.0	3.0	614
531	30	510.0	153.0	531	180	333.0	251.3	4.0	3.3	711
401	28	333.7	190.7	375	220	974.2	514.5	10.0	1.7	595
483	32	391.0	195.5	414	218	776.6	288.8	16.6	2.0	632
510	36	489.7	217.6	510	232	248.9	160.7	16.0	2.2	742
560	40	525.2	210.1	560	244	556.9	268.5	0.6	2.5	804
373	25	235.8	235.8	254	280	209.2	881.7	12.5	1.0	534
477	35	350.4	250.2	386	270	665.3	245.5	17.0	1.4	656
546	45	432.0	240.0	445	270	122.0	821.7	18.5	1.8	715
562	50	494.8	247.4	562	267	3122.7	178.0	3.5	2.0	829
494	42	340.1	291.5	394	311	876.5	354.6	31.6	1.1	705
608	48	403.6	302.7	440	343	884.2	974.2	14.4	1.3	783
565	54	459.8	306.5	491	350	668.5	53[illegible].0	35.8	1.5	841
679	60	505.5	303.3	523	340	371.5	990.6	42.8	1.6	863
604	49	352.4	352.4	395	380	929.9	498.2	22.6	1.0	775
617	56	373.8	327.1	430	358	1130.1	379.5	27.0	1.1	788
680	63	443.2	344.7	508	368	1785.2	222.7	55.6	1.2	876
640	70	500.5	350.4	554	390	895.6	357.3	31.6	1.4	944
721	64	390.6	390.6	454	449	903.1	1620.8	30.5	1.0	903
741	72	448.5	398.6	516	420	1516.5	262.5	86.2	1.1	936
753	80	489.0	391.2	525	447	610.8	792.1	39.1	1.2	972
816	81	457.5	457.5	516	489	804.2	595.2	112.0	1.0	1005
816	90	477.2	429.5	539	477	1264.4	571.8	102.5	1.1	1016
800	100	500.7	500.7	547	559	620.6	1147.3	109.7	1.0	1106
726	77	527.7	335.8	556	387	655.9	1168.9	69.3	1.5	943
812	84	602.8	351.6	654	402	1495.1	1105.5	69.0	1.7	1056
824	88	542.2	394.3	623	428	3166.7	398.6	92.5	1.3	1051
798	96	597.7	398.5	671	425	1705.6	385.9	56.0	1.5	1096
839	99	549.1	449.2	605	507	845.3	890.0	65.2	1.2	1112

Y = Schedule time — X_4 = Max. mach. load — X_8 = Weight factor

X_1 = Total operations — X_5 = Max. part time — X_9 = Part-Mach. ratio

X_2 = Av. mach. load — X_6 = Mach. variance — $X_{10} = X_4 + X_5$

X_3 = Av. part time — X_7 = Part variance

TABLE 2

REGRESSION ANALYSIS WITH 10 VARIABLES

VARIABLE	BETA	STD. BETA	"t"
PROD	-.244	-.039	-.194
$\overline{M}$	.256	.142	.349
$\overline{P}$	1.712	1.182	1.912
M_{max}	1.621	1.008	.046
P_{max}	.723	.540	.021
M_{var}	.012	.050	.795
P_{var}	.057	.119	2.163
WGT	.390	.076	1.089
R	37.522	.219	2.951
SUM	-1.286	-1.579	-.037

Y INTERCEPT = -97.739 CORRELATION INDEX: R^2 = .979

TABLE 3

REGRESSION OF 3 VARIABLES - CASE I

VARIABLE	BETA	STD. BETA	"t"
$\overline{M}$	.5911	.3272	6.737
$\overline{P}$	1.3173	.9093	11.537
R	36.9606	.2159	3.178

Y INTERCEPT = -106.412 CORRELATION INDEX: R^2 = .972

TABLE 4

REGRESSION OF 3 VARIABLES - CASE II

VARIABLE	BETA	STD. BETA	"t"
M_{max}	.5964	.3706	7.541
P_{max}	1.1195	.8355	10.693
R	34.0844	.1991	3.183

Y INTERCEPT = -106.936 CORRELATION INDEX: R^2 = .973

the sum of M_{max} and P_{max}, Figure 11, was calculated as:

$$Y = -40.4047 + .79977 \text{ Sum} \tag{5.4}$$

This equation accounted for 92.5 percent of the variation in the data and was independent of R and was used with the parametric linear program scheduling programs.

Since 2 and 3 machine schedules were responsible for most of the contributions to R in equations 5.1, 5.2, and 5.3, and since they also were the schedules which obtained most of the exact minimums, two additional runs of the multiple linear regression program were made in which all schedules having exact minimums were excluded. The first set, Table 5, reduced to:

$$Y = -38.26 + .724\,\bar{M} + 1.094\,\bar{P} \tag{5.5}$$

and the second set, Table 6, reduced to:

$$Y = -39.477 + .698M_{max} + .934P_{max} \tag{5.6}$$

Both of the above equations account for 95.5 percent of the variability of the data.

Because the estimating equations were derived from a particular universe, it was thought appropriate to transform the equations. A simple scale transformation resulted in the intercept of each equation being modified by a factor AVE/50, where AVE equals the average operation time in the sample schedules to be used and the 50 is the average operation time used in the Heller schedules from which the equations were derived.

There appears to be some inconsistency in applying the equations to the schedules generated in Chapter VI. As previously stated, each company using the proposed technique would be required to establish an estimating equation from its own production data and there would be no necessity for transforming the equations.

TABLE 5

REGRESSION OF 2 VARIABLES - CASE I

VARIABLE	BETA	STD. BETA	"t"
$\bar{M}$	.7243	.4321	7.966
$\bar{P}$	1.0940	.6518	12.017

Y INTERCEPT = -38.263 CORRELATION INDEX: R^2 = .9557

TABLE 6

REGRESSION OF 2 VARIABLES - CASE II

VARIABLE	BETA	STD. BETA	"t"
M_{max}	.6980	.4642	8.111
P_{max}	.9338	.6080	10.623

Y INTERCEPT = -39.477 CORRELATION INDEX: R^2 = .9551

Table 7 shows the results of running the Heller Scheduling Algorithm on 10 sets of data obtained from the P.L.P. programs which had utilized the transformed equation from 5.4. In this case the transformation produced unsatisfactory results but the original equation was highly significant.

TABLE 7

P.L.P. SCHEDULES VERSUS HELLER SCHEDULES

a	b	c	d	e
P.L.P.	Adj.P.L.P.*	Heller	(b-c)	$(d-\bar{d})^2$
119	86	91	-5	75.69
123	88	93	-5	75.69
121	86	92	-6	94.09
119	84	90	-6	94.09
119	85	90	-5	75.69
189	158 **	144	+13	86.49
127	92	97	-5	75.69
121	86	92	-6	94.09
119	84	91	-7	114.49
121	86	92	-6	94.09
	935	972	-37	880.10

$$s = \sqrt{\frac{\Sigma\,(d-\bar{d})^2}{n-1}} = \sqrt{\frac{880.10}{9}} = 9.89$$

H_o: $\bar{d} = 0$

$$t = \frac{\bar{d} - 0}{s/n^{1/2}} = \frac{3.7}{9.89/3.162} = 1.19 \text{ ,9d.f.}$$

$t^{9d.f.}_{21/2\%}$ = 2.262; Therefore accept hypothesis.

* The transformation derived in Chapter V to compensate for differences in population means and used in the computer program overcorrected and adjusted data eliminates the correction factor.

** This schedule did not converge but since the method of estimating schedule time is independent of the P.L.P. program, there is no justification for omitting it.

CHAPTER VI

APPLICATIONS

The scheduling time estimating functions which were derived in Chapter V may be utilized as limits for managerial control. All data necessary for using the estimating equations would be readily available when planning for production. By calculating the total schedule time for the production period, the planners would know whether or not the schedules could be met and how tight the scheduling might be since the equations are based on pseudo-optimum (minimum) time. The same technique could be used by an operations evaluations group to estimate the relative efficiency of production performed during a fixed period.

The mechanics of this type of control concept requires industrial investigations and was not in the realm of research going into this report. This dissertation was primarily concerned with proving the hypothesis that there would be a relatively simple relationship among scheduling input parameters and minimum scheduling time. The results substantiate the hypothesis and, in addition to the use as a control limit, per se, the relationship is most useful as an adjunct to linear programming and the balance of this chapter will discuss this application.

Linear programming may be used in planning and scheduling to produce two types of results. The first is to optimize profits by determining an optimum product mix and the second is to minimize cost by allocating part operations to the machines in such a manner that total cost is minimized. Both of these techniques are limited by their inability to adequately compensate for scheduling feasibility. The use of loading factors (i.e. using only 75% to 85% of actual available time as the machine hour constraints) partially defeats the purpose of using linear programming, since a 20% change in any constraint generally produces a completely new dissimilar solution.

By utilizing the optimizing parameter developed in this dissertation and linear programming with variable parameters, the solution to the above two linear programming problems can be adjusted to have a high probability of scheduling feasibility without the use of approximate loading factors. In order to illustrate the effect of loading factors and to show the results obtainable with the variable parameter method, two simple sample problems are herein presented.

First will be the determination of optimum mix for three parts made on three machines.

Profit = $\$10X_1 + \$14X_2 + \$8X_3$

X_1 requires 5 hours on M_1, 6 hours on M_2, and 2 hours on M_3.

X_2 requires 2 hours on M_1, 5 hours on M_2, and 4 hours on M_3.

X_3 requires 4 hours on M_1, 2 hours on M_2, and none on M_3.

During a 2 week scheduling period 80 hours are available for each machine. Technological ordering of each part is as follows:

$(211) \longrightarrow (112) \longrightarrow (313)$

$(121) \longrightarrow (222) \longrightarrow (323)$

$(131) \longrightarrow (232)$

Figure 16 shows the solution of the linear programming maximization of profit for optimum product mix using the 80 hour machine constraints. Figure 17 shows a similar solution where the machine hour constraints are all reduced by 20%. Figure 18 gives a solution where the constraints were selectively modified, 25% on machines 1 and 3 and no weighting of machine 2 to account for the fact that machine 1 was normally used for first operations and machine 3 for last operations and hence would cause bottlenecks. Comparison of the profit using only machine restraints is as follows: Unrestricted program, $260; 20% weighting, $208; 25-0-25 weighting, $245. Reducing the programs to the 80 hour period reduced the profit to $189, $189 and $142 respectively.

The assumption that each part is scheduled in one lot and splitting is not allowed permits optimum schedules to be calculated for the various linear program solutions. Times for the various schedules are calculated from the previous data and for the unrestricted program the times are as follows:

T(121) = 2 x 12.5 = 25 hours   T(131) = 4 x 8.75 = 35 hours

T(222) = 5 x 12.5 = 63 hours   T(232) = 2 x 8.75 = 18 hours

T(323) = 4 x 12.5 = 50 hours

	Z_o	0	0	0	\$10	\$14	\$8	
	X_o	X_4	X_5	X_6	X_1	X_2	X_3	θ
0	80	1	0	0	5	2	4	40
0	80	0	1	0	6	5	2	16
0	80	0	0	1	2	4	0	20
Z_1-Z_o		0	0	0	-10	-14	-8	
0	48	1	-2/5	0	13/5	0	16/5	15
\$14	16	0	1/5	0	6/5	1	2/5	40
0	16	0	-4/5	1	-14/5	0	-8/5	
Z_2-Z_o		0	14/5	0	34/5	0	-12/5	
\$ 8	15	5/16	-1/8	0	13/16	0	1	
14	10	-1/8	1/4	0	7/8	1	0	
0	40	1/2	5/2	1	-3/2	0	0	
Z_3-Z_o		3/4	5/2	0	35/4	0	0	

Total profit: 15 x \$8 + 10 x \$14 = \$260

Fig. 16. Calculation of optimum mix

	Z_o	0	0	0	\$10	\$14	\$8	
	X_o	X_4	X_5	X_6	X_1	X_2	X_3	θ
0	64	1	0	0	5	2	4	32
0	64	0	1	0	6	5	2	13
0	64	0	0	1	2	4	0	16
Z_1-Z_o		0	0	0	-10	-14	-8	
\$ 8	12	5/16	-1/8	0	13/16	0	1	
14	8	-1/8	1/4	0	7/8	1	0	
0	32	1/2	-1	1	-3/2	0	0	
Z_3-Z_o		3/4	5/2	0	35/4	0	0	

Total profit: 12 x \$8 + 8 x \$14 = \$208

Fig. 17. Initial and final matrix for 20% weighting

	Z_o	0	0	0	\$10	\$14	\$8	
	X_o	X_4	X_5	X_6	X_1	X_2	X_3	θ
0	60	1	0	0	5	2	4	30
0	80	0	1	0	6	5	2	16
0	60	0	0	1	2	4	0	15
Z_1-Z_o		0	0	0	-10	-14	-8	
0	10	.50	-1	1	-1.5	0	0	
\$ 8	8.75	.31	1.13	0	.82	0	1	
\$14	12.5	-.13	.25	0	.87	1	0	
Z_4-Z_o		.75	12.5	0	8.75	0	0	
Total Profit = 8.75 x \$8 + 12.5 x \$19 = 245								

Fig. 18. Initial and final matrix for 25-0-25 weighting

The Gantt chart in Figure 19 shows that it will take 138 hours to schedule the program and therefore the number of units shceduled and hence the profit, must be reduced by a factor of 80/138. Revised potential profit equals \$142. Calculations of time for the other two programs were made and results are shown in Gan t chart form in Figure 19. Neither of the two schedules can be met within the 80 hour limit and the potential profit of the 20% weighting factor is the same as the unrestricted problem while the potential profit of the 25-0-25 restriction is only \$142.

The results of the example on the following page suggest that another constraint should be added to the linear program model, namely that none of the parts can exceed a total of 80 hours. A program including this additional constraint for each product is shown in Figure 20. The solution to this program yielded a much different product mix

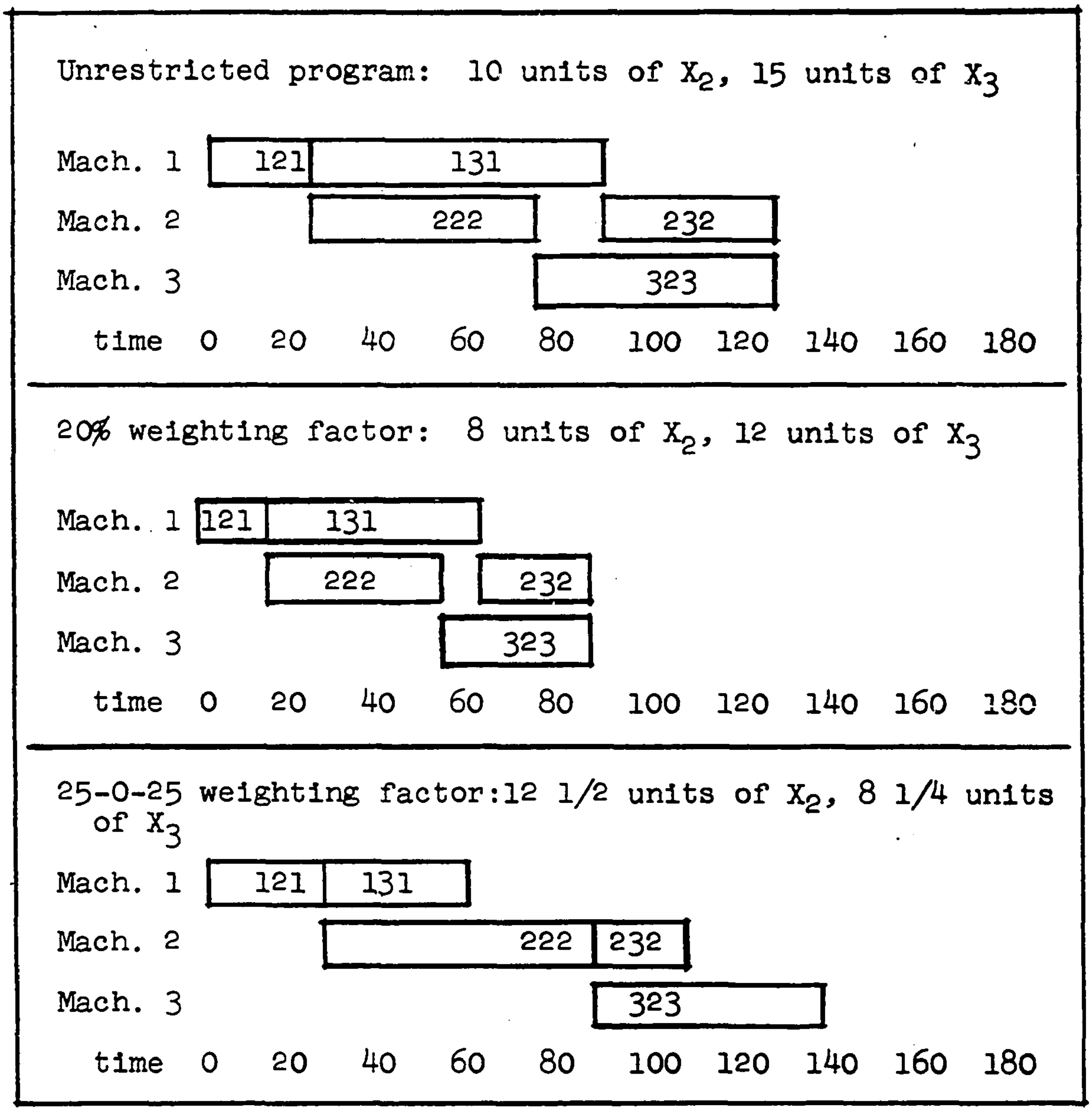


Fig. 19. Gantt charts of optimum linear programs

than before, and although it was feasible to schedule, it did not yield as much profit as the unrestricted program after adjustment for scheduling feasibility. This problem should amply illustrate the problem of using linear programming to determine optimum product mix. It was due to this problem that the scheduling parameters were hypothesized and then developed.

	Z_o	0	0	0	0	0	0	10	14	8
		X_4	X_5	X_6	X_7	X_8	X_9	X_1	X_2	X_3
0	80	1	0	0	0	0	0	5	2	4
0	80	0	1	0	0	0	0	6	5	2
0	80	0	0	1	0	0	0	2	4	0
0	80	0	0	0	1	0	0	13	0	0
0	80	0	0	0	0	1	0	0	11	0
0	80	0	0	0	0	0	1	0	0	6
Z_1-Z_o		0	0	0	0	0	0	-10	-14	-8
0	39.5	1.9	-3.7	0	1	1.35	0	0	0	0
8	12.5	.4	.3	0	0	.08	0	0	0	1
0	44.7	.3	.6	1	0	-.16	0	0	0	0
10	3.1	-.1	.3	0	0	-.10	0	1	0	0
14	7.3	0	0	0	0	.09	0	0	1	0
0	5.2	-2.6	-.2	0	0	-.50	1	0	0	0
Z_5-Z_o		2.2	0.6	0	0	0.9	0	0	0	0
Total Profit = 12.5 x \$8 + 3.1 x \$10 + 7.3 x \$14 = \$233.20										

Fig. 20. Initial and final matrix with added constraints

The scheduling parameters, as developed in Chapter 5, which should be incorporated in the linear programming model are calculated as follows:

Since the illustrative problem shown in Figure 20 has a part - machine ratio of 1:1, any of the estimating equations developed in Chapter V may be used. Due to the small number of operations, none of the equations can be expected to give as good results as the correlation coefficients indicate since the values lie at one end of the data used to calculate the regression lines.

Analysis of the solution to the linear program indicates that the following parameters are available:

	Part 1	Part 2	Part 3	M total
Mach. 1	16	15	50	81
Mach. 2	19	36	25	80
Mach. 3	6	29	-	35
P total	41	80	75	196

$P_{max} = 80$: $M_{max} = 81$: $R = 1$

AVE = 196/8 = 24: Sum = 80 + 81 = 161

$\overline{P} = \overline{M} = 196/3 = 65$

Eq. 5.1: $Y = -131.48 + 1.943\ \overline{P} + 94.710\ R$

$Y = 89.5$ Transformed: $Y = 158.3$

Eq. 5.2: $Y = -106.41 + .591\ \overline{M} + 1.317\ \overline{P} + 36.96\ R$

$Y = 54.6$ Transformed: $Y = 108.8$

Eq. 5.3: $Y = -106.94 + .596\ M_{max} + 1.120\ P_{max} + 34.08\ R$

$Y = 65.4$ Transformed: $Y = 119.7$

Eq. 5.4: $Y = -40.40 + .7998\ \text{Sum}$

$Y = 88.4$ Transformed: $Y = 109.4$

Eq. 5.5: $Y = -38.26 + .724\ \bar{M} + 1.09\ \bar{P}$

$Y = 79.9$ Transformed: $Y = 99.9$

Eq. 5.6: $Y = -39.48 + .698\ M_{max} + .934\ P_{max}$

$Y = 91.8$ Transformed: $Y = 112.2$

Although equation 5.6 gives the best results, knowing that the schedule can be completed in 93 units, equation 5.4 had been incorporated in the computer program and calculations of the solution, Figure 21, were readily available. Gantt charts for the initial program solution, Figure 20, and for the P.L.P. solution are given in Figure 22.

Iteration	Schedule hours
Initial solution	110.6
1st P.L.P. iteration	98.1
2nd P.L.P. iteration	100.9
3rd P.L.P. iteration	53.0
1st adjustment of θ	76.1
2nd adjustment of θ	78.3
3rd adjustment of θ	79.2
Final solution	

X_1: 4.08 x \$10 = \$ 40.80

X_2: 2.68 x \$14 = 37.52

X_3: 8.83 x \$ 8 = 70.64

Total Profit \$148.96

Fig. 21. Solution of P.L.P. Scheduling

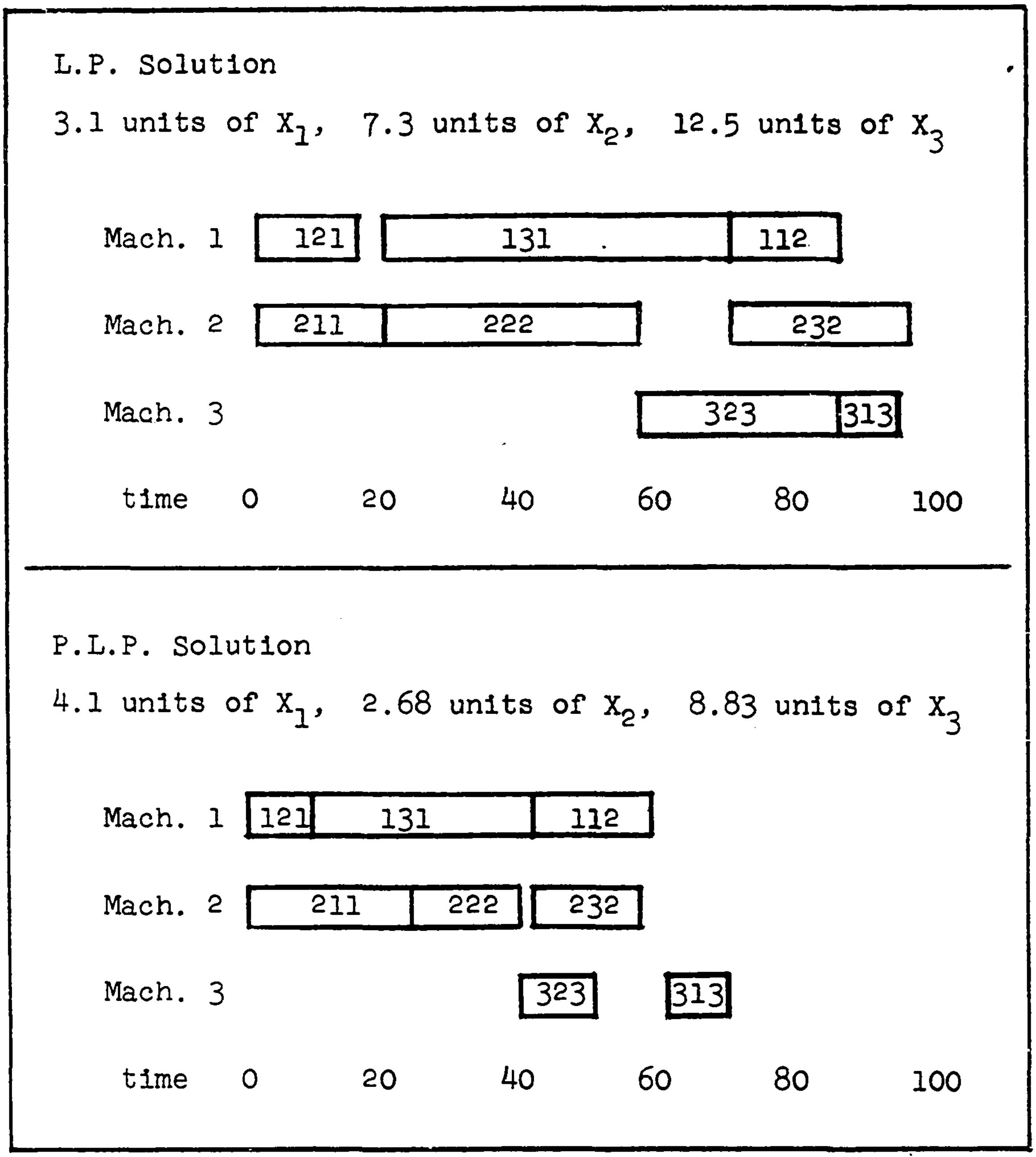


Fig. 22. Gantt Charts of Optimum Linear Programs

A second type of linear programming problem which requires adjustment for scheduling conflicts is the minimization of cost when parts can be made by more than one process. The following additional information is required for the linear programming solution of the example problem. Let X_{11}, X_{21}, and X_{31} designate parts X_1, X_2, and X_3 made in accordance with the original problem and let X_{12}, X_{22} and X_{32} represent parts X_1, X_2 and X_3 when made by a different method. For simplicity we will assume that the same three machines are used for both methods. X_{12} requires 3 hours on M_1, 4 hours on M_2 and 5 hours on M_3; X_{22} requires 3 hours on M_3, 2 hours on M_2 and 6 hours on M_1; X_{32} requires 4 hours each on M_1 and M_3. Since a minimum cost method would require fixed quantities of production for each part, the problem will be recast into a maximization of profit whereby the cost of making the parts by each method will be subtracted from their selling price. Profit per part for the first method will remain the same, and for the second method they will be \$8 for X_1, \$18 for X_2 and \$4 for X_3. The initial and final matrices of this problem are shown in Figure 23 without the 80 hour constraint on the parts and in Figure 25 with the constraints. In the first case the schedule cannot be met and the potential profit is 80/150 x \$320 = \$170. In the second case the optimum program cannot meet scheduling requirements but potential profit is now \$208. Gantt charts of the two schedules are shown in Figures 24 and 26. It is obvious that a better method is required for optimization of problems involving scheduling. The use of the previously developed scheduling parameters is one solution to this problem.

	Z_o	0	0	0	10	8	14	18	8	4
		X_4	X_5	X_6	X_{11}	X_{12}	X_{21}	X_{22}	X_{31}	X_{32}
0	80	1	0	0	5	3	2	6	4	4
0	80	0	1	0	6	4	6	2	2	0
0	80	0	0	1	2	5	4	3	0	4
Z_1-Z_o		0	0	0	-10	-8	-14	-18	-8	-4
18	10.0	.19	-.06	0	.56	.31	0	1	.62	.75
14	10.0	-.06	.19	0	.81	.56	1	0	.12	-.25
0	10.0	-.31	-.56	1	-2.15	1.82	0	0	-2.38	2.75
Z_3-Z_o		2.50	1.50	0	11.5	5.5	0	0	5.0	6.0
Total profit = 10.0 x \$18 + 10.0 x \$14 = \$320										

Fig. 23. Maximum profit with multiple methods

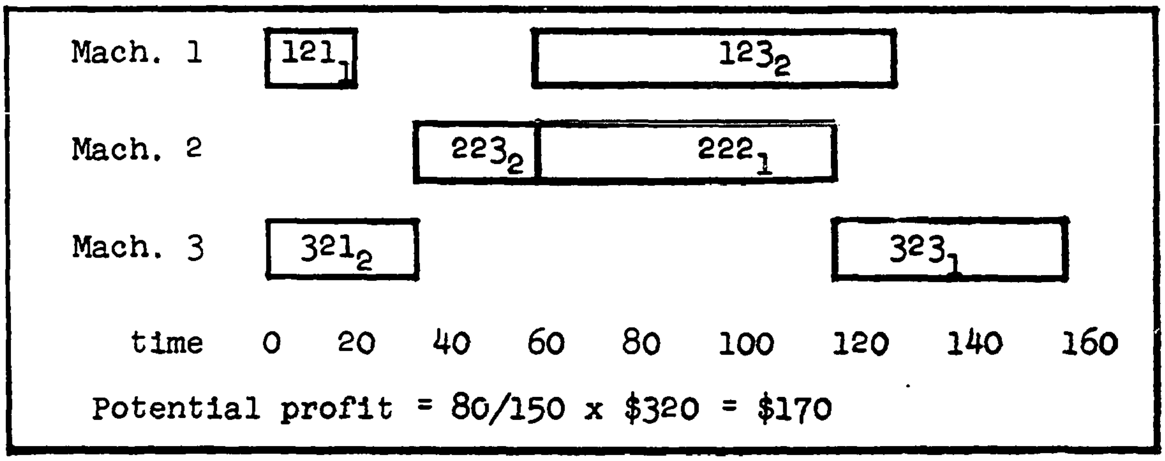
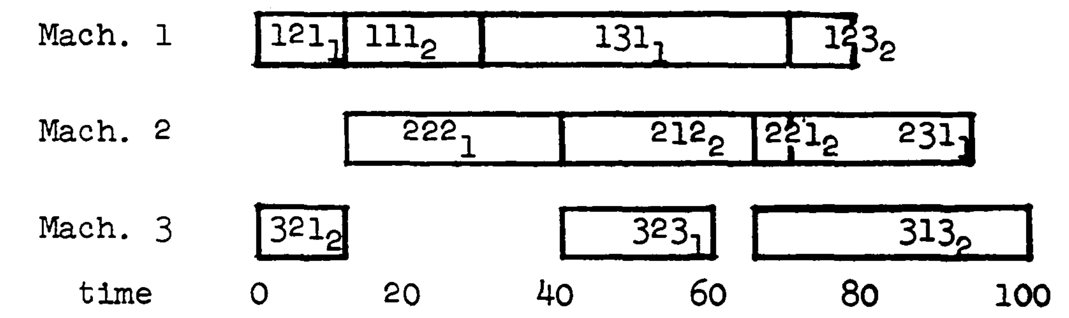


Fig. 24. Gantt Chart of Fig. 23 linear program

	Z_o	0	0	0	0	0	0	10	8	14	18	8	4
		X_1	X_2	X_3	X_4	X_5	X_6	X_{11}	X_{12}	X_{21}	X_{22}	X_{31}	X_{32}
0	80	1	0	0	0	0	0	5	3	2	6	4	4
0	80	0	1	0	0	0	0	6	4	6	2	2	0
0	80	0	0	1	0	0	0	2	5	4	3	0	4
0	80	0	0	0	1	0	0	13	12	0	0	0	0
0	80	0	0	0	0	1	0	0	0	12	11	0	0
0	80	0	0	0	0	0	1	0	0	0	0	6	8
	Z_1-Z_o	0	0	0	0	0	0	-10	-8	-14	-18	-8	-4
$ 8	9.8	.2	.2	0	-.10	.12	0	.6	0	0	0	1	.6
$14	5.0	-.1	.2	0	-.03	.02	0	.1	0	1	0	0	-.3
0	21.2	.1	-.1	1	-.39	-.28	0	-3.5	0	0	0	0	4.2
$ 8	6.7	0	0	0	.08	0	0	1.1	1	0	0	0	0
$18	1.8	.1	-.2	0	.04	.08	0	-.1	0	0	1	0	.4
0	21.2	-.9	-1.1	0	.61	.72	1	-3.5	0	0	0	0	4.2
	Z_6-Z_o	1.7	.6	0	.05	2.52	0	2.6	0	0	0	0	2.9

Total Profit = $234.20

Fig. 25. Maximum profit with added 80 hour constraints



Potential profit = 80/100 x $234.20 = $187.36

Fig. 26. Gantt chart of Fig. 25. linear program

The last example points out the need of some method to optimize various production processes in combination with a total schedule. The scheduling parameters which should be incorporated are derived from an analysis of the solution to the linear program of Figure 25 as follows:

	Part 12	Part 21	Part 22	Part 31	M total
Mach. 1	20	10	11	39	80
Mach. 2	27	30	4	20	81
Mach. 3	33	20	5	-	58
P total	80	60	20	59	219

$P_{max} = 80$ $M_{max} = 81$ Sum $= 80 + 81 = 161$

AVE $= 219/8 = 27$ $R = 4/3 = 1.333$

$\bar{M} = 219/3 = 73$ $\bar{P} = 219/4 = 55$

Eq. 5.1: $Y = -131.48 + 1.943\,\bar{P} + 94.710\,R$

$Y = 101.4$ Transformed: $Y = 161.65$

Eq. 5.2: $Y = -106.41 + .591\,\bar{M} + 1.317\,\bar{P} + 36.96\,R$

$Y = 58.6$ Transformed: $Y = 106.0$

Eq. 5.3: $Y = -106.94 + .596\,M_{max} + 1.120\,P_{max} + 34.08\,R$

$Y = 76.2$ Transformed: $Y = 139.6$

Eq. 5.4: $Y = -40.40 + .7998$ Sum

$Y = 88.4$ Transformed: $Y = 107.0$

Eq. 5.5: $Y = -38.26 + .724\,\bar{M} + 1.09\,\bar{P}$

$Y = 74.5$ Transformed: $Y = 92.1$

Eq. 5.6: $Y = -39.48 + .698\,M_{max} + .934\,P_{max}$

$Y = 101.8$ Transformed: $Y = 109.9$

As in the previous example, equations 5.1 and 5.6 without transformation show the best results. Equation 5.4, transformed, was used in the P.L.P. programs, initially, as the last two equations were calculated late in the research. In actual practice, the estimating equations would be calculated from production data and the problem of matching two different universes would be avoided.

Figure 27 shows the results of the P.L.P. calculations and Figure 28 includes Gantt charts for both the initial and the P.L.P. solutions.

Iteration	Schedule hours
Initial solution	113.4
1st P.L.P. iteration	104.6
2nd P.L.P. iteration	94.1
3rd P.L.P. iteration	0.00006
1st adjustment of θ	144.9
2nd adjustment of θ	144.9
3rd adjustment of θ	144.9
9th adjustment of θ	144.8
15th adjustment of θ	144.5
18th adjustment of θ	142.6
25th adjustment of θ	88.7
27th adjustment of θ	80.6

Final Solution:

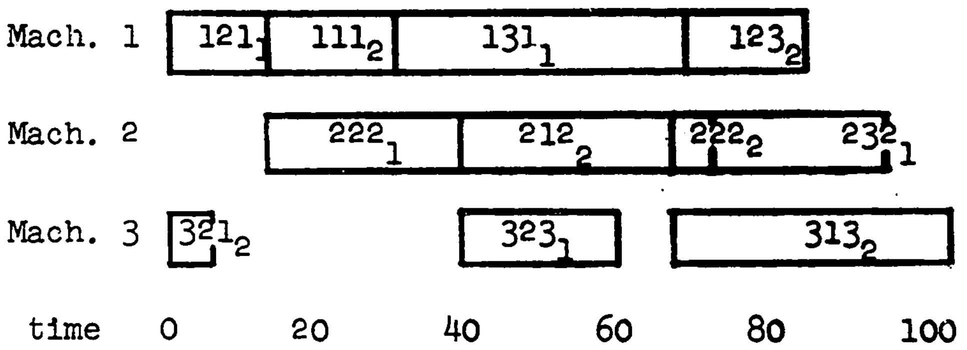
X_{11}:	3.43 x $10	=	$ 34.30
X_{22}:	4.04 x $18	=	72.72
X_{31}:	7.41 x $ 8	=	59.28
	Total Profit		$166.30

Fig. 27. Solution of P.L.P. Scheduling

L.P. Solution

6.7 units of X_{12}, 5.0 units of X_{21}

1.8 units of X_{22}, 9.8 units of X_{31}



P.L.P. Solution

3.4 units of X_{11}, 4.0 units of X_{22}, 7.4 units of X_{31}

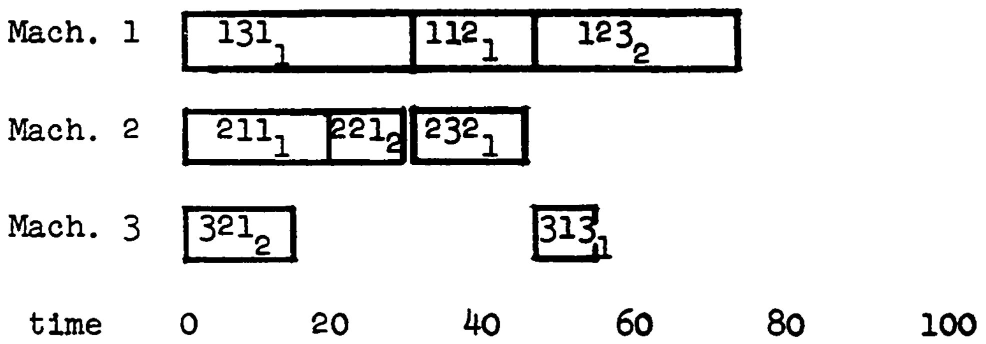


Fig. 28. Gantt Charts of Optimum Linear Programs

CHAPTER VII

CONCLUSIONS

The research conducted for this report has produced significant results, and while the optimum job-shop scheduling problem has not been solved, the empirical data accumulated throws additional light upon this subject.

The two fold purpose of this research, 1, providing useful techniques for industrial applications, and 2, providing data which would give insight into the 'scheduling problem' has been fulfilled. In addition to the concrete empirical results, the data poses several questions which require future investigations. A summary of the results, preceded by a list of qualifications, is as follows:

1. A sample size of 100 was used with the Heller algorithm instead of the preferred 200.
2. The range of schedule sizes was limited.
3. Only one set of normal times was used in the experiment.
4. No part to machine ratio of less than 1:1 was used.
5. Only one estimating function was utilized with the P.L.P. programs.

6. Only simulated data were utilized.

Considering the above mentioned limitations, it is remarkable that several linear relationships were found which can estimate minimum scheduling time from simple input parameters. Secondly, the use of the scheduling equation in parametric linear programming for optimum feasible schedules is a very useful innovation which has proven justifiable in the research thus far conducted.

Two avenues for further continuation and expansion of this research are open. The first is to generate additional data in the manner followed in this report without the above listed limitations. The second avenue is in analysing the results from a theoretical viewpoint. Several questions require answering before the secrets of job-shop scheduling can be unlocked. Some of the questions are:

1. What underlies the empirically determined linear relationship that determines the scheduling time and hence the delays?
2. What conditions are necessary for the modified Heller algorithm to pick minimum every time?
3. What underlies the result that 65 percent of the minimums were reached with sample sizes of less than 50?
4. The 65 percent of the samples in 3 had an average sample size of 12. The Heller scheduling algorithm would become economical if sample sizes could be this small. What sort of sequential

sampling plan can be devised to terminate the Heller algorithm?

5. What queueing model can represent the interferences that are present in job-shop scheduling?

The possibilities in the job-shop 'scheduling problem' are unlimited. This researcher hopes that the results in this dissertation will have contributed substantially to the understanding of this problem and that others may also continue along the avenues suggested herein.

LIST OF REFERENCES

1. Heller, J. "Some Numerical Experiments for an MxJ Flow Shop and Its Decision Theoretic Aspects," Operations Research, vol. 8 (1960), pp. 178-84.

2. Johnson, S. M. "Optimal Two- and Three-Stage Production Schedules With Set Up Time Included," Naval Research Logistics Quarterly, vol. 1 (1954), pp. 61-68.

3. Mitten, L. G. "Sequencing n Jobs on Two Machines With Arbitrary Time Lags," Management Science, vol. 5 (1959), pp. 293-98.

4. Akers, S. B. and J. Friedman. "A Non-numerical Approach to Production Scheduling Problems," Operations Research, vol. 3 (1955), pp. 429-42.

5. Rowe, A. J. "Toward a Theory of Scheduling," Journal of Industrial Engineering, vol. xi (1960), pp. 125-136.

6. Jackson, J. R. "Simulation Research on Job Shop Production," Naval Research Logistics Quarterly, vol. 4 (1957).

7. Sisson, R. L. "Methods of Sequencing in Job-shops - a Review," Operations Research, vol. 7 (1959), pp. 10-29.

8. Conway, R. W., B. M. Johnson, and W. L. Maxwell. "The Cornell Research Simulator," Department of Industrial and Engineering Administration, Cornell University, Ithica, New York, 1958.

9. Giffler, B. and G. L. Thompson. "Algorithms for Solving Production Scheduling Problems," Operations Research, vol. 8 (1960), pp. 487-503.

10. Bellman, R. "Mathematical Aspects of Scheduling Theory," Journal for the Society of Industrial and Applied Mathematics, vol. 4 (1956), pp. 168-205.

11. Muth, J. F. and G. L. Thompson (eds.). Industrial Scheduling. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1963.

12. Salveson, M. E. "A Computational Technique for the Scheduling Problem," Journal of Industrial Engineering, vol. xiii (1962), pp. 30-41.

13. Hardgrave, W. W. and G. L. Nemhauser. "A Geometric Model and A Graphical Algorithm for a Sequencing Problem," Operations Research, vol. 12 (1963), pp. 889-900.

14. Sasieni, M., A. Yaspan, and L. Friedman. Operations Research. New York: John Wiley and Sons Inc., 1959.

15. Heller, Jack and George Logemann. "An Algorithm for the Construction and Evaluation of Feasible Schedules," Management Science, vol. 8 (1962), pp. 168-83.

16. Heller, Jack. "Combinatorial, Probabilistic and Statistical Aspects of an M x J Scheduling Problem," A. E. C. Research and Development Report NYO-2540 (1959).

17. ________. "Some Problems in Linear Graph Theory That Arise in the Analysis of the Sequencing of Jobs Through Machines," A. E. C. Research and Development Report NYO-9487 (1960).

18. Giffler, B. and G. L. Thompson. "Algorithms for Solving Production - Scheduling Problems," Operations Research, vol. 8 (1960), pp. 487-503.

19. Andrus, W. E. Jr. and A. L. Becker. "A Production Schedule Computed on the I.B.M. 701," I.B.M. Technical Report No. 105.077.273. Endicott, New York: January 20, 1954.

20. Reid, M. "Job Shop Production Control on a Computer," Control Engineering, vol. 8 (1961), pp. 107-19.

21. Li, Jerome C. R. Introduction to Statistical Inference. Ann Arbor, Michigan: Edwards Brothers Inc., 1957.

22. Garvin, W. W. Introduction to Linear Programming. New York: McGraw-Hill, 1960.

23. Kramer, Clyde. "Multiple Regression" (A mineograph copy of a preliminary chapter in a proposed text) Virginia Polytechnic Institute, Blacksburg, Virginia, 1961.

BIBLIOGRAPHY

Ackoff, R. L. (ed.). *Progress in Operations Research, Vol. I.* (O.R.S.A. Publications in Operations Research No. 5) New York: John Wiley and Sons, Inc., 1961.

Avondo-Bodino, Giuseppe. *Economic Applications of the Theory of Graphs.* New York: Gordon and Breach, 1962.

Bellman, Richard. "Mathematical Aspects of Scheduling Theory," *Journal for the Society of Industrial and Applied Mathematics*, vol. 4 (1956), pp. 168-205.

________, and Oliver Gross. "Some Combinatorial Problems Arising in the Theory of Multi-stage Processes," *Journal for the Society of Industrial and Applied Mathematics*, vol. 2 (1954), pp. 175-83.

Bowman, E. H. "Production Scheduling by the Transportation Method for Linear Programming," *Operations Research*, vol. 4 (1956), pp. 100-03.

________. "The Schedule Sequence Problem," *Operations Research*, vol. 7 (1959), pp. 621-24.

________ and R. B. Fetter. *Analysis for Production Management.* rev. ed., Homewood, Illinois: Richard D. Irwin, Inc., 1961.

Brownlee, K. A. *Statistical Theory and Methodology in Science and Engineering.* New York: John Wiley and Sons, Inc., 1960.

Buffa, Elwood S. *Modern Production Management.* New York: John Wiley and Sons, Inc., 1961.

Chow, Y. and E. Cassignol. *Signal-Flow Graphs and Applications.* New York: John Wiley and Sons, Inc., 1962.

Conway, R. W. "Some Tactical Problems in Simulation Method," Rand Corporation Memorandum RM-3244-PR. Santa Monica, California: Rand Corporation, October 1962.

Conway, R. W. "An Experimental Investigation of Priority Assignment in a Job Shop," Rand Corporation Memorandum RM-3789-PR. Santa Monica, California: Rand Corporation, February 1964.

Dantzig, George B. "Recent Advances in Linear Programming," Management Science, vol. 2 (1956), pp. 131-44.

________. "A Machine-Job Scheduling Model," Management Science, vol. 6 (1960), pp. 191-96.

Dressin, S. A. and E. Reich. "Priority Assigment on a Waiting Line," Rand Corporation Memorandum P-845. Santa Monica, California: Rand Corporation, April 1956.

Ehrenfeld, S. and S. Ben-Tuvia. "The Efficiency of Statistical Simulation Procedures," Technometrics, vol. 4 (1962), pp. 257-75.

Ford, L. R. Jr. "Network Flow Theory," Rand Corporation Memorandum P-923. Santa Monica, California: Rand Corporation, 1956.

Fulkerson, D. R. "Flow Networks and Combinatorial Operations Research," Rand Corporation Memorandum RM-3378-PR. Santa Monica, California: Rand Corporation, October 1962.

Gass, S. I. Linear Programming. New York: McGraw-Hill, 1958.

Geisler, M. A. "The Sizes of Simulation Samples Required to Compute Certain Inventory Characteristics with Stated Precision and Confidence," Rand Corporation Memorandum RM-3242-PR. Santa Monica, California: Rand Corporation, October 1962.

________. "A Test of a Statistical Method for Computing Selected Inventory Model Characteristics by Simulation," Rand Corporation Memorandum RM-3623-PR. Santa Monica, California: Rand Corporation, May 1963.

Gomory, R. E. "Outline of an Algorithm for Integer Solutions to Linear Programs," Bulletin of the American Mathematical Society, vol. 64 (1958).

________. "An Algorithm for the Mixed Integer Problem," Rand Corporation Memorandum RM-2597. Santa Monica, California: Rand Corporation, July 1960.

Held, Michael and R. M. Karp. "A Dynamic Programming Approach to Sequencing Problems," Journal for the Society of Industrial and Applied Mathematics, vol. 10 (1962), pp. 196-210.

Johnson, S. M. "Sequential Production Planning Over Time at Minimum Cost," Management Science, vol. 3 (1957), pp. 435-37 and vol. 5 (1959), pp. 299-303.

Kahn, H. and A. W. Marshall. "Methods of Reducing Sample Size in Monte Carlo Computations," Operations Research, vol. 1 (1953), pp. 263-78.

Kim, W. H. and R. T. Chien. Topological Analysis and Synthesis of Communication Networks. New York: Columbia University Press, 1962.

Kruskal, J. B. Jr. "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem," Procedings of the American Mathematical Society, vol. 7 (1956), pp. 48-50.

Lescault, J. "Programmes Lineaires et Calculateurs Electroniques," Revue de Recherche Operationnelle, vol. 1 (1957).

Lesourne, J. "La Regulation Simultanee de la Production et des Stocks," Revue de Recherche Operationnelle, vol. 1 (1957).

Mood, A. M. Introduction to the Theory of Statistics. New York: McGraw-Hill, 1950.

Nicolson, D. L. and W. J. J. Hadnutt. "Production Planning and Control," Chartered Mechanical Engineer, vol. 9 (1962), pp. 522-29.

Ore, Oystein. Theory of Graphs. (American Mathematical Society Colloquium Publications Volume xxxviii) Providence, Rhode Island: American Mathematical Society, 1962.

Ostle, Bernard. Statistics in Research, 2nd ed., Ames, Iowa: The Iowa State University Press, 1963.

Page, E. "An Approach to the Scheduling of Jobs on Machines," Journal of the Royal Statistical Society, Series B, vol. 23, part 2 (1961), pp. 484-92.

Pahlitzsch, G. and U. Berr. "Ein Verfahren zur Fertigungssteuerung in der Mettalindustrie mit gekoppelten mathematischen Modellen," Werkstatts Technik, vol. 52 (1962), pp. 224-30 and pp. 401-06.

Riordan, John. Introduction to Combinatorial Analysis. New York: John Wiley and Sons Inc., 1958.

Saaty, Thomas L. Mathematical Methods of Operations Research. New York: McGraw-Hill, 1959.

Schmitt, H. "Produktionsplannung mit linear Programmierung," Elektronische Rechenanlagen, vol. 4 (1962), pp. 117-20.

Sisson, R. L. "Methods of Sequencing in Job Shops - A Review," Operations Research, vol. 7 (1959), pp. 10-29.

Vajda, S. Mathematical Programming. Reading, Massachusetts: Addison-Wesley Publishing Co., 1961.

Vazsonyi, A. Scientific Programming in Business and Industry. New York: John Wiley and Sons Inc., 1958.

Veinott, A. F. Jr. and H. Wagner. "Optimal Capacity Scheduling," Operations Research, vol. 10 (1962), pp. 518-46.

Wagner, Harvey M. "An Integer Linear Programming Model for Machine Scheduling," Naval Research Logistics Quarterly, vol. 6 (1959), pp. 131-40.

________ and R. J. Giglio. "Approximate Solutions to the Three Machine Scheduling Problem," Operations Research, vol. 12 (1964), pp. 305-324.

APPENDIX A

FLOW CHARTS AND FORTRAN LISTINGS

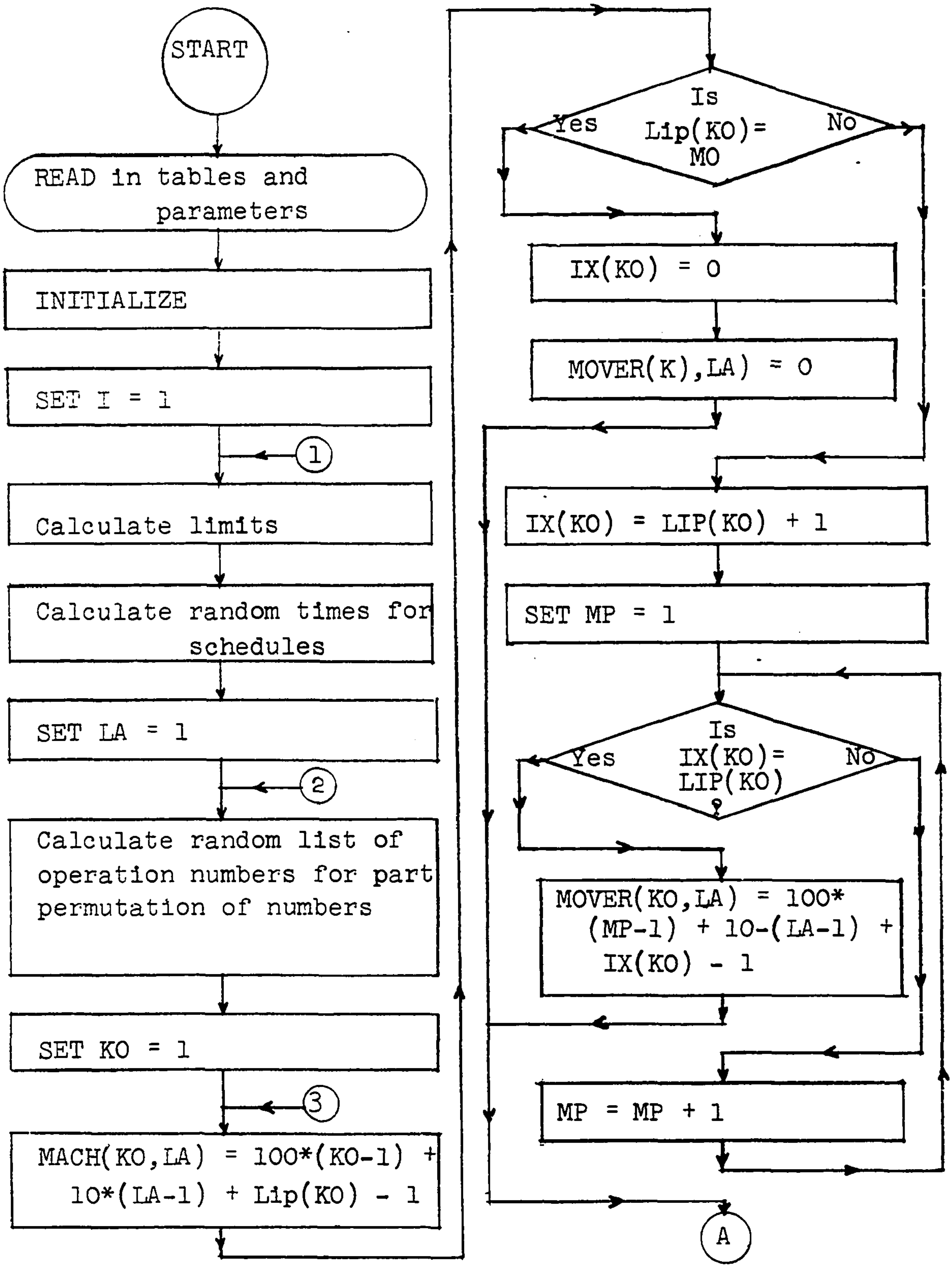


Fig. 29.--Flow chart for input data - part 1 of 2

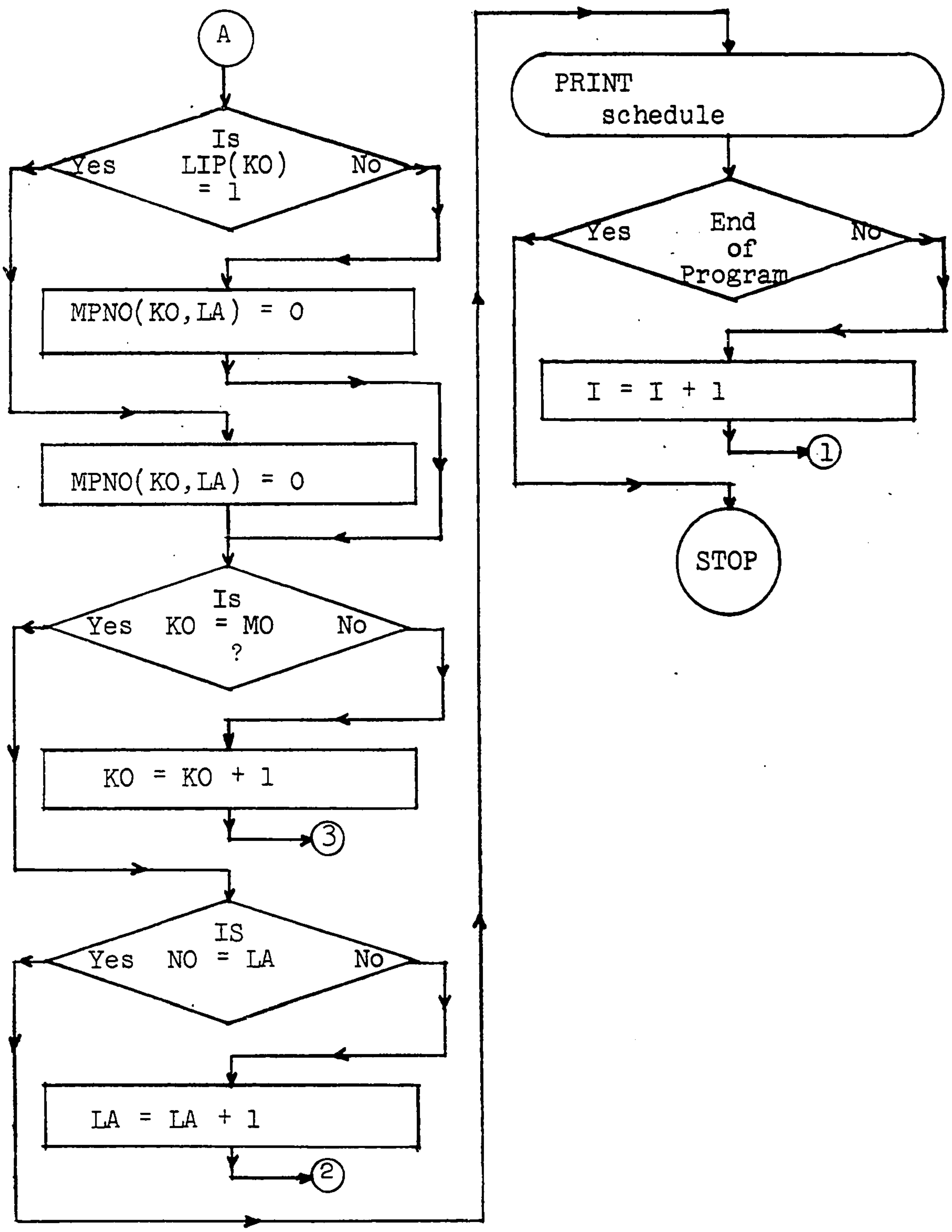


Fig. 29.--Flow chart for input data - part 2 of 2

```
C     RANDOMIZED DATA FOR INPUT TO MODIFIED HELLER SCHEDULING
      ALGORITHM
      DIMENSION LE(100),MIX(30),MACH(10,10),MOVER(10,10),MPNO
     1 (10,10),MIME(10,10),ITIE(9),IX(10),MC(34),NR(34),LIP(10)
      READ 26,(MIX(I),I=1,30)
   26 FORMAT(15I3)
      READ 2,(LE(I),I=1,100)
    2 FORMAT(20I2)
      READ 30,(ITIE(I),I=1,9)
   30 FORMAT(9I5)
      READ 3,(MC(I),I=1,34)
      READ 3,(NR(I),I=1,34)
    3 FORMAT(34I2)
      READ 4,L
    4 FORMAT(I5)
      DO 1200 I=1,34
      DO 100 J=1,10
      DO 100 K=1,10
      MACH(J,K)=0
      MOVER(J,K)=0
      MPNO(J,K)=0
  100 MIME(J,K)=0
      LO=3*I/4+1
      MO=MC(I)
      NO=NR(I)
      NUMRO=MO*NO
```

```
    MA=MO-1
    NA=NO-1
    LOAD=400+MIX(LO)
    DO 40 K=1,MO
    DO 40 M=1,NO
    I=L*LOAD
    MI=L/1000+1
 40 MIME(K,M)=LE(ML)
    DO 900 LA=1,NO
    DO 110 J=1,MO
    IX(J)=0
110 LIP(J)=0
    KAP=0
    DO 133 M=1,5000
    I=L*LOAD
    NL=L
    ID=0
    DO 140 IS=1,MA
    ID=ID+1
    IF(NL-ITIE(MA)) 115,140,140
140 NL=NL-ITIE(MA)
    ID=MO
115 DO 150 MS=1,MO
    IF(LIP(MS)-ID) 150,133,150
150 CONTINUE
    KAP=KAP+1
```

```
    LIP(KAP)=ID
    IF(KAP-MO) 133,135,135
133 CONTINUE
    PRINT 62
 62 FORMAT(1X,7HCYCLING)
    STOP
135 DO 900 KO=1,MO
    MACH(KO,LA)=100*(KO-1)+10*(LA-1)+LIP(KO)-1
    IF(LIP(KO)-MO) 165,162,165
162 IX(KO)=0
    MOVER(KO,LA)=0
    GO TO 143
165 IX(KO)=LIP(KO)+1
    DO 850 MP=1,MO
    IF(IX(KO)-LIP(MP)) 850,830,850
830 MOVER(KO,LA)=100*(MP-1)+10*(LA-1)+IX(KO)-1
    GO TO 143
850 CONTINUE
143 IF(LIP(KO)-1) 200,200,205
200 MPNO(KO,LA)=0
    GO TO 900
205 MPNO(KO,LA)=1
900 CONTINUE
    DO 300 J=1,MO
    DO 300 K=1,NO
300 PUNCH 303,MACH(J,K),MOVER(J,K),MPNO(J,K),MIME(J,K),MO,NO
```

```
  303 FORMAT(4I4,40X,2I4)
 1200 CONTINUE
      PRINT 1414
 1414 FORMAT(1X14HEND OF PROGRAM)
      STOP
      END
```

```
-91+91-83+77-69+67-61+59-53+37-29+27-21+19-13
+03-11+13-19+21-27+29-37+53-59+61-67-77+69+83
242830323334353536373738383939404040404141
424242434343444444454545454646464747474747
4848484849494949505050505151515152525252
535353535454545555555555565656575757585858
5959606060616162626363646565666768707276
5000033333250002000016667142861250011111110000
020202020202020203030303030404040405050505060606060707070708080
8090910
0405060708091005070809100708091005070910070809100708091008091
0091010
45329
```

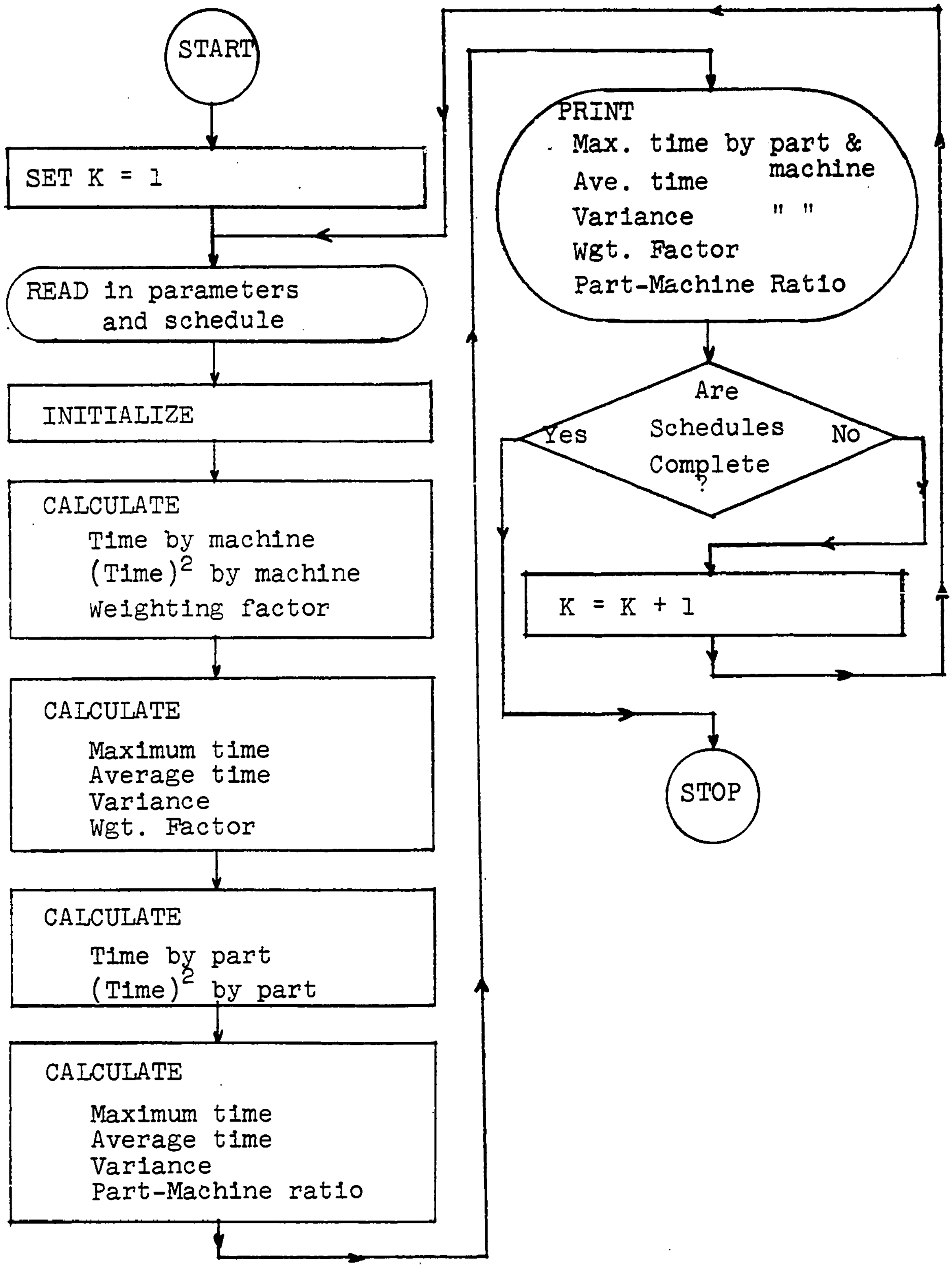


Fig. 30.--Analysis of input data

```
C     ANALYSIS OF INPUT DATA
      DIMENSION MACH(10,12),MIME(10,12),MTIME(10),NPTIME(12),
     1KWGT(10)
      DO 400 K = 1,39
      READ 1,M,N
    1 FORMAT(3X,2I2)
      DO 2 I = 1,M
      DO 2 J = 1,N
    2 READ 3, MACH(I,J),MIME(I,J)
    3 FORMAT(I4,8X,I4)
      X = M
      Y = N
      DO 6 I = 1,M
      MTIME(I) = 0
      KWGT(I) = 0
      DO 4 J = 1,N
      MTIME(I) = MTIME(I) + MIME(I,J)
    4 KWGT(I) = MACH(I,J)-1000*(I-1)-10*(J-1) + KWGT(I)
    6 CONTINUE
      MT = MTIME(1)
      DO 20 I = 2,M
      IF(MT-MTIME(I)) 21,20,20
   21 MT = MTIME(I)
   20 CONTINUE
      SUMX = 0.0
      SOSX = 0.0
```

```
   DO 30 I = 1,M
   TIME = MTIME(I)
   SUMX = SUMX + TIME
30 SOSX = SOSX + TIME*TIME
   XBAR = SUMX/X
   VARX=(SOSX-XBAR*SUMX)/(X-1.0)
   LWGT=0
   MWGT=0
   DO 22 I=1,M
   MWGT=MWGT+KWGT(I)
22 LWGT=LWGT+KWGT(I)*KWGT(I)
   WGT=LWGT-MWGT/M*MWGT
   WGT=WGT/(X-1.0)
   DO 8 J=1,N
   NPTIME(J)=0
   DO 7 I=1,M
 7 NPTIME(J)=NPTIME(J)+MIME(I,J)
 8 CONTINUE
   SUMY=0.0
   SOSY=0.0
   DO 31 I=1,N
   TIME=NPTIME(I)
   SUMY=SUMY+TIME
31 SOSY=SOSY+TIME*TIME
   YBAR=SUMY/Y
   VARY=(SOSY-YBAR*SUMY)/(Y-1.0)
```

```
      NP=NPTIME(1)
      DO 23 I=2,N
      IF(NP-NPTIME(I)) 24,23,23
   24 NP=NPTIME(I)
   23 CONTINUE
      RATIO=Y/X
      PUNCH 9,M,MT,XBAR,VARX,WGT,N,NP,YBAR,VARY,RATIO
    9 FORMAT(2(2I4,3F8.3))
  400 CONTINUE
      PRINT 1414
 1414 FORMAT(1X,14HEND OF PROGRAM)
      STOP
      END
```

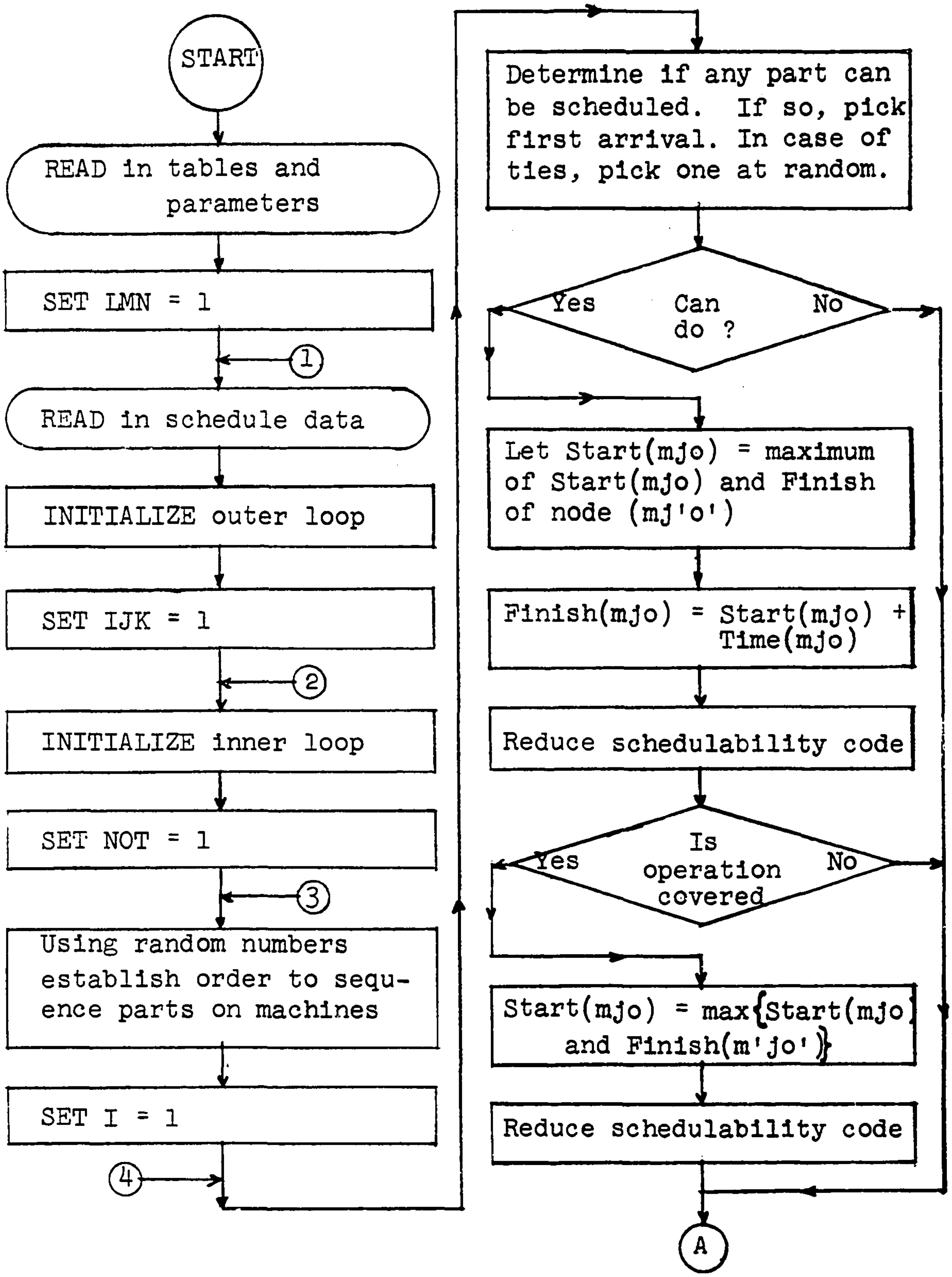


Fig. 31.--Flow chart of scheduling program - part 1 of 2

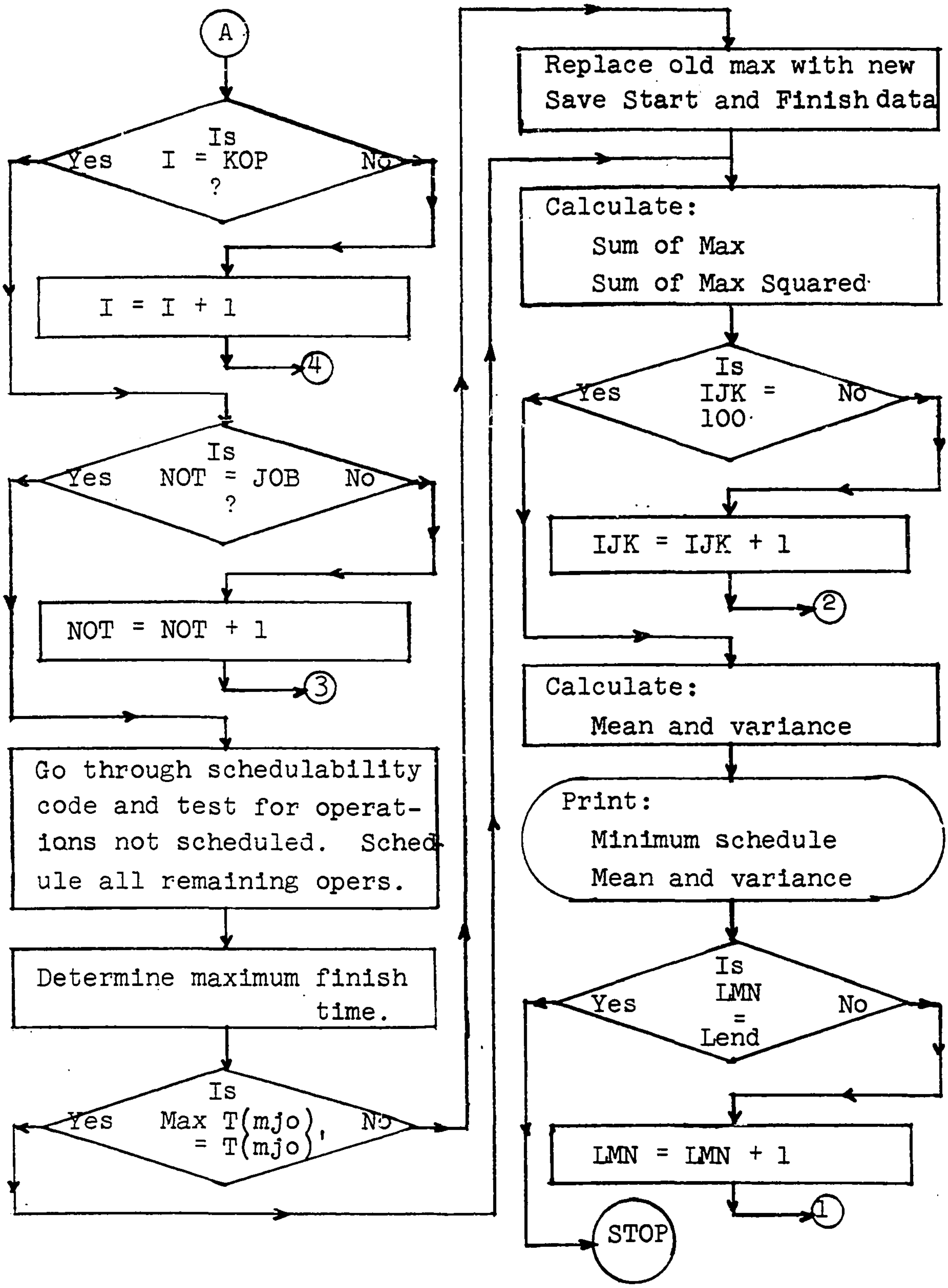


Fig. 31.--Flow chart of scheduling program - part 2 of 2

```
C     MODIFIED HELLER SCHEDULING ALGORITHM
      DIMENSION MACH(10,10),MOVER(10,10),MPNO(10,10),MIME(10,
     110),JPNO(10,10),MART(10,10),MINIS(10,10),JART(10,10),JI
     2NIS(10,10),LIP(10),ICHCK(10),ITIE(10),IXNAY(10),MIX(30)
     3,LOFX(10),JCHCK(10)
      READ 31,LEND,L
   31 FORMAT(2I5)
      READ 30,(ITIE(I),I=1,9)
   30 FORMAT(9I5)
      READ 26,(MIX(I),I=1,30)
   26 FORMAT(15I3)
      DO 850 LMN=1,LEND
      READ 20,NUMRO,KOP,JOB
   20 FORMAT(I3,I2,I2)
      DO 21 I=1,KOP
      DO 21 J=1,JOB
   21 READ 22,MACH(I,J),MOVER(I,J),MPNO(I,J),MIME(I,J)
   22 FORMAT(4I4)
      KSUM=0
      KBIG=0
      SUM=0.0
      SOS=0.0
      KMAX=99999
      LKOP=KOP-1
      LAX=ITIE(LKOP)
      KJOB=JOB-1
```

```
      DO 450 IJK=1,100
      KRAD=IJK/7+1
      LOAD=400+MIX(KRAD)
      DO 120 I=1,KOP
      LOFX(I)=0
      IXNAY(I)=0
      DO 120 J=1,JOB
      MART(I,J)=0
      MINIS(I,J)=0
  120 JPNO(I,J)=MPNO(I,J)
      DO 660 NOT=1,JOB
      DO 130 M=1,KOP
  130 LIP(M)=0
      K=0
      DO 133 M=1,1000
      L=L*LOAD
      ID=(L-1)/LAX+1
      DO 150 J=1,KOP
      IF(LIP(J)-ID)150,133,150
  150 CONTINUE
      K=K+1
      LIP(K)=ID
      IF(K-KOP)133,135,135
  133 CONTINUE
      PRINT 134
  134 FORMAT(1X,14HGENERATOR LOOP)
```

```
      STOP
  135 DO 181 I=1,KOP
      K=LIP(I)
      M=0
      DO 182 J=1,JOB
      IF(JPNO(K,J))155,165,155
  165 M=M+1
      ICHCK(J)=MART(K,J)
      JCHCK(J)=J
      GO TO 182
  155 ICHCK(J)=99999
  182 CONTINUE
      IF(M)175,181,175
  175 IXNAY(K)=IXNAY(K)+1
      DO 210 IX=1,KJOB
      KMO=JOB-IX
      DO 210 JX=1,KMO
      JO=JX+1
      IF(ICHCK(JX)-ICHCK(JO))210,210,215
  215 KOLD=ICHCK(JX)
      ICHCK(JX)=ICHCK(JO)
      ICHCK(JO)=KOLD
      MOLD=JCHCK(JX)
      JCHCK(JX)=JCHCK(JO)
      JCHCK(JO)=MOLD
  210 CONTINUE
```

```
      NA=0
      DO 220 IX=2,JOB
      IF(ICHCK(1)-ICHCK(IX))192,220,220
  225 PRINT 226
  226 FORMAT(1X,12HERROR AT 225)
      STOP
  220 NA=NA+1
      GO TO 240
  192 IF(NA)300,300,240
  240 L=L*LOAD
      IE=(L-1)/ITIE(NA)+1
      GO TO 330
  300 IE=1
  330 IE=JCHCK(IE)
      IF(IXNAY(K)-1)396,455,356
  356 IF(LOFX(K)-MART(K,IE))455,455,405
  405 MART(K,IE)=LOFX(K)
      GO TO 455
  396 PRINT 397
  397 FORMAT(1X,9HERROR 396)
      STOP
  455 MINIS(K,IE)=MART(K,IE)+MIME(K,IE)
      LOFX(K)=MINIS(K,IE)
      JPNO(K,IE)=JPNO(K,IE)-1
      IF(MOVER(K,IE))392,181,392
  392 JK=(MOVER(K,IE))/100+1
```

```
      LK=MOVER(K,IE)-100*(JK-1)
      LK=LK/10+1
      JPNO(JK,LK)=JPNO(JK,LK)-1
      IF(MART(JK,LK)-LOFX(K))380,181,181
  380 MART(JK,LK)=LOFX(K)
  181 CONTINUE
  660 CONTINUE
  712 KAT=0
      DO 810 I=1,KOP
      IF(IXNAY(I)-JOB)815,810,810
  815 DO 810 J=1,JOB
      IF(JPNO(I,J))810,820,830
  820 JPNO(I,J)=-1
      IF(LOFX(I)-MART(I,J))825,825,835
  835 MART(I,J)=LOFX(I)
  825 MINIS(I,J)=MART(I,J)+MIME(I,J)
      LOFX(I)=MINIS(I,J)
      IXNAY(I)=IXNAY(I)+1
      IF(MOVER(I,J))840,810,840
  840 JK=(MOVER(I,J))/100+1
      LK=MOVER(I,J)-100*(JK-1)
      LK=LK/10+1
      JPNO(JK,LK)=JPNO(JK,LK)-1
      IF(MART(JK,LK)-LOFX(I))950,810,810
  950 MART(JK,LK)=LOFX(I)
  830 KAT=1
```

```
810 CONTINUE
    IF(KAT)714,714,712
714 MINK=MINIS(1,1)
    DO 500 I=1,KOP
    DO 500 J=1,JOB
    IF(MINK-MINIS(I,J))505,500,500
505 MINK=MINIS(I,J)
500 CONTINUE
    IF(KMAX-MINK)515,525,510
515 IF(MINK-KBIG)525,525,516
516 KBIG=MINK
    GO TO 525
510 KMAX=MINK
    LAST=IJK
    DO 520 I=1,KOP
    DO 520 J=1,JOB
    JART(I,J)=MART(I,J)
520 JINIS(I,J)=MINIS(I,J)
525 FINIS=MINK
    SUM=SUM+FINIS
    SOS=SOS+FINIS*FINIS
450 CONTINUE
    XBAR=SUM/100.0
    VAR=(SOS-XBAR*SUM)/99.0
    STD=SQRTF(VAR)
    PRINT 1700
```

```
1700 FORMAT(1X,9HOPERATION,3X,5HSTART,3X,6HFINISH)
     DO 700 I=1,KOP
     DO 700 J=1,JOB
 700 PRINT 710,MACH(I,J),JART(I,J),JINIS(I,J)
 710 FORMAT(1X,1X,I5,5X,I5,4X,I5)
     PRINT 740,KMAX,XBAR,VAR,STD,LAST,KBIG
 740 FORMAT(1X,5HKMAX=,I5,2X,5HXBAR=,F8.2,2X,4HVAR=,F8.2,6H
    1 STD=,F8.2,2X,5HLAST=,I5,2X,5HKBIG=,I5)
 850 CONTINUE
     PRINT 1414
1414 FORMAT(1X,14HEND OF PROGRAM)
     STOP
     END
```

```
C     DOOLITTLE PROCEDURE FOR 12 VARIABLES[1]
      DIMENSION A(13,13),B(13,13),AY(13),BY(13),BETA(13)
      DIMENSION C(13,13),X(13),SUM(13),SCP(13,13),CCP(13,13)
     1,SCPY(13)
      DIMENSION CCPY(13),SB(13),XBAR(13)
      READ 330,JKLM
  330 FORMAT(I2)
      DO 105 MS=1,JKLM
      READ 1,M,N
    1 FORMAT(I3,I3)
      N1=M
      SUMY=0.0
      SSY=0.0
      CSSY=0.0
      B1=N
      DO 99 I=1,M
      SUM(I)=0.0
      SCPY(I)=0.0
      CCPY(I)=0.0
      DO 98 J=1,M
  502 C(I,J)=0.0
      SCP(I,J)=0.0
   98 CCP(I,J)=0.0
   99 CONTINUE
```

[1]This program was furnished by the University of Oklahoma Medical School Computing Center.

```
      DO 15 K=1,N
      READ 2,Y,X(1),X(2),X(3),X(4),X(5),X(6),X(7),X(8),X(9),
     1X(10),X(11),X(12)
    2 FORMAT(13F6.1)
      SUMY=SUMY+Y
      SSY=SSY+Y*Y
      DO 15 I=1,M
      SUM(I)=SUM(I)+X(I)
      SCPY(I)=SCPY(I)+Y*X(I)
      DO 6 J=1,M
    6 SCP(I,J)=SCP(I,J)+X(I)*X(J)
   15 CONTINUE
      DO 17 I=1,M
      DO 13 J=1,M
      CCP(I,J)=SCP(I,J)-((SUM(I)*SUM(J))/B1)
   13 CONTINUE
      CCPY(I)=SCPY(I)-((SUMY*SUM(I))/B1)
      XBAR(I)=SUM(I)/B1
   17 CONTINUE
      CSSY=SSY-((SUMY*SUMY)/B1)
      YBAR=SUMY/B1
      I=1
      AY(I)=CCPY(I)
      DO 4 J=1,M
    4 A(I,J)=CCP(I,J)
      IF(A(I,I)) 145,146,145
```

```
145 BY(I)=AY(I)/A(I,I)
    DO 14 J=1,M
 14 B(I,J)=A(I,K)/A(I,I)
146 DO 108 I=2,M
    L=I-1
    DO 18 J=I,M
    CFY=0.0
    CFA=0.0
    DO 147 K=1,L
    CFY=A(K,I)*BY(K)
147 CFA=CFA+A(K,I)*B(K,J)
    A(I,J)=CCP(I,J)-CFA
    IF(A(I,I)) 16,108,16
 16 B(I,J)=A(I,J)/A(I,I)
 18 CONTINUE
    AY(I)=CCPY(I)-CFY
    BY(I)=AY(I)/A(I,I)
108 CONTINUE
    DO 80 I=1,M
 80 BETA(I)=0.0
    I=M
    BETA(I)=BY(I)
 20 I=M-1
    L=I+1
    CFB=0.0
    DO 22 K=L,N1
```

```
 22 CFB=CFB+B(I,K)*BETA(K)
    BETA(I)=BY(I)-CFB
    M=M-1
    IF(M-1) 21,21,20
 21 RSS=0.0
    M=N1
    PRINT 217
217 FORMAT(15X,19HDOOLITTLE PROCEDURE)
    PRINT 220
220 FORMAT(23HREGRESSION COEFFICIENTS)
    PRINT 221
221 FORMAT(2X,1HI,6X,7HBETA(I),9X,4HMEAN,8X,11HSTD BETA(I))
311 DO 109 I=1,M
    RSS=RSS+BETA(I)*CCPY(I)
    STDB=BETA(I)*SQRTF(CCP(I,I)/CSSY)
    PRINT 500,I,BETA(I),XBAR(I),STDB
500 FORMAT(I4,3F15.6)
109 CONTINUE
    CFIC=0.0
    DO 201 I=1,M
    CFIC=CFIC+BETA(I)*(SUM(I)/B1)
201 CONTINUE
    YINCT=YBAR-CFIC
    PRINT 212,YINCT
212 FORMAT(12HY INTERCEPT=,F15,4)
    DF2=M
```

```
    Df1=N-1
    RMS=RSS/DF2
    RESS=CSSY-RSS
    REDF=DF1-DF2
    REMS=RESS/REDF
    F=RMS/REMS
    PRINT 206
206 FORMAT(17HAOV OF REGRESSION)
    PRINT 207
207 FORMAT(12X,2HDF,10X,2HSS,15X,2HMS,15X,1HF)
    PRINT 208,DF2,RSS,RMS,F
208 FORMAT(10HREGRESSION,F5.0,F15.3,F15.3,F15.3)
    SY=SQRT((CSSY-RSS)/REDF
    PRINT 209,REDF,RESS,REMS
209 FORMAT(8HRESIDUAL,2X,F5.0,F15.3,F15.3)
    PRINT 210,DF1,CSSY
210 FORMAT(5HTOTAL,5X,F5.0,F15.3,F15.3)
    RMCC=RSS/CSSY
    RMCC=SQRT(RMCC)
    RSQ=RMCC*RMCC
    PRINT 213
213 FORMAT(32HMULTIPLE CORRELATION COEFFICIENT)
    PRINT 222,RMCC,RSQ
222 FORMAT(2HR=,F15.5,5X,10HR SQUARE = ,F15.5)
    PRINT 216
216 FORMAT(26HREDUCTION DUE TO LAST BETA)
```

```
      DO 29 I=1,N1
      REDBN=AY(I)*BY(I)
   29 PRINT 214,REDBN
  214 FORMAT(F15.4)
   41 M=N1
   42 I=M
      J=M
      CFC=0.0
   43 IF(I-J) 51,52,53
   52 K5=J+1
      IF(K5-N1) 62,62,61
   62 DO 154 K=K5,N1
      CFC=CFC+B(I,K)*C(I,K)
  154 CONTINUE
   61 IF(A(I,J)) 60,45,60
   60 C(I,J)=(1.0/A(I,J))-CFC
   45 I=I-1
      IF(I) 56,44,43
   44 M=M-1
      IF(M) 42,56,42
   51 L=I+1
      DO 55 K=L,N1
   55 C(I,J)=C(I,J)-B(I,K)*C(K,J)
   53 C(J,I)=C(I,J)
      GO TO 45
   56 M=N1
```

```
      PRINT 211
  211 FORMAT(3X,1HI.10X,2HSB,15X,1HT,12X,2HDF,5X,15HSSB(I)
     1ADJUSTED)
      DO 200 I=1,M
      SB(I)=SQRT(C(I,I)*SY*SY)
      T=BETA(I)/SB(I)
      XSS=(BETA(I)*BETA(I))/C(I,I)
      PRINT 205,I,SB(I),T,REDF,XSS
  205 FORMAT(I4,4F15.4)
  200 CONTINUE
      PRINT 223
  223 FORMAT(3X,1HI,3X,IHJ,7X,6HC(I,J))
  202 DO 203 I=1,M
      DO 203 J=I,M
  203 PRINT 215,I,J,C(I,J)
  215 FORMAT(I4,I4,5X,E14.8)
  105 CONTINUE
      PRINT 1414
 1414 FORMAT(14HEND OF PROGRAM)
      STOP
      END
```

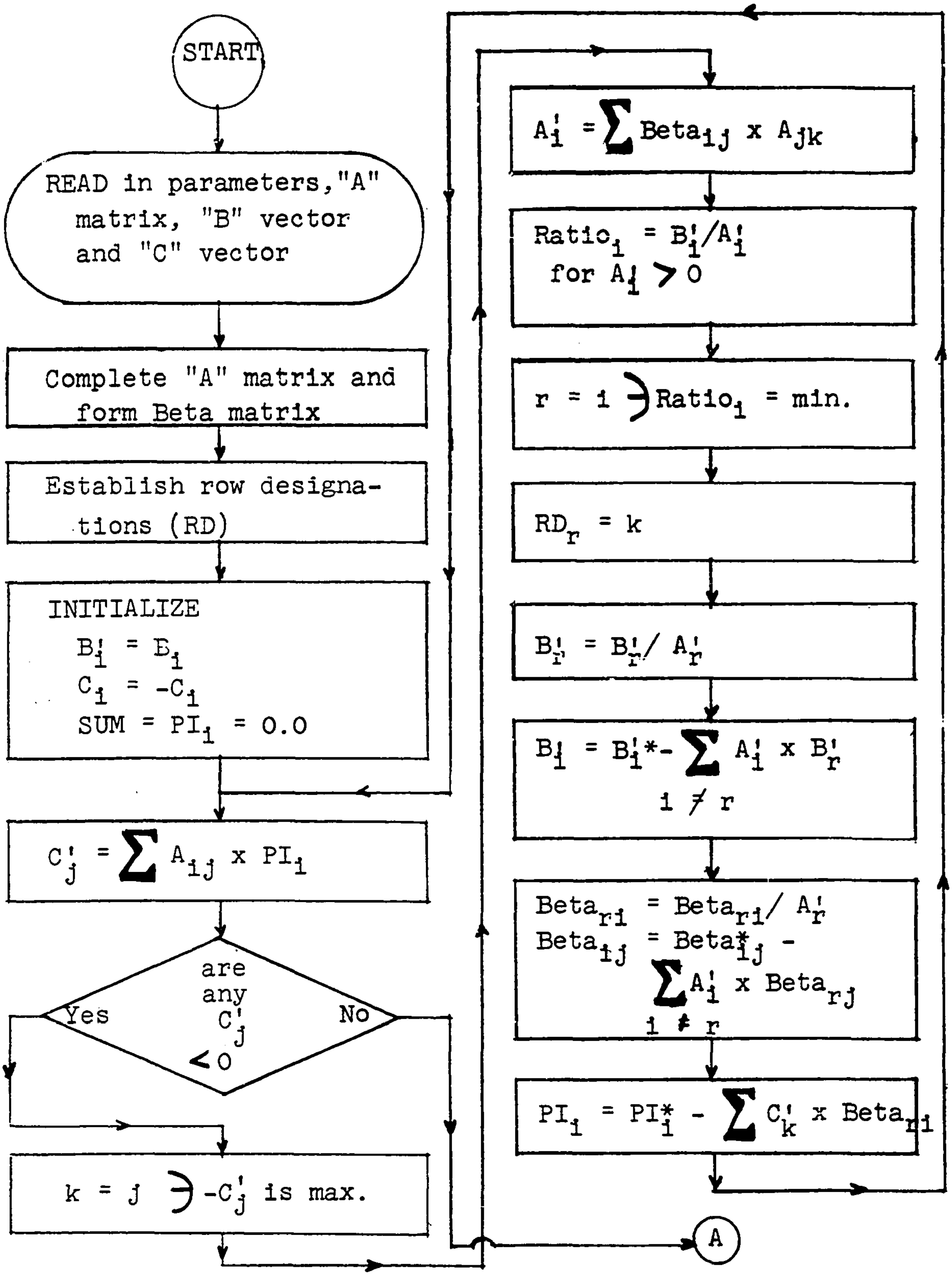


Fig. 32. Flow chart of P.L.P. program - part 1 of 4

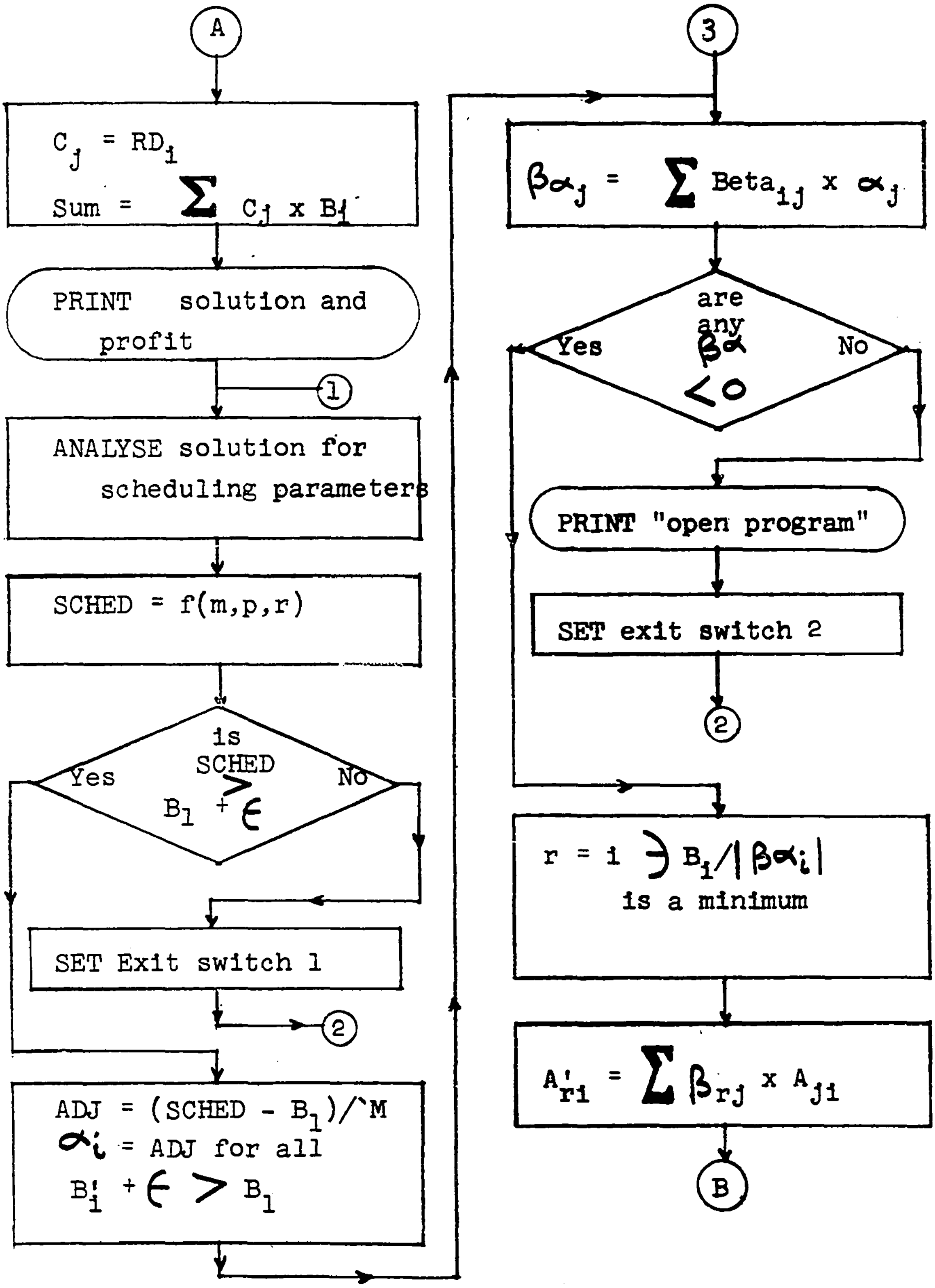


Fig. 32. Flow chart of P.L.P. program - part 2 of 4

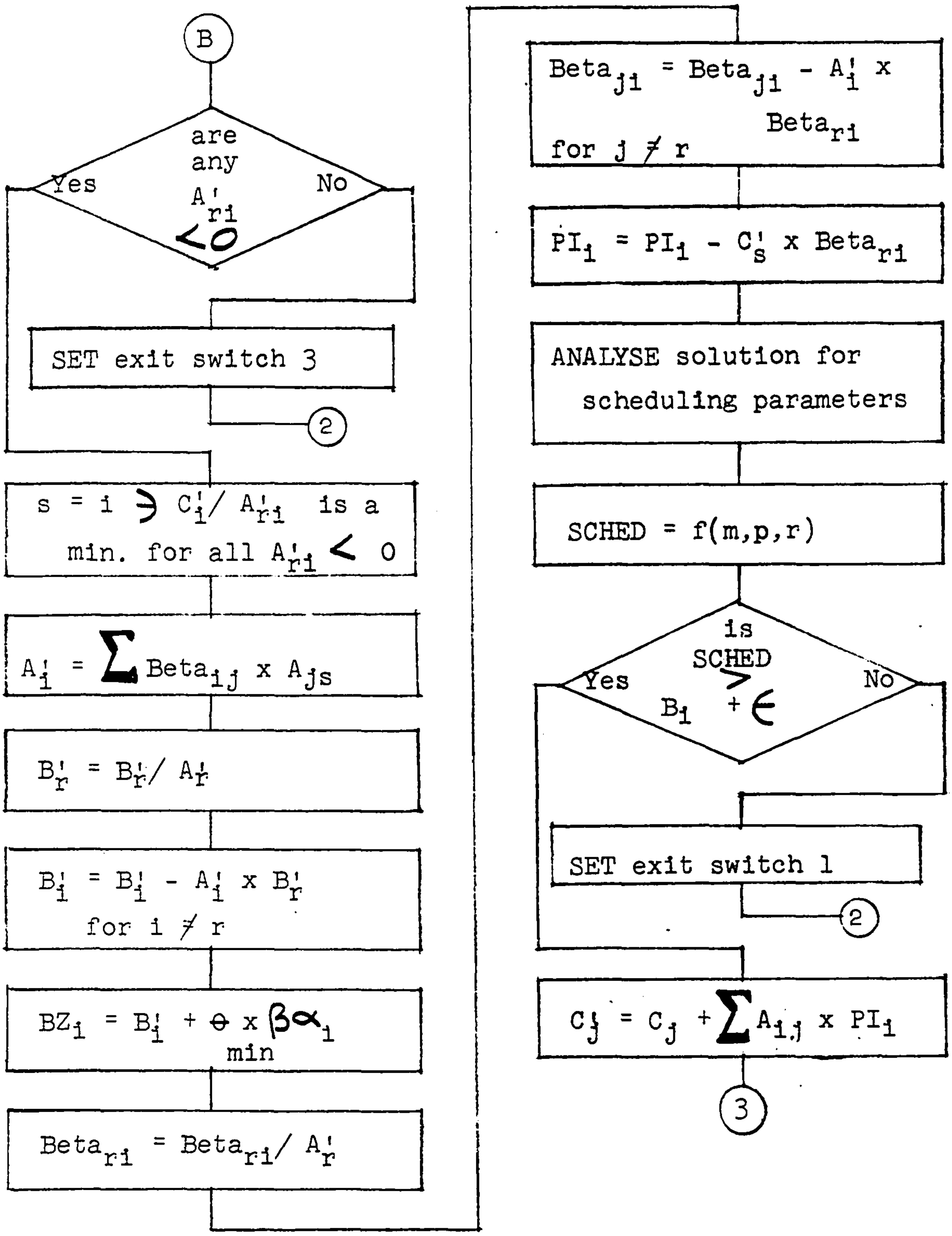


Fig. 32. Flow chart of P.L.P. program - part 3 of 4

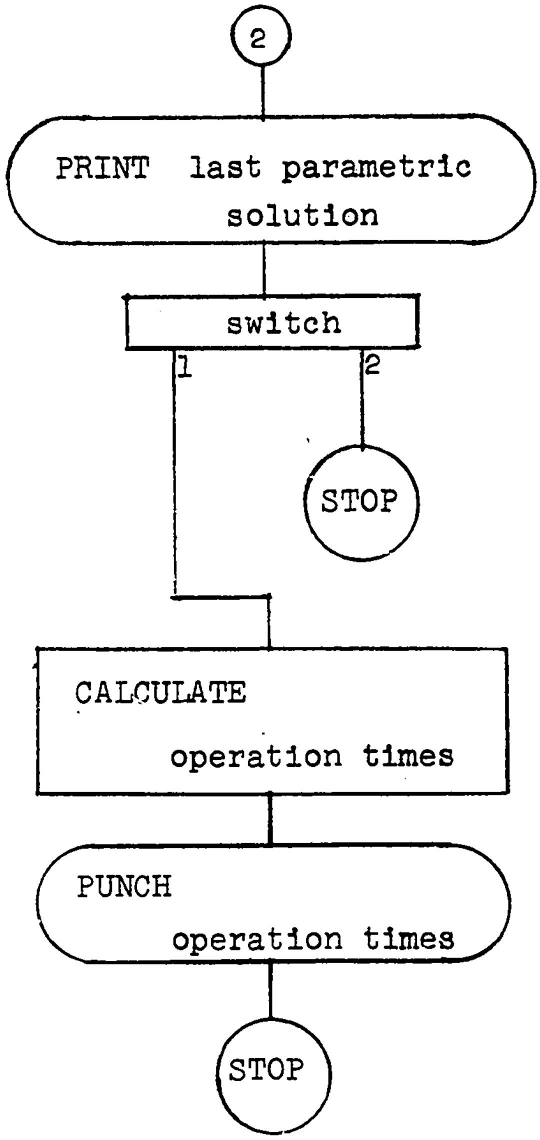


Fig. 32. Flow chart of P.L.P. program - part 4 of 4

```
C     PARAMETERIZED LINEAR PROGRAM SCHEDULING
      DIMENSION A(14,24),B(14),IX(14),C(24),BETA(14,14),CP(2
     14),PI(14),AP(14),BP(14),RATIO(24),BOLA(14),ALPHA(14),
     2THETA(14),ARP(24),BALPHA(14),P(14)
      READ 1410,JIM
 1410 FORMAT(I2)
      DO 1414 JAKE=1,JIM
      READ 1,M,N,K,MO,MA
    1 FORMAT(5I2)
      T=M
      DO 2 I=1,M
      READ 3,A(I,1),A(I,2),A(I,3),A(I,4),A(I,5),A(I,6),A(I,7
     1),A(I,8)
    2 READ 3,A(I,9),A(I,10),A(I,11),A(I,12),A(I,13),A(I,14),A
     1(I,15),A(I,16)
    3 FORMAT(8F6.1)
      READ 3,(B(I),I=1,M)
      READ 3,(C(I),I=1,K)
      CLARE=1.0
      JOHN=1
      JAY=1
      MARK=1
      TOM=0.0
      THE=0.0
      KEN=0
      KR=2
```

```
      JUD=1
      JO=0
      DO 4 I=1,M
      NO=N+I
      DO 4 J=1,M
      IF(J-I) 41,40,41
   41 A(J,NO)=0.0
      BETA(J,I)=0.0
      GO TO 4
   40 A(J,NO)=1.0
      BETA(J,I)=1.0
    4 CONTINUE
      DO 7 I=1,M
      THETA(I)=0.0
      ALPHA(I)=0.0
      IX(I)=I+N
      BP(I)=B(I)
      BALPHA(I)=0.0
    7 PI(I)=0.0
      DO 6666 I=1,K
 6666 C(I)=-C(I)
  100 DO 8 J=1,K
      CP(J)=0.0
      DO 88 I=1,M
   88 CP(J)=CP(J)+A(I,J)*PI(I)
    8 CP(J)=CP(J)+C(J)
```

```
      DO 47 I=1,K
      IF(ABSF(CP(I))-0.99E-06)48,48,47
   48 CP(I)=0.0
   47 CONTINUE
      GO TO (2222,77),JAY
 2222 IT=1
      DO 9 I=2,K
      IF(CP(1)-CP(I))9,9,10
   10 CP(1)=CP(I)
      IT=I
    9 CONTINUE
      IF(CP(IT))11,12,12
   11 DO 13 I=1,M
      AP(I)=0.0
      RATIO(I)=0.0
      DO 13 J=1,M
   13 AP(I)=BETA(I,J)*A(J,IT)+AP(I)
      GO TO (1776,27),JAY
 1776 DO 14 I=1,M
      IF(AP(I))34,34,15
   15 RATIO(I)=BP(I)/AP(I)
      GO TO 14
   34 RATIO(I)=99.0E49
   14 CONTINUE
      KT=1
      DO 16 I=2,M
```

```
   IF(RATIO(1)-RATIO(I))16,16,17
17 RATIO(1)=RATIO(I)
   KT=I
16 CONTINUE
   IF(RATIO(1)-99.0E49)27,26,27
26 PRINT 33
33 FORMAT(1X,17HNO POSITIVE RATIO)
   GO TO 1414
27 IX(KT)=IT
   BP(KT)=BP(KT)/AP(KT)
   DO 31 I=1,M
   IF(I-KT)32,31,32
32 BP(I)=BP(I)-AP(I)*BP(KT)
31 CONTINUE
   DO 18 I=1,M
18 BETA(KT,I)=BETA(KT,I)/AP(KT)
   DO 19 I=1,M
   DO 19 J=1,M
   IF(KT-I)20,19,20
20 BETA(I,J)=BETA(I,J)-AP(I)*BETA(KT,J)
19 CONTINUE
   DO 21 I=1,M
21 PI(I)=PI(I)-CP(IT)*BETA(KT,I)
   CLARE=CLARE+1.0
   IF(CLARE-50.0)100,100,44
44 PRINT 45,JAKE
45 FORMAT(1X,2HLP,I3,1X,7HCYCLING)
```

```
      GO TO 1414
   12 SUM=0.0
      DO 22 I=1,M
      LO=IX(I)
   22 SUM=SUM+C(LO)*(BP(I)+THE*BALPHA(I))
      DO 23 I=1,M
      LO=IX(I)
   23 PRINT 24,IX(I),C(LO),BP(I),BALPHA(I)
   24 FORMAT(1X,1HX,I2,F6.1,2E14.8)
      PRINT 25,SUM,THE
   25 FORMAT(1X,2E14.8)
      JAY=2
      GO TO (95,1414,1212),JOHN
  609 DO 412 I=1,M
  412 ALPHA(I)=-0.01*B(I)
      JUD=1
   77 DO 50 I=1,M
      BALPHA(I)=0.0
      DO 50 J=1,M
   50 BALPHA(I)=BETA(I,J)*ALPHA(J)+BALPHA(I)
      JOHN=2
      NOW=0
      DO 51 I=1,M
      IF(BALPHA(I))52,53,53
   53 THETA(I)=9.9E48
      GO TO 51
   52 THETA(I)=BP(I)/ABSF(BALPHA(I))
```

```
      NOW=1
   51 CONTINUE
      IF(NOW)91,91,92
   91 PRINT 93,JAKE
   93 FORMAT(1X,12HOPEN PROGRAM,I4)
      JOHN=3
      JERK=3
      GO TO 12
   92 KT=1
      THE=THETA(1)
      DO 54 I=2,M
      IF(THE-THETA(I))54,54,55
   55 THE=THETA(I)
      KT=I
   54 CONTINUE
      JERK=1
      GO TO (1060,95), JUD
 1060 DO 63 I=1,K
   63 RATIO(I)=9.0E48
      KID=0
      DO 56 I=1,K
      ARP(I)=0.0
      DO 56 J=1,M
   56 ARP(I)=ARP(I)+BETA(KT,J)*A(J,I)
      DO 2243 I=1,K
      IF(ABSF(ARP(I))-0.99E-06)2244,2244,2243
```

```
2244 ARP(I)=0.0
2243 CONTINUE
     MARK=MARK+1
     IF(MARK-40)713,713,912
 912 PRINT 932
 932 FORMAT(1X,14HNOT CONVERGING)
     GO TO 1414
 713 DO 57 I=1,K
     IF(ARP(I))58,57,57
  58 RATIO(I)=CP(I)/ABSF(ARP(I))
     KID=1
  57 CONTINUE
     IF(KID)73,73,59
  73 PRINT 74
  74 FORMAT(1X,19HPARAMETRIC SOLUTION)
     JOHN=3
     GO TO 12
  59 IT=1
     DO 60 I=1,K
     IF(ARP(I))62,60,60
  62 IF(RATIO(1)-RATIO(I))60,60,61
  61 RATIO(1)=RATIO(I)
     IT=I
  60 CONTINUE
     JAY=2
     GO TO 11
```

```
   95 JOHN=3
      SUMMA=0.0
      DO 300 J=1,MO
      BOLA(J)=0.0
      DO 1918 I=1,M
      LO=IX(I)
      IF(C(LO))302,1918,302
  302 BOLA(J)=BOLA(J)+A(J,LO)*(BP(I)+THE*BALPHA(I))
      DIV=DIV+1.0
 1918 CONTINUE
      SUMMA=SUMMA+BOLA(J)
  300 CONTINUE
      DO 400 I=1,M
      P(I)=0.0
      LO=IX(I)
      IF(C(LO))402,400,402
  402 DO 403 J=1,MO
  403 P(I)=P(I)+A(J,LO)*(BP(I)+THE*BALPHA(I))
  400 CONTINUE
      BOY=BOLA(1)
      DO 500 I=2,M
      IF(BOY-BOLA(I))502,500,500
  502 BOY=BOLA(I)
  500 CONTINUE
      PA=P(1)
      DO 560 I=2,M
```

```
      IF(PA-P(I))503,560,560
  503 PA=P(I)
  560 CONTINUE
      SCHED=0.8*(PA+BOY-SUMMA/DIV)
      PRINT 332,JAKE,SCHED
  332 FORMAT(1X,11HSCHEDULE NO,I3,1X,6HEQUALS,E14.8)
      GO TO (6868,8686),KR
 8686 IF(SCHED-B(1))340,340,99
  340 JERK=1
      KR=1
      THE=SCHED/B(1)*THE
      GO TO 95
 6868 IF(ABSF(SCHED-B(1))-0.02*B(1))12,12,6870
 6870 THE=THE+((SCHED-B(1))/B(1))*THE
      KEN=KEN+1
      IF(KEN-40)95,95,912
      GO TO (1060,609,1414),JERK
 1212 DO 600 I=1,M
      LO=IX(I)
      IF(C(LO))601,600,601
  601 DO 603 J=1,MO
      TIME=A(J,LO)*(BP(I)+THE*BALPHA(I))
      MIME=TIME
  603 PUNCH 602,MIME,JAKE,J,LO
  602 FORMAT(12X,I4,20X,I4,I4,I4)
  600 CONTINUE
```

```
      GO TO (1414,95,1414),JERK
 1414 CONTINUE
      PRINT 1415
 1415 FORMAT(1X,14HEND OF PROGRAM)
      STOP
      END
```

APPENDIX B

PARAMETRIC LINEAR PROGRAMMING

There are two methods of parametric linear programming described in the literature. One is to vary the cost coefficients and the other is to vary the constraints (Garvin 22). Due to the duality of linear programming the methods are equivalent. Since the hour constraints were of concern in this dissertation, the latter method was utilized.

The computer program (see Appendix A) utilized in the dissertation was taken directly from Garvin's text and only a graphical explanation will be presented at this time. Figure 33 is a two dimensional linear programming problem in which two products are made using four machines with the following constraints:

	Part 1	Part 2	Constraint
Machine 1	5	6	30
Machine 2	7	4	28
Machine 3	10	0	30
Machine 4	0	7	28
Profit	\$1	\$1	

The adding of a scheduling constraint that all production must be produced within 30 hours shifts the L.P.

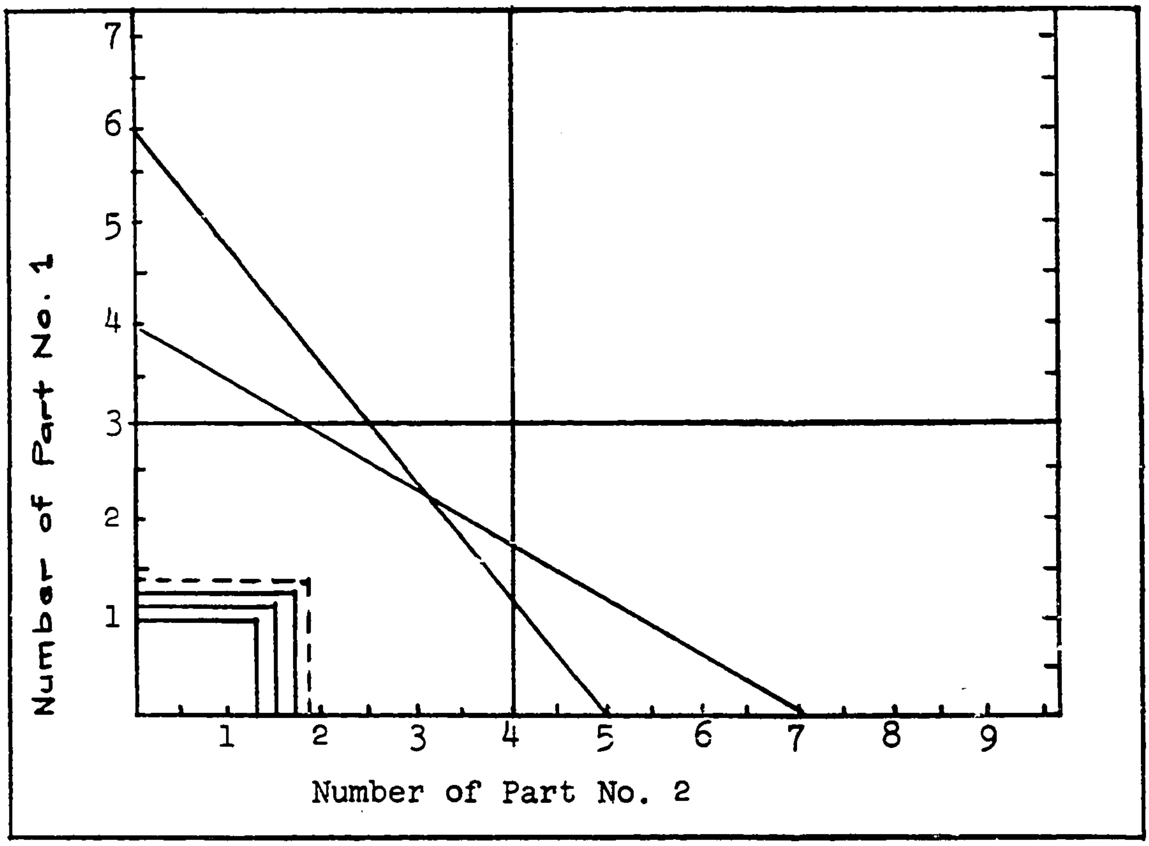


Fig. 31. Simple Parametric Linear Program

solution and the area of maximum profit becomes a rectangle with the greatest profit at the corner furthest from the origin. This point is then evaluated for scheduling time using equation 5.4. If the time is greater than the allowed 30 hours, the convex set is shrunk toward the origin. In the two dimensional case, each of the dotted lines in Figure 33 are moved parallel to themselves toward the origin. A new linear programming solution is found and the scheduling time is again evaluated. When a scheduling time equal to or less than 30 hours is found the parametric linear programming is terminated.

A detailed study of the above program indicates that there are many points in the set at which the scheduling time can be met, but the profit valuation varies and at the present time there seems to be no simple searching method for finding the optimum profit schedule which can be met. In the exploratory experiments made, it was found that the schedule time converged rapidly with only small changes in constraints and that the profit for the schedule was near optimum. Further experimental and theoretical research in this area may improve this technique so that it can be used for optimization of other types of non-linear programs.