A PRODUCTION SCHEDULING SIMULATOR

by

HARIPADA NANDA

B. Sc. (Engineering) Mechanical, Utkal University, India, 1961

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1963

Approved by:

George F. Schrader
Major Professor

LD
2668
T4
1963
N36
Spec. Coll.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

INTRODUCTION

In the last few years in the study of operations research simulation has provided a ground floor for research and experimentation. Simulation models have been constructed to analyze complex industrial systems with interacting factors where usual mathematical techniques have not been proved to be very successful. Simulation as a general field of activity has to do with design, building and manipulation and study of models (3). The models used for simulation purposes are of two types: analytical models and training models. The use of analytical models has been for mathematical analysis, design, forecast, and do not include human decision makers. Training models are intended to teach a group of decision makers the behavior of the simulated system. A simulation exercise using a training model has been called a game. Two major types of games are in use. The first type is designed to provide the participants the overall picture of the system and to develop within them an understanding of the basic organizational relationships. They are interactive, competitive and deal with decision-making at the top management level. The second type of games are called "Functional" games. They are designed to provide participants with experience of dealing with one or more kinds of problems in a specific company (3).

A functional game for scheduling production facilities has been the subject of this thesis. The aim was to design a decision-making exercise for scheduling a job-shop by assigning relative priority among jobs. A job-shop of eight unlike machines was selected for such a purpose. The interval of decision-making was selected to be a five-day-week consisting of two eight-hour shifts per day. A group of trainees were assumed to play this exercise at the beginning of each decision-interval.

The job-shop was scheduled for a total period of eighty hours taking into consideration the relative priorities. A number of random discontinuations* in processing were provided to alter the actual working at the above schedule. At the end of each decision interval a report stating the status of the job-shop throughout the week was assumed to be handed over to the players. The report consisted of a statement of the outcome of the scheduling decisions in terms of a brief income statement along with a list of jobs delayed, total idle machine hours and number of other useful informations.

For the above procedure, a digital computer algorithm was formulated. The computer programs were written in the following steps.

Program I:

(a) Generating number of orders per week.

(b) Determining machine sequences of each new order.

(c) Determining processing time corresponding to each sequence.

(d) Determining material cost pertaining to each job.

(e) Determining the number of break downs in processing.

Program II

(a) Scheduling the job-shop.

(b) Locating the starting hours of random break downs.

(c) Determining the duration of each of such break downs.

(d) Transforming the schedule due to break downs.

(e) Computing the pertinent costs, prices, profits to serve as reports to the players.

---

* The term break down of machines has been used alternatively.

All calculations pertaining to the simulation were carried out on the IBM 1620 digital computer, using the Fortran II programming system.

JOB-SHOP SCHEDULING

A job-shop presents a situation of sequencing processing of a number of jobs through a limited number of facilities. A job spends a period of time in some or all facilities. Processing time is established according to time standards. The predetermined time to be spent by a job on a particular machine may be subjected to random fluctuations such as: material shortage, worker absenteeism, break down of machines, etc. The aim of research has been to opitimize the use of facilities effectively in processing jobs, so as to satisfy a certain measure of effectiveness. Effectiveness is measured in terms of minimum cost, maximum profit, minimum processing time, meeting due dates, etc., which ever is most appropriate, with knowledge of the data pertaining to each job a decision-maker faces the problem of scheduling. Scheduling is to establish the order of processing of jobs through available facilities.

A number of researchers (9, 10) have attempted to solve the scheduling problem by thermodynamic, combinational, integer-linear programming and waiting line approaches. The integer-linear-programming approach for solving general classes of scheduling problems has been introduced by Wagner (12) and further developed by Bowman (1), Manne (7), and Wagner (12) himself. This technique seems to converge to an optimum solution, but computatiohal aspects of these problems have not been feasible through modern computing machines.

The impossibility or the partial possibility of solving scheduling problems by the help of mathematical rigor has suggested itself to be attempted from the angle of simulation. Sisson (10) divides this kind of simulation into three major classes: A, B, and C. In approach A the number of sequences to be examined is reduced to a number of feasible cases by combinational analysis. Through proper recognition of given routings and equivalence of various sequences, the total number of combinations can be reduced to a number of feasible cases which contain an optimum solution. From these limited cases Monte Carlo sampling is employed to pick the optimum with a certain degree of confidence. This method is due to Giffer and Thompson (3, 4).

In case B, no algorithm is used to eliminate non-optimum schedules, but the optimum can be picked with any confidence interval desired, by adopting a suitable sampling procedure. This approach is due to Heller (5).

The approach C, as described by Sisson, (10) is the most general simulation approach. The whole job-shop situation is represented by a computer program. Different priority rules describing relative importance of jobs at different points in time are formulated. From a number of jobs waiting to be processed by any machine, the one with topmost priority is assigned to the machine.

Rowe (8) has developed six different priority rules. A priority rule calculates the priority index of a job and a scheduling decision is made as above, when a facility becomes free. By simulation in a computer Rowe (8) found priority-decision rules described below to give good results.

1. First come first served within same class of jobs, classified according to their dollar values. When there is a choice between the two value classes, the one with the highest value is assigned first. Rowe (8) found this rule to be suitable to meet due dates closely.

2. Processing the next job in the queue with lowest processing time. This rule tends to minimize total expected waiting time.

Looking at scheduling as an activity the most practical approach would be to schedule for achieving maximum profit or minimum cost in the long run. This will need an overall view of the scheduling operation involving consideration of different factors giving rise to costs. A good blend of all such factors from overall viewpoint of the job-shop should lead close to the optimum. In the present state of scheduling research, it will be worthwhile to build up the intuition of human decision-makers for the scheduling operation. This thesis was aimed at developing a job-shop scheduling exercise to impart the idea of relative importance of jobs in a job-shop.

THE MODEL

The model presented assumes a job-shop consisting of eight machines scheduled for a period of a five-day-week. Each day consists of two eight-hour shifts.

The scheduling operation done by computer is based on the following assumptions.

1. No machine can process more than one job at a time.
2. Each operation once started, must be performed to completion.
3. A known, finite time is required to perform each operation and each operation must be completed before any operation which it must precede can begin.
5. Each job must be processed by designated sequences of operations.
6. The sequences are such that a job goes through a machine only once.
7. There is only one of each type of machine.

8. The time required to move jobs between operations is negligible.
9. The incoming jobs do not have any fixed delivery time. A job enters the shop at the beginning of processing and leaves the shop after it is processed completely.
10. In the intervals between subsequent operations a job is stored in-between the processing facilities. In-process inventory is assumed to be unlimited.

FACTORS CONSIDERED

Jobs

The job-shop receives a set of new orders each week. The new jobs, along with the jobs carried over from the previous week, are put to processing in a week's period. Weekly demand has been considered to vary from a maximum of 15 orders to a minimum of 12 orders. There are equal chances of receiving any number of orders ranging from 12 to 15. The weekly demand may be affected by the deliveries committed in the week previous to the week under consideration. It has been assumed that the actual demand in any week will be reduced by the fraction of the jobs delayed in the previous week.

$$\text{Actual weekly demand} = \text{Random orders} \times \frac{\text{Jobs delivered in the previous week}}{\text{Jobs handled in the previous week}} .$$

Sequences

A job can go through three to six operations, any number of operations within this range having an equal chance of occurrence. The sequence of processing a job is independent of the sequences of the other jobs. There is equal chance of a job being processed by any of the eight machines at any stage of its processing.

Processing Times

Processing time of a job in any operation is considered in units of hours and is likely to vary in between three hours and eight hours. There is equal chance of occurrence of any length of hours in this range.

Shifts

A day consists of two eight-hour shifts, first eight hours standing for the first shift and the next eight hours for the second shift. The week is described in terms of hours instead of days; so the first day of the week consists of the first sixteen hours, the second day the next sixteen hours, etc.

Break Downs

Discontinuities in processing have been named as break downs. Such a discontinuity in processing can happen due to shortage of materials, break down of machines, worker absenteeism and a number of other interruptions in the manufacturing processes. Break downs per week were assumed to be poisson distributed with a mean of three break downs per week. Also a poisson distribution with a mean of four hours per break down was assumed to describe the duration of break downs. A break down is likely to occur in any machine at the hour when the machine is in operation.

Delayed Jobs

Any job which either cannot be scheduled in a week's time or cannot be completed due to break down is delayed and carried over to the week next to the week under consideration. As mentioned earlier, jobs delayed in a week create a customer dissatisfaction which goes to reduce the subsequent demand.

This consideration has been considered to be optional, depending upon the method of play the Director chooses. The policy of the job-shop is to lose the profit on any job delayed for the first time. The selling price for the same job is reduced by one quarter of its previous price for each additional week of delay.

Idle Time

Any time a processing facility is without a job, it is said to be idle. Idle hours are almost unavoidable due to the sequencial processing in a job-shop. Idle hours existing in the prepared schedule may increase due to break downs in the manufacturing process. Idle time represents unproductive hours of the job-shop resulting in an opportunity cost or a loss of profit.

Costs

In a job-shop situation cost can arise due to a number of factors. Here, five basic costs were taken into consideration. They are: (1) over-head cost, (2) labor cost, (3) processing cost, (4) material cost, (5) in-process inventory cost.

(1) Over-head cost arises due to over-all plant services; here a job-shop over-head of $800 per week has been chosen.

(2) The labor force has been assumed to be fixed. The labor rate per hour has been fixed to be $3.00 for the first shift and $4.00 for the second shift. Under the assumption of eight workers working in a shift, the total labor cost has been fixed at $2,240 per week.

(3) Besides the pay of workers, running cost of each machine has been assumed to be $2.00 per hour. This cost arises only when the machine is in operation.

(4) The materials of different jobs vary in nature, composition and volume. However, the jobs have been divided into three groups according to the cost of materials. The first, second and third groups consist of jobs with high, medium and low material cost, respectively. Under the assumption that the material of a job can be of any type irrespective of the operations, a job can fall into any of the above groups. The material costs of high value, medium value and low value jobs have been fixed at 100, 75 and 25 dollars, respectively.

(5) For the job-shop under study, it has been assumed that a job once started in any machine, does not go out of the shop unless all the operations on it have been finished. In between the operations a job is stored in the space between the processing facilities. A cost of 10 cents per hour per job has been arbitrarily chosen to account for in-process inventory.

Total cost is the sum of all the individual costs mentioned above.

Total cost = over-head cost + labor cost + processing cost + material cost + in-process inventory cost.

Leaving aside all the above costs a few opportunity costs arise in a job-shop situation due to idle machine hours. Out of them the cost needing most important consideration is the cost of wasted labor. This has been named here as unutilized labor cost.

Total unutilized labor cost in dollars

$$= 2240 - 3 \times \text{first shift hours} - 4 \times \text{second shift hours}.$$

Price

The selling price of any job is calculated by a pricing rule. In order to apply the pricing rule, the jobs are classified into two categories: undelayed jobs and delayed jobs. The price of any undelayed job is taken to be two times the estimated cost to produce the job.

The cost of a job is estimated on the basis of the over-head cost, material cost, processing cost and the average cost of the labor hours to be spent to produce the job.

$$\begin{aligned}\text{Estimated cost in dollars} &= \text{material cost} + \text{processing cost} + \text{labor cost} \\ &\quad + \text{over-head cost per each new order} \\ &= \text{material cost} + 2 \times \text{total processing hours} \\ &\quad + \frac{3+4}{2} \times \text{total processing hours} \\ &\quad + \frac{800}{\text{Number of new orders}} .\end{aligned}$$

Price of an undelayed job in dollars = 2 × estimated cost.

The price of a delayed job, delayed for the first time, is estimated to be the estimated cost of the job. For any subsequent delay, the price is reduced by one quarter of the price fixed for the same job in the week before.

THE PROGRAM

As already mentioned in the Introduction, the computer program was written in two parts. Program I or the first part of the program was written to generate scheduling data for one decision interval. The purpose of Program II or the second part of the program was to determine the results of a play of the game.

A brief outline of the method of computation adopted in each of the programs is described in this section. The detailed program in Fortran language along with flow charts are included in the Appendix.

Program I

This program goes to determine (1) the orders, (2) sequences, (3) processing times, (4) material cost corresponding to each order and (5) the number of break downs per week.

Determining number of orders. A set of random number intervals were defined to specify a particular number of orders, so that an equal number of random numbers belong to each interval.

Random number interval	Orders
0000 - 2499	12
2500 - 4999	13
5000 - 7499	14
7500 - 9999	15

In order to determine the number of orders, a four digit random number is matched against the list shown above. The number of orders corresponding to the interval to which the random number belongs is determined.

The number of orders determined as above is reduced by a constant of customer dissatisfaction to compute the actual demand of a week.

$$\text{Actual demand} = \text{orders} \times \text{constant of consumer dissatisfaction.}$$

Following the notations of the program;

$$NEUJB = DD \times \frac{JBPRV - LTJOB}{JBPRV}.$$

where NEUJB = Number of new orders.

DD = Orders determined from the table.

JBPRV = Total number of jobs handled in the previous week.

LTJOB = Number of jobs delayed in the previous week.

Determining Sequences. At the first step, the number of operations per each job is determined by a set of random numbers. The machine sequences for each operation is determined arbitrarily from a group of one digit random numbers.

Random number interval	Number of operations
0000 - 2499	3
2500 - 4999	4
5000 - 7499	5
7500 - 9999	6

The above table shows a list of random number intervals assigned to different numbers of operations. For each job a random number of four digits is matched against these intervals to determine the number of operations it has to go through.

According to the assumptions, a job can go through any arbitrary machine in any of its operations. The machine corresponding to any operation is determined by reading in one digit from a random number card. As a machine number of zero is an impossibility and a number of nine is not consistent with the assumption of eight machines, only the numbers ranging from one to eight are accepted as job sequences.

After all the sequences of a job are determined, they are arranged in a matrix form. In the program the sequence matrix is denoted by KSEK, where rows represent job numbers and the columns represent the operation numbers. KSEK (I, J) denotes a machine in which the J^{th} operation of the I^{th} job is performed.

Determining Processing Times. Processing time is determined for each sequence by assigning random number intervals to different processing hours. The technique adopted is the same as explained before in this section.

Random number interval	Processing time in hours
0000 - 1665	3
1666 - 3331	4
3332 - 4997	5
4998 - 6663	6
6664 - 8329	7
8330 - 9999	8

The processing times are arranged in a matrix form, the rows and columns of the matrix represent job numbers and operation numbers respectively. In the program, this matrix is denoted by NPT. NPT (I, J) stands for the processing hours of the I^{th} job in its j^{th} operation.

Determining Material Cost. Random number intervals are assigned to different material costs as listed below.

Random number interval	Material cost
0000 - 3332	50
3333 - 6665	75
6666 - 9999	100

A random number of four digits is matched against these intervals to determine the material cost for any job.

Determining Number of Break Downs. The number of break downs in any week is determined on the assumption that a fractional random number represents the cumulative poisson probability of the number of break downs.

Each block in Fig. 1 shows the probability of occurrence of a particular number of break downs written under the block.

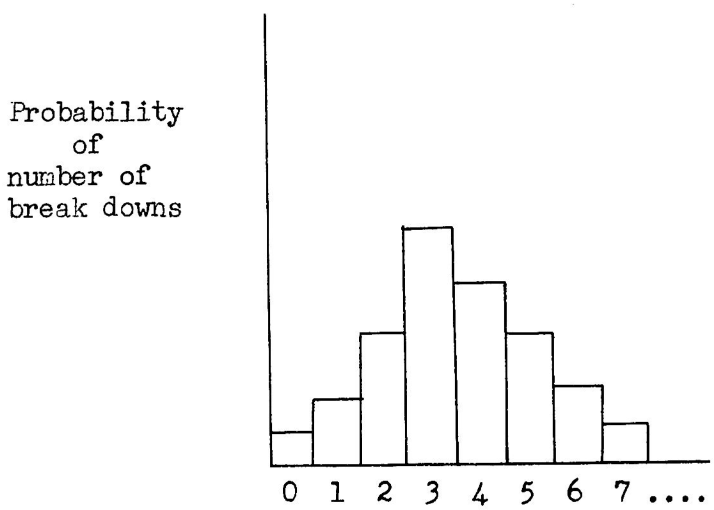


Fig. 1. Number of break downs. XBD

The following procedure is adopted to compute the number of break downs. A random number of four digits is read in. This number is divided by 10,000 to convert it to a fraction. The next operation is to add the areas of the successive blocks in the sketch starting from left to right, until the total area is equal or a little greater than the fractional random number. The corresponding value of the number of break downs determines the break downs per week.

At the end of the program, the total number of jobs handled in a week is determined by adding the new orders with the jobs carried over from the week previous to the week under consideration.

Input/Output.

Table 1. Data Cards

Card group number	Number of cards	Description	Instruction
1	1	Number of jobs carried over	Punch in first 3 columns
2	1	Total number of jobs handled in the previous week	Punch in first 3 columns
3		A random number deck	Prepare a deck with 30 to 40 random digits punched in the first four columns of a card.

Table 2. Punch Output

Card group number	Number of cards	Description	Instruction
1	1	Statement "Jobs Ordered This Week" with new orders punched in columns 27-29	
2	1	Title card	
	Equal to the New Orders	Sequences of all the new orders	Preserve these cards for running with Program II
3	1	Title card	
	Equal to the New Orders	Set of cards representing the processing time data	Preserve these cards for running with Program II
4	1	Number of break downs punched in columns 1-3	Preserve this card, in case break downs are taken into consideration

Program II

The purpose of Program II was (a) to schedule the job-shop for a period of eighty hours, (b) to provide a number of break downs over such a schedule and to transform the schedule accordingly, and (c) to report the results of the game to the participants. The computing procedure adopted in each part is explained separately.

Part (a). In practice a job-shop schedule is represented by a Gnatt Chart. The Gnatt Chart is a graphical representation of a schedule in which the bars represent machine time drawn to a particular scale. The computing technique adopted here for scheduling utilizes a numerical representation of the Gnatt Chart. This technique is explained briefly in the following lines.

A hypothetical schedule of 3 machines for a period of six hours is shown in Fig. 2 in the form of the Gnatt Chart in which the shaded bars represent machine time and the blank ones represent idle time. The same idea of the schedule could have been imparted by drawing Fig. 3.

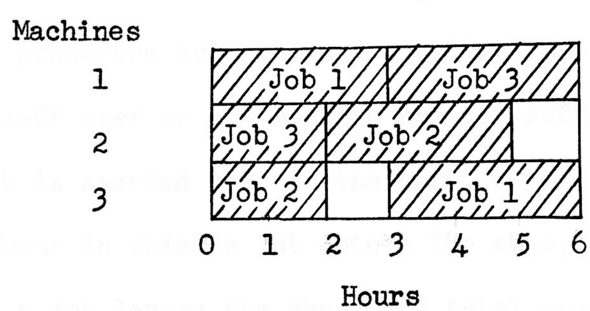


Fig. 2. A Gnatt Chart

Machines	Hours 1	2	3	4	5	6
1	1	1	1	3	3	3
2	3	3	2	2	2	0
3	3	2	0	1	1	1

Fig. 3

The rows of Fig. 3 represent machines and the columns represent hours. To signify the presence of any job on a machine in a particular hour, the job number is entered in the cell corresponding to the machine and hour. Idle time of a machine in any hour is represented by a cell-entry of zero.

This program was written to develop a schedule for eight machines for a total period of 80 hours utilizing a matrix of dimension 8 × 80. This matrix has been named the Original Schedule.

The following procedure is followed for scheduling: The job with highest priority number is scheduled over all its operations by assigning the job number to the corresponding cells of the scheduling matrix. The job next in importance is scheduled in the remaining blank hours or the blank cells of the matrix. This procedure is continued until all the jobs have been scheduled. If the hours left over on any machine are not sufficient for the completion of a job, the job is carried over to the next week. Records of the starting hour denoting the hour in which a job enters the shop, the ending hour denoting the time in which a job leaves the shop, and total processing hours are kept for each job. The ending hour of a delayed job is taken to be the last hour or the eightieth hour of the week.

Part (b). The aim of this part of the program was to locate a few random break downs over the existing schedule and effect changes over such a schedule.

Provision of Break Downs. At the outset, a matrix of the same size as the original schedule is formed. To locate break downs in such a matrix the following procedure is adopted.

For each break down the starting hour, duration and the machine where it occurs, are determined by reading in random numbers. A non-zero random number of two digits, with its value limited to 80, is taken as the starting hour of break down. A non-zero one digit random number not exceeding eight in value, determines the machine where the break down has occurred. Care has been taken to locate the starting hour of a break down, only when there is a job on the machine. Duration of any break down is determined from a poisson distribution with a mean of four hours. After determining the data pertaining to break downs they are located in the break down matrix by assigning -50 to the corresponding cell values.

Transformation of the Original Schedule. Due to the occurrence of break downs the original schedule goes through changes. Program was written to make these changes, such that the same sequences of jobs are maintained. A new matrix of 8×80 dimension is formed to carry the transformed schedule.

1. The Original Schedule does not undergo any change until the first hour of the first break down. In case there is no break down, this part of the program is skipped through.

2. All the cell values of the break down matrix are compared against the corresponding cell values of the Original Schedule. In case of a break down occurring in any hour of a machine, the machine's schedule is moved to the right by one hour until an idle hour is met to balance this. After such an idle hour is reached, the schedule of the particular machine does not change.

The movement of the schedule of one machine may cause a job to appear with two or more machines in the same hour. Such an occurrence presents an impossibility, showing that the job is present at more than one machine. Therefore, when a job occurs on more than one machine in the same hour, the policy is to identify the machine showing the earliest operation on the job. The schedules of the rest of the machines are moved to the right by one hour.

The above procedure is repeated, until all the cells of the break down matrix have been compared with the corresponding cells of the original schedule.

In any shifting operation, if any job is moved beyond the eightieth hour, the total processing time spent on the job is reduced by one hour and the job is carried over to the next week.

Part (c). This part of the program serves to compute relevant informations as regards the status of the job-shop through one week.

At the outset, the time spent in processing in different shifts is calculated. This computation is carried out by examining the cell values of the final scheduling matrix, and adding one, each time a positive number appears. The successive addition, mentioned above is carried out in intervals of eight hours, where the intervals represent first shift and second shift, alternatively. Following a similar procedure, as above, total idle time is computed by identifying the cells of the scheduling matrix with zeros.

The next step in computation, is to determine the price of each job with the information of material costs and processing times being read in. For all new orders, the jobs are identified as delayed and undelayed jobs, and the prices are determined using the pricing rules already mentioned. The same procedure is carried out for jobs, carried over from the previous week, with the exception that, instead of using material cost, etc., as data the prices determined in the previous week is adjusted by a pricing rule to determine the new price for each job. Prices of all jobs, delivered in the current week are summed up to give the Total Revenue earned.

The computation of the revenue is followed by the determination of a number of costs. Cost of the in-process inventory is calculated by considering the total number of hours spent by the jobs in-between the processing facilities. All the costs, incurred during the week's period, are summed up to determine the total cost incurred in the week. It may be noted here that an estimation of unutilized labor is done by figuring the difference between the total pay and workers and the cost of utilized labor hours. This is not taken into consideration in the total cost calculation, it serves to give an idea of labor utilization only.

The difference between Total Revenue and total cost is computed to determine the Net Profit which is further added to the Cumulative Profit brought over from the previous week. The result is a Cumulative Profit at the end of the week under consideration.

The following equations will explain the method of computation.

$$KOJB = 5.5\ KSUM + \frac{800}{NEUJB}$$

$KRVNU = 2\ KOJB$ For undelayed jobs

$KRVNU = KOJB$ For delayed jobs

$$KCOST = 800 + (8 \times 280) + 2(KSFT1 + KSFT2) + KTCST + KCINV$$

$$LCOST = 8 \times 280 - (3\ KSFT1 + 4\ KSFT2)$$

$$NPRFT = KSVNU - KCOST$$

where,

KOJB = Estimated cost of a job

KRVNU = Price of a job

KSUM = Total processing hours required to produce a job

KCOST = Total cost incurred in the week's period

LCOST = Cost of unutilized labor hours

NPRFT = Net profit

KSVNU = Total revenue earned in the week

NEUJB = Number of orders received in the week

KPRFT = Cumulative profit

Input/Output.

Table 3. Data Cards

Card group number	Number of cards	Description	Instruction
1	1	Total number of jobs handled in the current week	Punched in Cols. 1-3
2	1	New orders received	Punched in Cols. 1-3
3	Same as the total number of jobs	Sequences	Recovered from the output of Object Program I
4	Same as the total number of jobs	Processing time	Recovered from the output of Object Program I
5	1	Decision form	A job occupying any position in the decision form is punched in two columns
6	1	Number of break downs	Recovered from the output of Object Program I
7	New orders received	Material cost	Recovered from the output of Object Program I
8	No. of jobs carried over from the previous week	Price of the jobs	Recovered from the previous week's output
9	1	Cumulative profit	Recovered from previous week's output

Table 4. Punch Output

Card group number	Number of cards	Description	Instruction
1	1	Title card	
	38	Original schedule is punched in four groups. Each group consists of eight cards representing the schedule for intervals of twenty hours. One group is separated from the next by two blank cards	
2	1	Title card	
3	1	Title card	
	38	Final schedule is punched in four groups. The arrangement of cards is the same as mentioned under Card group 1.	
4	1	Title card	
	1	Idle time and the number of jobs delayed without considering break downs punched in Cols. 29-31 and 43-45, respectively	
5	1	Total machine hours in first shift and second shift punched in Cols. 30-32 and 51-52, respectively	
6	1	Idle hours caused due to break downs punched in Cols. 53-55	
7	1	Title card	
	1	A list of the delayed jobs punched from Col. 25.	
8	1	Title card	
	Equal to new orders	Each card contains the following information: Job No. punched in Cols. 21-22 Material Cost punched in Cols. 30-32 Total processing hours required to produce the job punched in Cols. 42-44 Price of the job punched in Cols. 55-57	Preserve the price cards of all delayed jobs, to be used as data for the next

Table 4. (continued)

Card group number	Number of cards	Description	Instruction
9	Equal to jobs carried over	Each card contains the following information: Job No. punched in Cols. 21-22 Material cost punched in Cols. 55-57	
10	1	Title card	
	1	In-process inventory cost punched in Cols. 27-30 Unutilized labor cost punched in Cols. 51-54	
11	1	Title card	
	1	Total revenue punched in Cols. 22-26. Total cost punched in Cols. 37-41. Net Profit punched in Cols. 54-58.	
12	1	Cumulative profit punched in Cols. 63-67	Preserve it for using as a data card for the next play of the game

Operating Procedure

A. Program 1.

1. Console settings

Parity	Stop
I/O	Stop
O'Flow	Program
Program Switch 1	Off
Program Switch 2	Off
Program Switch 2	Off
Program Switch 4	Off

2. (a) Zero memory - Type 16000 1000 000 Rs (Press Release and start alternatively). Press Instant-Stop, and then press Reset.

   (b) Load the object program in reader hopper. Press Load.

   (c) "Load Subroutines" Will be typed out on the typewriter. Then press Start.

   (d) After the object program is read in, "Enter Data" Will be typed out by the typewriter. Then load data cards as indicated under Table 1. Press Reader Start and then Press the Punch Start. Results will be punched out and a Manual Light will come out on the console indicating the end of the program.

3. Restart Procedures. The possibility of a halt occurring due to incorrect punching of data is very small as only two data cards are punched to run this program. However, in case a punching error is detected, data should be punched correctly and the following restart procedure may be tried.

   Press Reset and Insert. Type $490040\bar{9}$ RS. Reader No Feed light will appear on the console. At this stage all data cards can be reloaded again without loading the object program.

B. Program II.

1. Console Settings: The setting of console switches are same as adopted for the operation of Program I.

2. Steps (a) through (d) are same as the operation of Program I. Load data card groups 1-6 with a deck of random numbers, the same as used in the previous program.

(e) Follow the steps as directed by the console lights. After a number of cards are punched intermittently, Manual light will appear on the console.

At this step take out all data cards from the reader and press the Non Product Run Out on the reader unit to recover the last two cards. Then press Start.

The information "Load Material Cost, Price and Profit Data" will be typed out and Reader No Feed light will appear on the console. Then load data card groups 7-9 on the Reader Hopper and press Reader Start.

(f) After punching a few cards, the computer will again come to "Manual" mode. Press Start. The statement "Load Data for the Next Team" will be typed out and Reader No Feed light will appear on the console.

At this step recover all data cards staying on the reader unit. Then load data cards groups 5 and 6 where data card group 6 represents the decision form for the second team.

Steps (e) and (f) are repeated over and over again for computing results for each team.

3. Restart Procedures. In case of a halt or an error typed by the typewriter all the data cards are checked to see if they are in proper order and punched correctly. A few restart procedures are suggested.

(1) In case of an error detected in data card groups 1-5, Press Reset and Insert and type $490040\bar{9}$ Rs. Reader No Feed light will appear on the console. Load corrected data and press Reader Start.

(ii) If an error occurs in punching data card group 6, press Reset and Insert, and type 4912215 RS. Reader No Feed light will come out on the console. Load corrected data and follow on.

(iii) In case the data cards loaded in step (f) of the operation are not in order the information "Material Cost Cards are Out of Sequence" will be typed out. But this error will not stop the computation.

In this case, arrange the cards in proper order. Press Instant Stop, Reset and Insert. Type 4912925 RS. The information "Load Material Cost, Price and Profit Data" will be typed on the typewriter. Then, follow the procedure the same as Step (e).

INFORMATION FOR PLAYING THE GAME

Control of the Game by the Director

The play of the game is supposed to be conducted by a person, called the "Director".

Director Background. The director should be familiar with assumptions under which this simulator is written. In case of need, he should be prepared to change different parameters associated with the generation of demand, sequences, processing hours, material cost and also the pricing policy. This change would be necessary only when it is felt that changing one or more parameters will present a more realistic game to the players.

The following steps should be taken in controlling the playing of the game.

Briefing. The players should be provided with instructions for making decisions. It is the responsibility of the director to explain the day to day scheduling operation and its importance. An explanation of the various costs involved in a job-shop may help the players in making most valuable decisions. Players may be divided into different groups, alternatively a single individual may form a team.

Playing the Game. The game can be played in various ways depending upon the choice of the director. Two possible ways of play has been considered here.

1. The first and the foremost way to play this game will be to issue the same number of jobs to each player at the beginning of a week. Depending upon the decisions of the players the job-shop will be scheduled by the computer and the results will be issued at the end of the week. Any number of jobs delayed, will be carried over to the next week. Each player may have a different number of jobs carried over. The director will generate demand for the next week on the computor and issue them to the players. Here, the assumptions regarding the generation of demand, will be a little different. Demand will not be considered to depend upon the number of jobs delayed in the previous week. In order to be consistent with the computer program to generate demand, it is advisable to use zero numbers of delayed jobs as data. Again for every week, the number of break downs should be taken to be zero. A little discussion seems to be necessary at this point. The nature of a break down is such that it can only start at an hour when the machine is in operation. As a matter of fact, the schedule of one player is different from the other. If break down is considered to happen at any hour over a particular machine for the schedule of one player, the same hour and the machine considered may possibly be an idle time in another player's schedule. This supports the fact

that no comparison can be made between the weekly performance of player if break down is provided.

This way of playing the game is based on the fact that in any length of time, the total number of jobs issued is the same for all players and each player proceeds on with the result of its own decision.

This method of play is associated with two main advantages of a management game. That is, the performance of a player can be directly measured at any time by the cumulative profit earned by him. The other advantage is the creation of a good competitive atmosphere among players. The only disadvantage of this procedure is that this may result in too many delayed jobs each week.

2. The game can be played by a single player who may try to improve his decisions by playing it several times. Only in this case, all the provisions made in the program can be utilized fully. The result of decision of the player in one week, will affect the incoming orders in the next week by the delivery rule already discussed. In playing the game a number of break downs can be appropriately provided to test the decisions in a case of random break downs. This will provide the player with the experience of decision-making in a situation more or less practical.

The procedure of play will be exactly the same, as described under the first procedure, except that the break downs generated at random by the Program I can be utilized as a data while playing the game in the computer.

Basic Duties of the Director.* Each time the director has three basic duties to perform. They are: generating orders, issuing them to the players,

* However, in the Procedure 2 the player may be considered as his own director.

collecting player decisions, and handing over the results of such decisions to the players. The step by step instructions are listed below for conducting the game following any of the procedures mentioned before.

(a) The director has to run the order generation program in the computer to generate orders for a week. The data to be used will depend on the procedure to be adopted in playing such a game. The evaluation of performance of a player, or a group of players, should be the basis of the value of a measure of effectiveness. The usual measure of effectiveness adopted for such a play is the cumulative profit. Some other measure of effectiveness like total unutilized labor cost may also be a criteria of evaluation of the performance of players.

(b) From the generated data, he should take out the one with number of break downs punched on it. The rest of the cards are printed and one copy, accompanied by a decision form, is handed over to each team.

(c) The director will fix up a decision-making-time for the players. Within this time a player has to fill up the Decision Form explained under Section (d) below.

(d) Editing the Decision Form. A decision form is shown for the sample problem having 20 positions. Each position stands for the corresponding priority number. Each position is provided two rooms for entering a job number. A job assigned to any position will occupy both the columns. Any deviation from this procedure in filling up the decision form should be detected and replaced by a correct entry.

(e) The next job of the director is to organize data in the same order as mentioned under Input/Output list of Program II.

The following instructions should be borne in mind.

(1) If the game is played for the first time, a cumulative profit of zero should be taken.

(2) For any subsequent play of the game, the cumulative profit earned by the specific group is taken as data.

(f) The output of the results of the game for each team should be printed by a general purpose board and a copy of it should be handed to the team.

Under both the procedures the director has to determine the remaining operations and the corresponding processing times of each delayed job by comparing it with the data sheet of the previous week. This will be much easier to determine, if the sequences are checked starting with the last and proceeding forward. This job may be assigned to the players, the director being only responsible to check the correctness of the sequences and processing times so determined.

Critiquing. After playing the game for eight to ten times, the director may hold a critique session. This will include a discussion of experiences earned during the play of the game. The director may be asked questions regarding the model of the game. Some parameters of the game, like the demand, number of operations per job, pricing rules and different unit costs may be changed to more appropriate values so as to present a more realistic picture of the game. Discussion may also range as far as the validity of such a game model. The director should take active interest in explaining the assumptions of the model to the players and welcome suggestions for developing a better one.

Sample Problem

This problem was worked out for the first week signifying the starting of the game. The game was played (a) without considering break downs; (b) with the consideration of three break downs occurring in a week. Part (a) of the problem corresponds to either of the procedures 1 and 2 and Part (b) pertains to a single player.

The data generated for the problem, and the results have been shown in the following pages. The priority decisions made are entered in the decision form shown below.

Player No. 1																				Week 1
Priority number	1		2		3		4		5		6		7		8		9		10	
Job number	0	9	1	2	0	1	0	4	1	3	1	4	1	0	0	5	0	2	0	3
Priority number	11		12		13		14		15		16		17		18		19		20	
Job number	1	5	0	8	1	1	0	7	0	6										

Fig. 4. Decision Form

Data for the Sample Problem.

JOBS ORDERED THIS WEEK 15

S E Q U E N C E S

7	1	2	8	6	4	0	0
3	8	5	1	0	0	0	0
2	6	3	0	0	0	0	0
4	5	1	7	6	3	0	0
3	5	7	2	0	0	0	0
1	5	4	0	0	0	0	0
4	1	7	0	0	0	0	0
3	7	8	0	0	0	0	0
3	6	1	7	8	2	0	0
5	2	6	3	1	0	0	0
2	8	7	0	0	0	0	0
5	3	2	6	7	8	0	0
6	3	1	4	2	0	0	0
4	5	3	6	7	0	0	0
1	2	4	0	0	0	0	0

PROCESSING TIME IN HOURS

8	4	7	4	8	5	0	0
8	5	6	5	0	0	0	0
3	6	4	0	0	0	0	0
8	6	5	6	8	8	0	0
4	6	3	7	0	0	0	0
5	6	8	0	0	0	0	0
8	4	4	0	0	0	0	0
5	5	4	0	0	0	0	0
6	3	4	5	7	8	0	0

Processing Times(Continued)

4	3	7	5	3	0	0	0
7	4	5	0	0	0	0	0
8	4	6	6	4	8	0	0
3	6	4	5	3	0	0	0
6	4	8	6	8	0	0	0
3	4	7	0	0	0	0	0

JOB NO	MATERIAL COST
1	100
2	100
3	75
4	75
5	50
6	50
7	100
8	100
9	100
10	50
11	100
12	50
13	50
14	50
15	100

3

TOTAL NO OF JOBS HANDLED THIS WEEK 15

Part (a).

ORIGINAL SCHEDULE MACHINES X HOURS

15	15	15	6	6	6	6	6	0	9	9	9	9	1	1	1	1	4	4	4
3	3	3	15	15	15	15	0	0	0	0	0	12	12	12	12	12	12	1	1
9	9	9	9	9	9	0	0	12	12	12	12	13	13	13	13	13	13	14	14
4	4	4	4	4	4	4	4	14	14	14	14	14	14	15	15	15	15	15	15
12	12	12	12	12	12	12	12	4	4	4	4	4	4	14	14	14	14	10	10
13	13	13	0	0	0	9	9	9	3	3	3	3	3	3	0	0	0	12	12
1	1	1	1	1	1	1	1	0	0	0	0	0	9	9	9	9	9	0	0
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	9	9

4	4	13	13	13	13	0	0	0	0	0	0	—	—	—	—	—	—	—	7
1	1	1	1	1	9	9	9	9	9	9	9	9	13	13	13	10	10	10	0
14	14	14	14	14	14	5	5	5	5	2	2	2	2	2	2	2	2	3	3
15	0	0	0	0	0	13	13	13	13	13	7	7	7	7	7	7	7	7	6
10	10	6	6	6	6	6	6	0	0	5	5	5	5	5	5	0	0	0	0
12	12	12	12	0	0	14	14	14	14	14	14	0	0	0	0	0	0	0	0
—	—	—	—	12	12	12	12	4	4	4	4	4	4	14	14	14	14	14	14
9	9	9	9	9	0	0	0	12	12	12	12	12	12	12	12	1	1	1	1

7	7	7	0	0	0	0	0	0	0	0	2	2	2	2	2	0	0	0	0
—	—	—	—	—	5	5	5	5	5	5	5	11	11	11	11	11	11	11	0
3	3	8	8	8	8	8	0	0	0	0	0	0	0	0	0	4	4	4	4
6	6	6	6	6	6	6	0	1	1	1	1	1	0	0	0	0	0	0	0
—	—	—	—	—	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	4	4	4	4	4	4	4	4	10	10	10	10
14	14	5	5	5	0	0	8	8	8	8	8	7	7	7	7	0	0	0	0
2	2	2	2	2	0	0	0	0	0	0	0	8	8	8	8	0	0	0	11

Original Schedule (continued)

——————————————— 10 10 10 0 0 0 0 0 0 0 0

4 4 4 4 10 10 10 10 10 0 0 0 0 0 0 ——————

10 10 10 0 0 0 0 0 ——————————

—————— 11 11 11 11 11 0 0 0 0 0 0 *——————

11 11 11 0 0 0 0 0 ——————————

A REPORT OF THE RESULTS OF THE GAME

IDLE TIME	JOBS DELAYED
283	0

FIRST SHIFT HRS	SECOND SHIFT HRS
198	159

IDLE TIME DUE TO BREAKDOWNS 0

THE FOLLOWING JOBS ARE NOT DELIVERED

JOB NO	MAT COST	PROCESSING HRS	PRICE
1	100	36	702
2	100	24	570
3	75	13	398
4	75	41	706
5	50	20	426
6	50	19	414
7	100	16	482
8	100	14	460
9	100	33	668
10	50	22	448

Job No., etc. (continued)

11	100	16	482
12	50	36	602
13	50	21	436
14	50	32	558
15	100	14	460

INPROCESS INVENTORY COST	UNUTILIZED LABOR COST
16	1010

TOTAL REVENUE	TOTAL COST	NET PROFIT
7812	4920	2892

CUMULATIVE PROFIT 2892

Part (b).

ORIGINAL SCHEDULE MACHINES X HOURS

Same as Part (a)

FINAL SCHEDULE AFTER BREAKDOWNS MACH X HR

15	15	15	0	0	6	6	6	6	6	9	9	9	9	0	0	0	1	1	1
3	0	0	0	3	3	15	15	15	15	0	0	12	12	12	12	12	12	0	0
9	9	9	9	9	9	0	0	12	12	12	12	13	13	13	13	13	13	14	14
4	4	4	4	4	4	4	4	14	14	14	14	14	14	15	15	15	15	15	15
12	12	12	12	12	12	12	12	4	4	4	4	4	4	14	14	14	14	10	10
13	13	13	0	0	0	9	9	9	3	3	3	3	3	3	0	0	0	12	12
1	1	1	1	1	1	1	1	0	0	0	0	0	0	9	9	9	9	9	0
																			9

Final Schedule (continued)

1	4	4	4	4	4	13	13	13	13	0	0	0	0	0	0	0	—	—	—
—	1	1	1	1	1	1	1	9	9	9	9	9	9	9	9	13	13	13	10
14	14	14	14	14	14	5	5	5	5	2	2	2	2	2	2	2	2	3	3
15	0	0	0	0	0	0	0	*—	—	13	13	13	13	13	7	7	7	7	7
10	10	6	6	6	6	6	6	0	0	5	5	5	5	5	5	0	0	0	0
12	12	12	12	0	0	14	14	14	14	14	14	0	0	0	0	0	0	0	0
—	—	—	—	12	12	12	12	4	4	4	4	4	4	14	14	14	14	14	14
9	9	9	9	9	9	0	0	12	12	12	12	12	12	12	12	1	1	1	1

—	—	—	7	7	7	7	0	0	0	0	2	2	2	2	2	0	0	0	0
10	10	0	0	0	5	5	5	5	5	5	5	11	11	11	11	11	11	11	0
3	3	8	8	8	8	8	0	0	0	0	0	0	0	0	—	4	4	4	4
7	7	7	6	6	6	6	6	6	6	6	1	1	1	1	1	0	0	0	0
—	—	—	—	—	2	2	2	2	2	2	0	0	0	0	0	0	0	*—	0
1	1	1	1	1	1	1	1	4	4	4	4	4	4	4	4	10	10	10	10
14	14	5	5	5	0	0	8	8	8	8	8	7	7	7	7	0	0	0	0
2	2	2	2	2	0	0	0	0	0	0	0	8	8	8	8	0	0	0	11

—	—	—	—	—	—	—	—	—	10	10	10	0	0	0	0	0	0	0	0
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
4	4	4	4	10	10	10	10	10	0	0	0	0	0	—	—	—	—	—	—
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
10	10	10	0	0	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—
—	—	—	11	11	11	11	11	0	0	0	0	0	—	—	—	—	—	—	—
11	11	11	0	0	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—

A REPORT OF THE RESULTS OF THE GAME

IDLE TIME	JOBS DELAYED
283	0

FIRST SHIFT HRS	SECOND SHIFT HRS
193	164

IDLE TIME DUE TO BREAKDOWNS 0

THE FOLLOWING JOBS ARE NOT DELIVERED

JOB NO	MAT COST	PROCESSING HRS	PRICE
1	100	36	702
2	100	24	570
3	75	13	398
4	75	41	706
5	50	20	426
6	50	19	414
7	100	16	482
8	100	14	460
9	100	33	668
10	50	22	448
11	100	16	482
12	50	36	602
13	50	21	436
14	50	32	558
15	100	14	460

INPROCESS INVENTORY COST UNUTILIZED LABOR COST

Instruction for Participants

Introduction. The game is designed to enable the player or a group of players to gain experience in Job-shop-scheduling practice. In this exercise an attempt has been made to develop the intuition of players in realizing the relative importance of different jobs. The players are advised to build a comprehensive outlook of the job-shop situation with emphasis on the different costs incurred. The decision-making procedure followed by a player should be based on a set of logical considerations which may undergo changes, as he gains experience in playing this game.

This game deals with a job-shop of eight different machines to be scheduled for a five-day week, each day consisting of two eight-hour shifts. New orders are received at the beginning of the week. Total number of orders handled in a week's period is the sum total of jobs carried over from the previous week and the incoming new orders. All jobs are brought into the shop just before the starting of the first operation and they are taken out of the shop right after the processing is complete. As such, there is no fixed delivery time for any job. Any job which cannot be completed in a week's time is delayed and carried over to the next week. In between operations a job may be stored waiting for the next operation to be performed on it. Storage costs are 10 cents per job per hour. There is no limit to the in-process inventory. Processing costs $2.00 per hour per job. Cost of labor is $3.00 per hour in the first shift and $4.00 in the second shift. The price of any job is fixed by a pricing policy. The policy of the shop is to charge a price two times the estimated cost of a job. Any job, delayed for the first time, is sold without any profit. Any subsequent delay will reduce the price of the job by one fourth of the price fixed in the previous week.

Informations to Make Decision. In the beginning of each decision interval the director of the game will provide a list of new orders received with their sequences, processing times and cost of materials. The player will also receive the results of his previous decision.

A player has to assign numbers to the delayed jobs picked up from the schedule of the previous week. The delayed job appearing first in the list is assigned the next higher number after the new orders. The rest of the delayed jobs are numbered according to their order of appearance in the list. Next step is to find out the incomplete sequences and processing times for each delayed job. This is easily accomplished by following the Printed Schedule* and comparing each operation with the sequence and processing time requirement.

Decision-making. A player will go to make decision as regards the relative importance of jobs. The decisions made are entered in the decision form shown in Fig. 4. Any job considered to be the most important, will occupy the first position in the decision form. Two spaces are provided in the decision form for each entry.

Results and Reports

Results and reports of the game are submitted to each team in three parts as described below.

---

* In the case of a single player this operation is done over the final schedule.

(1) Original Schedule. The schedule of the shop without considering break downs is termed as the "original schedule." "Original Schedule" is printed in the result sheet in four parts, where each part represents the schedule of the shop for twenty hours. The rows and columns of the matrix represent machines and hours respectively, with the job numbers appearing as the matrix elements. For example, if job 10 appears under column 4 and row 1, it shows that the tenth job is scheduled on the first machine in the fourth hour.

The informations regarding total idle time and the number of delayed jobs under the present schedule is printed underneath the original schedule.

(2) Final Schedule.* The next item on the results sheet represents the schedule under which the job-shop has gone through as a result of a few break-downs occurring at random.

(3) A Report of the Results of the Game. The third part provides the following informations as the result of the game.

(a) Processing times in different shifts.

(b) Idle hours created as a result of break downs.

(c) Price of all jobs.

(d) In-process inventory cost.

(e) Unutilized labor cost.

(f) A statement of income showing revenue, total cost, net profit and cumulative profit.

Item 'b' stating the idle time due to break down, may not help the player in evaluating his decision; still then it will give him an idea as regards the effect of break downs in job-shop. This may be considered as a valuable information for a job-shop trainee.

---

* This is not reported in case of procedure 1.

AN EVALUATION OF THE DESIGN OF THE GAME

Any gaming exercise is valid, if it serves the purpose for which it is designed. The evaluation of a game from this angle, can only be made after the game has been played many times. Still then, an attempt can be made to evaluate a design from a number of basic considerations. Kibbee (6) has mentioned four basic constraints over which a game model is formed. They are; purpose, simplicity, verisimilitude and reality.

Purpose

As already mentioned, the present design aims to serve a single purpose: to teach the players the relative importance of different jobs in a job-shop. The present knowledge is inadequate to say that such a game will serve the purpose. This suggests to evaluate this exercise from the angle of the remaining constraints.

Simplicity

Simplicity is a vital consideration in the design of any gaming exercise. Simplicity can be judged in consideration of four main facets. They are: simplicity of participation, simplicity of computation, simplicity of administration and simplicity of construction.

The fourth element deals with the length of time spent on programming for the computer. This does not seem to be a consideration under academic activity, so discussion is based on the first three only.

The simplicity of participation has been kept fairly simple, the task of a player being limited to arrange the jobs in a particular order. A few minutes of decision-making time may be considered to be good enough for such a task. Simplicity judged from the computational viewpoint, is debatable. The computational steps covered for adjusting the schedule after random break downs, are numerous. The larger the number of break downs, the longer is the computing time. The digital algorithm formulated, needs extensive storage locations, which suggests for an expensive computer to play a similar game involving a larger job-shop scheduled for a longer interval of time.

Simplicity of administration, in other words, conducting the play is a consideration solely to assist the director. This exercise assumes a model, which will not need much explanation on the part of the director, making briefing simple. The handling of data cards and running the program on the computer has been made handy by a number of automatic computer instructions.

Realism and Verisimilitude

Realism and verisimilitude are two important considerations pertaining to a game model. Realism is considered important in case, the game goes to describe a specific situation. For example, the purpose of the game may be to show the price-demand behavior of a particular commodity. In contrast with realism, verisimilitude means the appearance of reality to the players. A training-purpose game stresses mainly on the appearance of reality only.

The game dealt here, has ignored some considerations pertaining to a real job-shop situation like: the assumption regarding the delivery and routing of jobs, etc. These assumptions do not seem to interfere in realizing the relative importance of different jobs, they mearly present a situation different from reality to some extent. A little attempt of imparting reality to the exercise has been made in providing break downs. This may not affect decision-making very much, still it will go to provide an idea of the changes that can occur over a prepared schedule due to random discontinuations in processing. An increase in the amount of idle time, increase in the number of jobs delayed and increase in the cost of in-process inventory caused due to break downs may be a matter of interest for a trainee.

CONCLUDING REMARKS

The simulator presented in this thesis has been designed to work with over-all priorities assigned to different jobs. In practice, the importance of different jobs are subjected to changes, as they move towards completion. This suggests a priority order system changing with time. Rowe (8) cited some priority rules, which were small algebraic expressions, used to compute the priority index of a group of jobs waiting to be processed, and one having the lowest number was assigned to a machine. In playing a job-shop scheduling game manually, a player may calculate priority index to aid him in schedulihg effectively. The same is difficult to adopt in case of a computer game. However, a program can be written to utilize a large number of priority rules and the task of the player may be to select one of the rules. This will only be another version of a problem solving simulation, the spirit of gaming being completely lost in it. The characteristics of a gaming exercise needs the decisions to be made in numericals. The following procedure may help to develop a good computer game for job-shop scheduling.

The players of the game will be given the task of assigning relative priority to jobs on a machine basis. In other words, a player has to arrange the jobs in a particular order for a specific machine. This will result in a decision form of the form of a matrix, where the columns will represent machines and rows will represent priority positions. The matrix cells will occupy job numbers and in the absence of a job, a cell will bear a zero. This will occur as the matrix will specify the maximum dimensions of jobs and machines. A computer program can be written to schedule to job-shop with this new system. Such a program can be tied with Program II (b) and (c) to get a report of the results in a similar form.

ACKNOWLEDGMENT

The author wishes to express his sincere gratitude to Professor J. J. Smaltz, Department of Industrial Engineering for his assistance and encouragement in the formulation of this thesis. The author also wishes to express his gratitude to Dr. George F. Schrader, Head of the Department Industrial Engineering for providing valuable suggestions and corrections in developing this thesis.

REFERENCES

(1) Bowman, E. H.
The schedule sequencing problem. Operations Research, Vol. 7, 1959, 621 p.

(2) Decon, Amos R. L.
AMA Management Report No. 55: General Management Division, American Management Association, Inc., 5 p.

(3) Giffer, B.
Mathematical solution of explosion and scheduling problems. IBM Research Report RC-118. IBM Research Center Business Systems Research, Yorktown Heights, New York, June 1959.

(4) Giffer, B. and G. L. Thompson.
Algorithms for solving production scheduling problems. IBM Research Report RC-118. June 1959.

(5) Heller, J.
Combinatorial, probabilistic and statistical aspects of an M × J scheduling problem, Report NYO-2540. AEC Computing and Applied Mathematics Center, New York University, February 1959, 12 p.

(6) Kibbee, Joel M., Clifford J. Craft and Burt Nanus.
General Aspects of Game Design, Management Games, Chapter 7. New York: Reinhold Publishing Corporation, 1961.

(7) Manne, Alan S.
On the job-shop scheduling problem. Cowles Foundation Discussion Paper. No. 73, May 8, 1959.

(8) Rowe, Alan J.
Toward a theory of scheduling. Journal of Industrial Engineering, Vol. 11, No. 2, March-April, 1960. 125 p.

(9) Sisson, R. L.
Sequencing Theory. Publications in Operations Research, No. 5. Operations Research Society of America. Chapter 7. New York: John Wiley and Sons, Inc., p. 311-321.

(10) Sisson, R. L.
Methods of sequencing in job-shops - A Review. Operations Research, Vol. 7, 1959. 10 p.

(11) Smith, W. E.
Various optimizers for single stage production. Naval Research Logistics Quarterly. Vol. 3, 1956. 3 p.

(12) Wagner, H. M.
An integer linear programming model for machine scheduling. Naval Research Logistics Quarterly. Vol. 5, 1959. 131 p.

APPENDIX

Program I

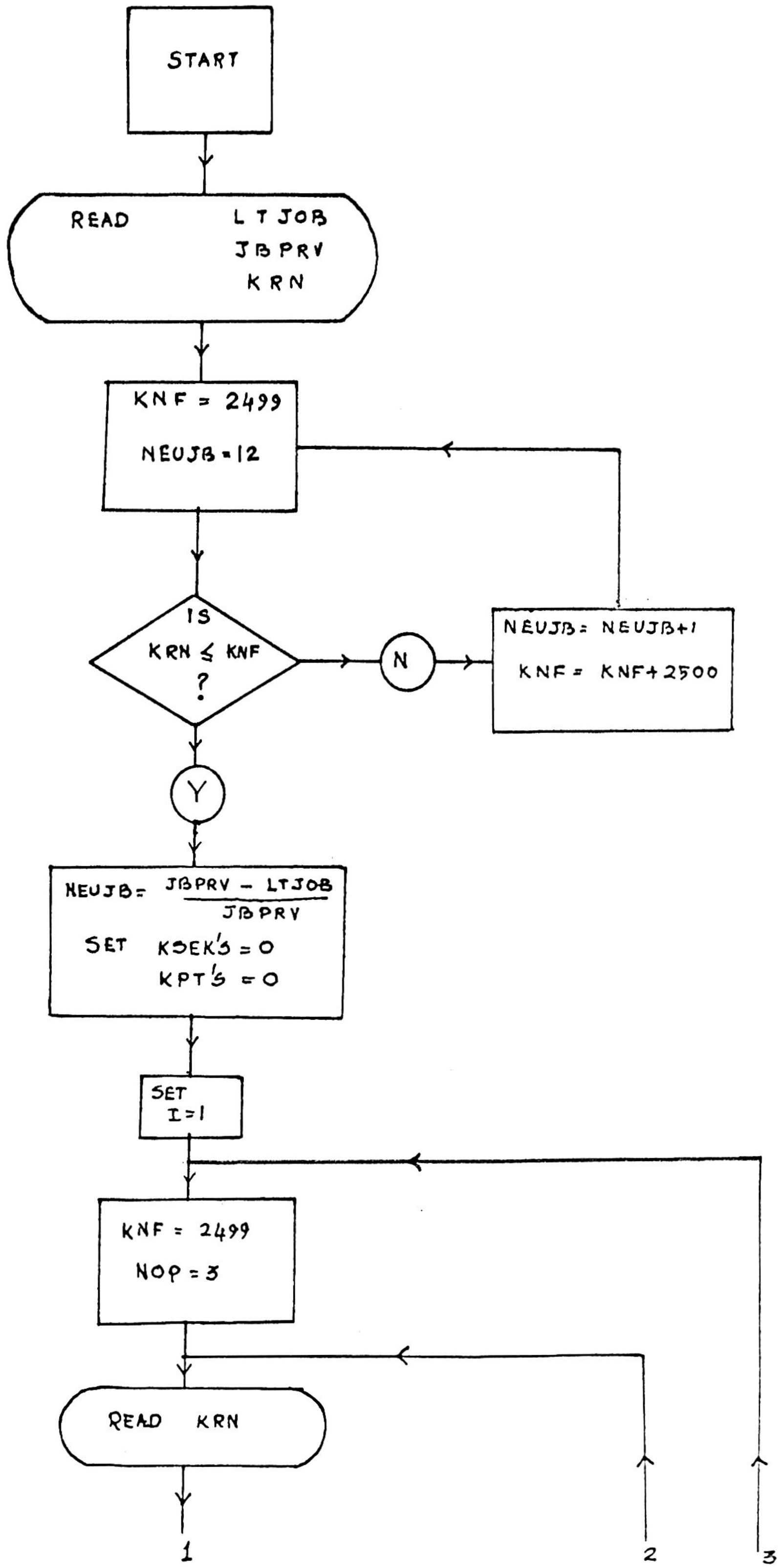
```
C     A PROGRAM TO GENERATE ORDERS FOR THE JOB-SHOP
      DIMENSIONKSEK(30,8),KPT(30,8)
      READ503,LTJOB
      READ503,JBPRV
      READ500,KRN
  503 FORMAT(I3)
  500 FORMAT(I4)
      KNF=2499
      NEUJB=12
  504 IF(KRN-KNF)501,501,502
  502 NEUJB=NEUJB+1
      KNF=KNF+2500
      GOTO504
  501 DD=NEUJB
      PRVDD=JBPRV
      DIFF=JBPRV-LTJOB
      CONST=DIFF/PRVDD
      NEUJB=DD*CONST
      PUNCH303,NEUJB
  303 FORMAT(22HJOBS ORDERED THIS WEEK4XI3)
      DO215I=1,NEUJB
      DO216J=1,8
      KSEK(I,J)=0
      KPT(I,J)=0
  216 CONTINUE
  215 CONTINUE
```

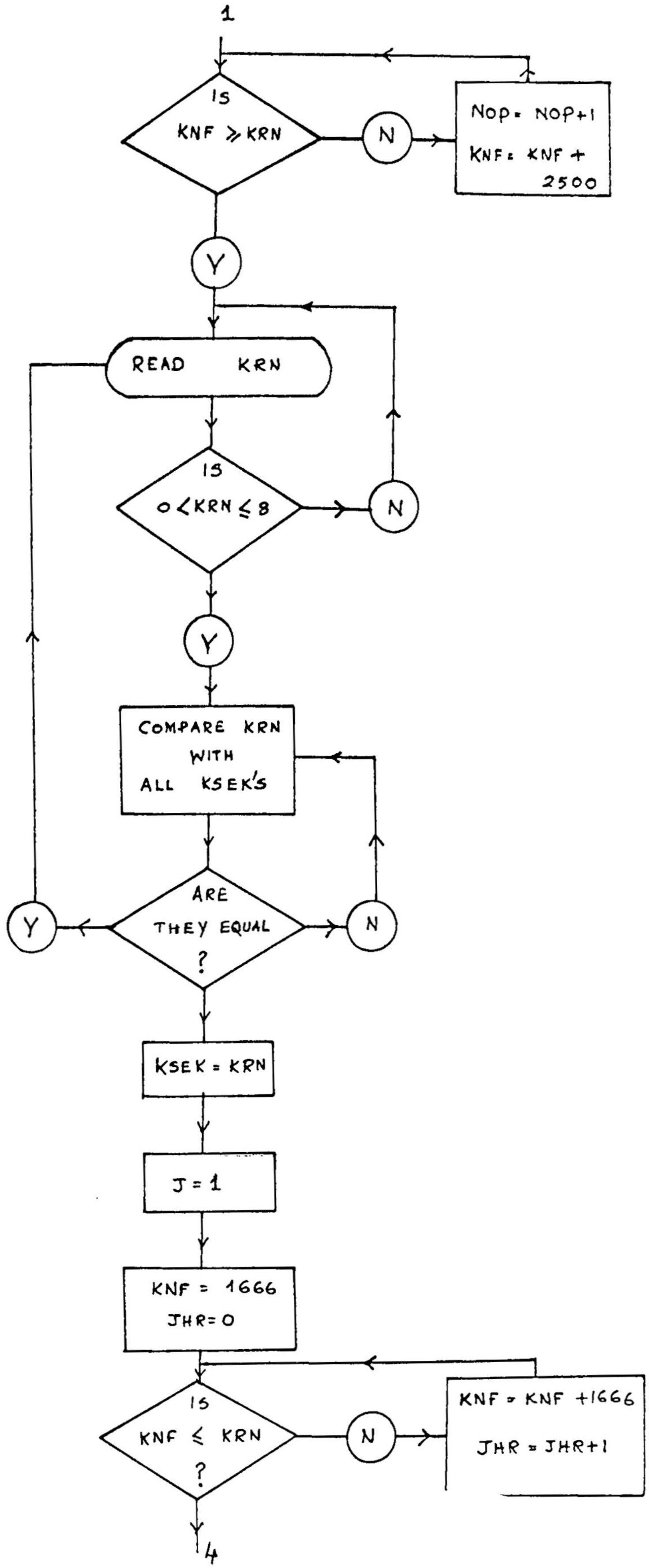
```
      DO201I=1,NEUJB
      KNF=2499
      NOP=3
      READ500,KRN
  205 IF(KNF-KRN)203,204,204
  203 KNF=KNF+2500
      NOP=NOP+1
      IF(NOP-6)205,204,204
  204 DO206J=1,NOP
  208 READ207,KRN
  207 FORMAT(I1)
      IF(KRN)208,208,219
  219 IF(KRN-8)217,217,208
  217 DO218K=1,J
      IF(KSEK(I,K)-KRN)218,208,218
  218 CONTINUE
  209 KSEK(I,J)=KRN
      KNF=1665
      JHR=3
      READ500,KRN
  214 IF(KRN-KNF)213,213,212
  212 KNF=KNF+1666
      JHR=JHR+1
      GOTO214
  213 KPT(I,J)=JHR
  206 CONTINUE
  201 CONTINUE
      PUNCH300
```

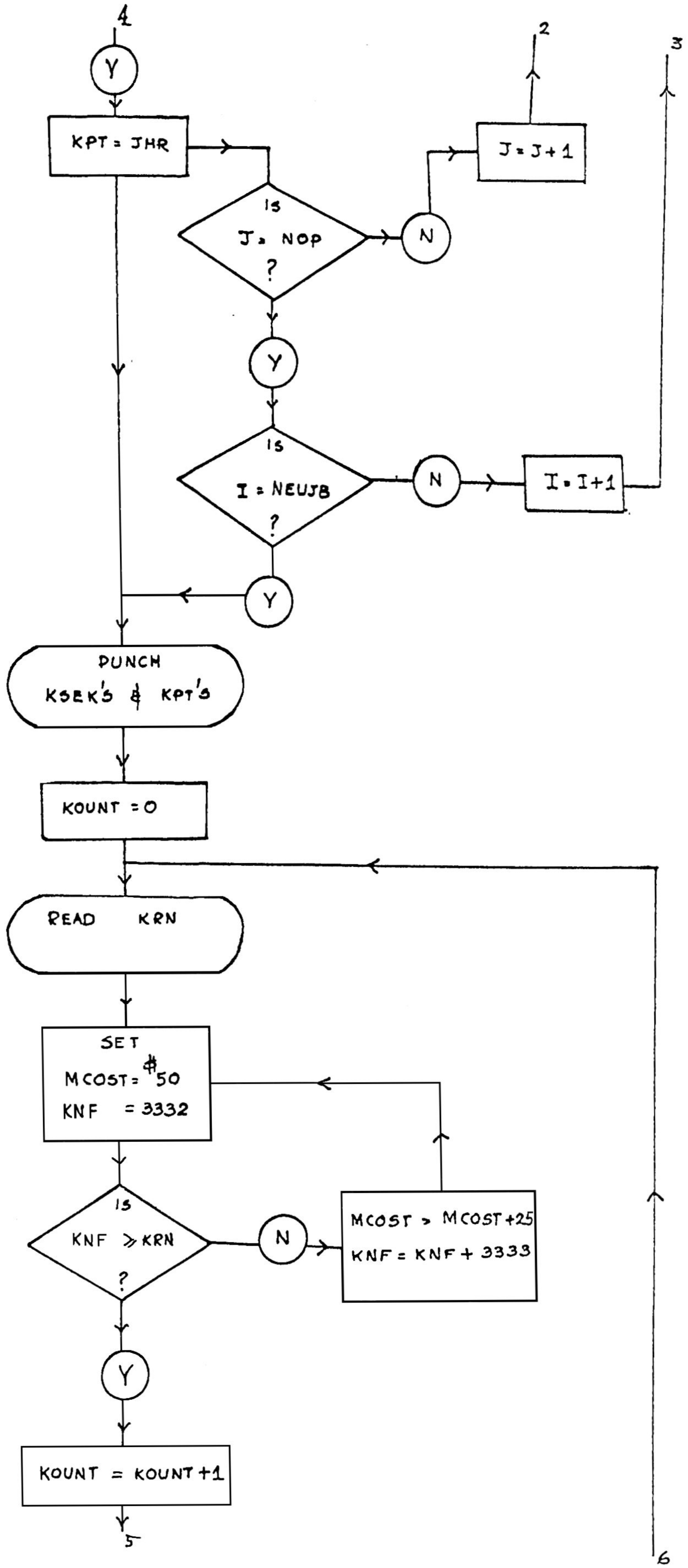
```
      PUNCH301
      PUNCH210,((KSEK(I,J),J=1,8),I=1,NEUJB)
  300 FORMAT(25HS   E   Q   U   E   N   C   E   S)
  301 FORMAT(24HPROCESSING TIME IN HOURS)
      PUNCH210,((KPT(I,J),J=1,8),I=1,NEUJB)
  210 FORMAT(8I3)
      PUNCH304
  304 FORMAT(23HJOB NO     MATERIAL COST)
      KOUNT=0
    3 READ12,KRN
      KNF=3332
      MCOST=50
   32 IF(KRN-KNF)100,100,29
   29 MCOST=MCOST+25
      KNF=KNF+3333
      GOTO32
  100 KOUNT=KOUNT+1
      PUNCH305,KOUNT,MCOST
  305 FORMAT(2XI2,11H           I3)
      IF(KOUNT-NEUJB)3,5,5
    5 READ12,KRN
      RN=KRN
      RN=RN/10000.
      BD=0.
      FBS=0.
      BDF=1.
      X=1.
```

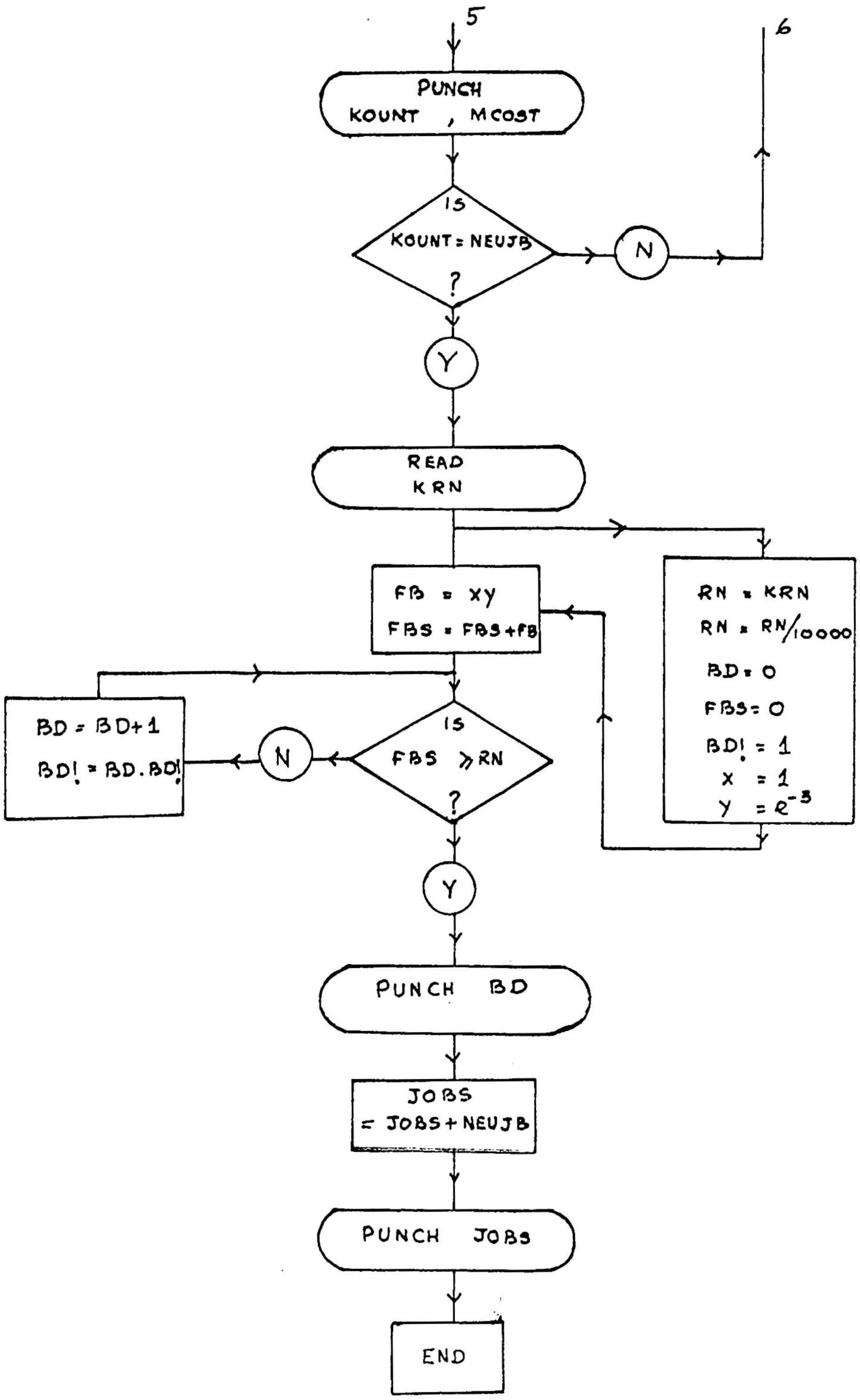
```
      Y=1./EXPF(3.)
      GOTO33
   16 X=((3.**BD)/BDF)
   33 FB=X*Y
      FBS=FBS+FB
      IF(RN-FBS)30,30,23
   23 BD=BD+1.
      BDF=BDF*BD
      GOTO16
   30 KBD=BD
      PUNCH27,KBD
   12 FORMAT(I4)
   27 FORMAT(I3)
      JOBS=NEUJB+LTJOB
      PUNCH302,JOBS
  302 FORMAT(34HTOTAL NO OF JOBS HANDLED THIS WEEKI3)
      END
```

Flow Chart for Program I




1
IS
KNF ≥ KRN
N
NOP = NOP+1
KNF = KNF + 2500
Y
READ KRN
IS
0 < KRN ≤ 8
N
Y
COMPARE KRN WITH ALL KSEK'S
ARE THEY EQUAL ?
Y
N
KSEK = KRN
J = 1
KNF = 1666
JHR = 0
IS
KNF ≤ KRN
?
N
KNF = KNF +1666
JHR = JHR+1
4


4
Y
KPT = JHR
IS
J = NOP
?
N
J = J + 1
2
Y
IS
I = NEUJB
?
N
I = I+1
3
Y
PUNCH
KSEK'S & KPT'S
KOUNT = 0
READ KRN
SET
MCOST = $50
KNF = 3332
IS
KNF ≥ KRN
?
N
MCOST = MCOST +25
KNF = KNF + 3333
Y
KOUNT = KOUNT +1
5
6


5
6
PUNCH
KOUNT , MCOST
IS
KOUNT = NEUJB
?
N
Y
READ
KRN
RN = KRN
RN = RN/10000
BD = 0
FBS = 0
BD! = 1
X = 1
Y = e^-3
FB = XY
FBS = FBS+FB
IS
FBS ≫ RN
?
N
BD = BD+1
BD! = BD.BD!
Y
PUNCH BD
JOBS
= JOBS + NEUJB
PUNCH JOBS
END

Program II

Part (a)

```
C     A PROGRAM TO SCHEDULE THE JOB-SHOP FOR  FIVE-DAY WEEK
      DIMENSIONMY(8,80),MX(8,80),MZ(8,80),LBD(8,80),LBF(80),KSEK(30,8),
     1KSS(8),NPR(30),NPT(30,8),JDEL(20),KSTRT(20),KEND(20),KKS(20)
  312 FORMAT(I4)
  301 FORMAT(I3)
  300 FORMAT(8I3)
      READ301,JOBS
      READ301,NEUJB
      READ300,((KSEK(I,J),J=1,8),I=1,JOBS)
      READ300,((NPT(I,J),J=1,8),I=1,JOBS)
  891 L=0
      CINV=0.
      READ950,(NPR(I),I=1,JOBS)
  950 FORMAT(20I2)
      DO450I=1,20
  450 JDEL(I)=0
      DO302I=1,8
      DO302J=1,80
  302 MX(I,J)=0
      DO900I=1,20
      KKS(I)=0
      KSTRT(I)=0
  900 KEND(I)=0
      M=1
  318 IF(JOBS-M)321,315,315
  315 K=NPR(M)
      IR=0
```

```
      IR=0
      DO308KM=1,8
      IF(KSEK(K,KM))308,307,310
  308 CONTINUE
      GOTO307
  307 KEND(K)=KSR
      M=M+1
      GOTO318
  310 I=KSEK(K,KM)
      KPT=NPT(K,KM)
      IR=IR+1
      KOUNT=0
      DO313J=IR,80
      IF(MX(I,J))313,311,316
  316 KOUNT=0
  313 CONTINUE
      L=L+1
      JDEL(L)=NPR(M)
      GOTO307
  311 KOUNT=KOUNT+1
      IF(KOUNT-KPT)313,306,306
  306 KSR=J
      KR=J-KPT+1
      IF(KM-1)320,901,902
  901 KSTRT(K)=KR
  902 KKS(K)=KKS(K)+KPT
      DO304J=KR,KSR
  304 MX(I,J)=NPR(M)
```

```
      GOTO308
321   PUNCH252
252   FORMAT(26X34HORIGINAL SCHEDULE MACHINES X HOURS)
      PUNCH600,((MX(I,J),J=1,20),I=1,8)
      PUNCH705
      PUNCH600,((MX(I,J),J=21,40),I=1,8)
      PUNCH705
      PUNCH600,((MX(I,J),J=41,60),I=1,8)
      PUNCH705
      PUNCH600,((MX(I,J),J=61,80),I=1,8)
600   FORMAT(10X20I3)
705   FORMAT(//)
      PUNCH254
254   FORMAT(23X35HA REPORT OF THE RESULTS OF THE GAME)
      IDTMW=0
      DO701J=1,80
      DO702I=1,8
      IF(MX(I,J))320,703,702
703   IDTMW=IDTMW+1
702   CONTINUE
701   CONTINUE
      PUNCH704
704   FORMAT(26X25HIDLE TIME     JOBS DELAYED)
      PUNCH709,IDTMW,L
709   FORMAT(28XI3,12XI3)
```

Part (b)

```
C      A PROGRAM TO PROVIDE BREAK DOWNS OVER THE ORIGINAL SCHEDULE
       READ301,KBD
       IF(KBD)320,260,261
  260  DO262I=1,8
       DO262J=1,80
  262  MZ(I,J)=MX(I,J)
       GOTO281
  261  DO111J=1,80
       DO111I=1,8
  111  MY(I,J)=0
       K=1
  335  READ334,KRN
  334  FORMAT(I2)
       IF(KRN)320,335,340
  340  IF(KRN-80)336,336,335
  336  J=KRN
  342  READ337,KRN
  337  FORMAT(I1)
       IF(KRN)342,342,341
  341  IF(KRN-8)339,339,342
  339  I=KRN
       IF(MX(I,J))320,335,343
  343  READ312,KRN
       RN=KRN
       RN=RN/10000.
       BDH=0.
       BDS=0.
```

```
      BDHF=1.
      X=1.
      Y=1./EXPF(4.)
      GOTO350
  353 X=((4.**BDH)/BDHF)
  350 FHB=X*Y
      BDS=BDS+FHB
      IF(RN-BDS)351,351,352
  352 BDH=BDH+1.
      BDHF=BDHF*BDH
      GOTO353
  351 IF(BDH)320,343,263
  263 KBDH=BDH
      KKK=J+KBDH-1
      DO345KK=J,KKK
  345 MY(I,KK)=-50
      IF(K-1)320,370,371
  370 LF=J
  371 K=K+1
      IF(K-KBD)335,335,344

C     A PROGRAM FOR DETERMINING THE ACTUAL JOB-SHOP SCHEDULE AFTER
C     BREAK DOWNS HAVE OCCURRED
  344 DO201I=1,8
  201 KSS(I)=0
      IF(LF-1)37,9,10
   10 LK=LF-1
```

```
      DO5J=1,LK
      DO5I=1,8
    5 MZ(I,J)=MX(I,J)
    9 DO7J=LF,80
      DO7I=1,8
    7 MZ(I,J)=0
      DO70J=1,80
      DO70I=1,8
   70 LBD(I,J)=0
      DO72J=1,80
   72 LBF(J)=0
   11 J=LF
   12 DO23I=1,8
      IF(MY(I,J))21,24,320
   24 DO30JK=J,80
   30 MZ(I,JK)=MX(I,JK)
      GOTO23
   21 MZ(I,J)=0
      LBD(I,J)=0
      IF(MX(I,J))320,23,26
   26 JJ=J
   28 K=JJ+1
      IF(MX(I,JJ))320,56,960
   56 DO57JJP=K,80
   57 MZ(I,JJP)=MX(I,JJP)
      GOTO23
  960 IF(LF-80)27,97,320
   27 MZ(I,K)=MX(I,JJ)
```

```
    LBD(I,K)=MZ(I,K)
102 JJ=JJ+1
    IF(JJ-80)28,95,23
 95 IF(MX(I,JJ))320,23,97
 97 DO620MJ=1,L
    IF(JDEL(MJ)-MX(I,JJ))620,909,620
620 CONTINUE
    MM=MX(I,JJ)
909 KKS(MM)=KKS(MM)-1
    L=L+1
    JDEL(L)=MX(I,JJ)
 23 CONTINUE
962 IF(LF-80)961,37,320
961 JKK=J+1
 91 DO41IKK=1,8
    IF(LBD(IKK,JKK))320,41,42
 41 CONTINUE
    GOTO96
 42 LBP=LBD(IKK,JKK)
    N=0
    DO43II=1,8
    IF(MZ(II,JKK))320,43,80
 80 IF(LBP-MZ(II,JKK))43,44,43
 44 N=N+1
    KSS(N)=II
 43 CONTINUE
    IF(N-2)41,120,120
120 DO400JS=1,8
```

```
      DO121NN=1,8
      IF(KSEK(LBP,JS)-KSS(NN))121,124,121
  121 CONTINUE
  400 CONTINUE
  124 KK=KSS(NN)
      DO125NR=1,8
      IF(KSS(NR))320,125,126
  126 IF(KSS(NR)-KK)150,125,150
  125 CONTINUE
      GOTO41
  150 IF(JKK-80)45,309,37
  309 II=KSS(NR)
      LBD(II,JKK)=0
      MZ(II,JKK)=0
      GOTO125
   45 II=KSS(NR)
      DO46JNN=JKK,80
   46 LBF(JNN)=MZ(II,JNN)
      DO47JX=JKK,80
      JNK=JX+1
      IF(LBF(JX))320,49,48
   48 MZ(II,JNK)=LBF(JX)
      LBD(II,JNK)=MZ(II,JNK)
  114 IF(JX-80)47,99,49
   99 IF(LBF(JX))320,47,280
  280 DO621MJ=1,L
      IF(JDEL(MJ)-LBF(JX))621,920,621
  621 CONTINUE
```

```
      L=L+1
      JDEL(L)=LBF(JX)
      MM=LBF(JX)
  920 KKS(MM)=KKS(MM)-1
   47 CONTINUE
   49 MZ(II,JKK)=0
      LBD(II,JKK)=0
      DO51JXX=1,80
   51 LBF(JXX)=0
      GOTO125
   96 J=J+1
      IF(J-80)52,50,37
   50 DO60I=1,8
      IF(MY(I,J))61,60,320
   61 MZ(I,J)=0
   60 CONTINUE
      GOTO37
   52 DO53JR=J,80
      DO53IX=1,8
   53 MX(IX,JR)=MZ(IX,JR)
      GOTO12
   37 PUNCH253
  253 FORMAT(21X41HFINAL SCHEDULE AFTER BREAKDOWNS MACH X HR)
      PUNCH1,((MZ(I,J),J=1,20),I=1,8)
      PUNCH705
      PUNCH1,((MZ(I,J),J=21,40),I=1,8)
      PUNCH705
      PUNCH1,((MZ(I,J),J=41,60),I=1,8)
```

```
      PUNCH705
      PUNCH1,((MZ(I,J),J=61,80),I=1,8)
    1 FORMAT(10X20I3)
```

Part (c)

```
C     A PROGRAM TO REPORT THE RESULTS OFTHE GAME
  281 KSFT1=0
      KSFT2=0
      K=8
      KK=1
      KOUNT=1
  274 DO270J=KK,K
      DO250I=1,8
      IF(MZ(I,J))320,250,290
  290 GOTO(271,272,271,272,271,272,271,272,271,272),KOUNT
  271 KSFT1=KSFT1+1
      GOTO250
  272 KSFT2=KSFT2+1
  250 CONTINUE
  270 CONTINUE
  273 K=K+8
      KK=KK+8
      KOUNT=KOUNT+1
      IF(KOUNT-10)274,274,275
  275 PUNCH255
  255 FORMAT(23X35HFIRST SHIFT HRS     SECOND SHIFT HRS)
      PUNCH276,KSFT1,KSFT2
  276 FORMAT(29XI3,18XI3)
```

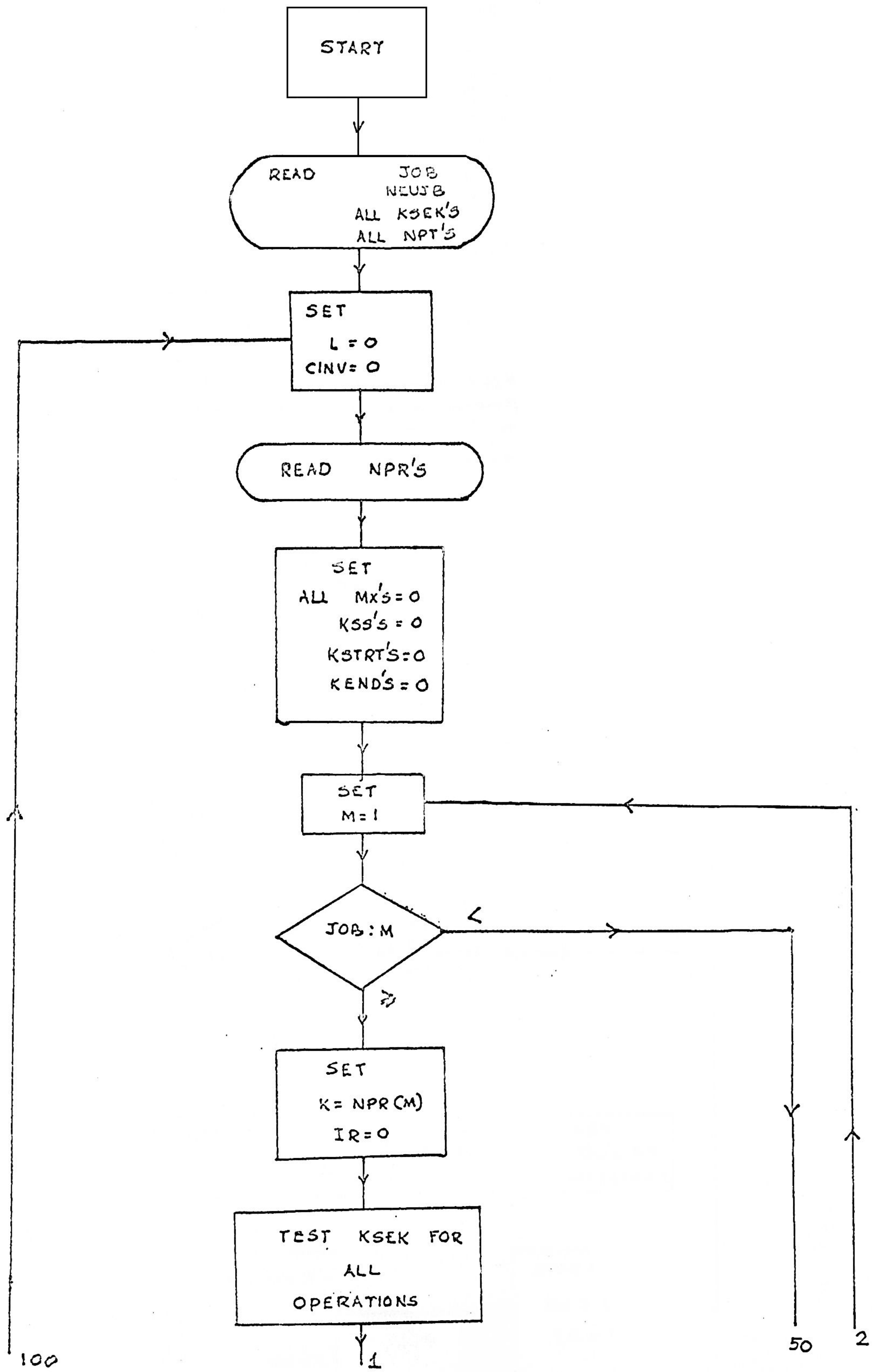
```
      IDTM=640-(KSFT1+KSFT2)
      IDTMB=IDTM-IDTMW
      PUNCH706,IDTMB
  706 FORMAT(22X30HIDLE TIME DUE TO BREAKDOWNS   I3)
  267 PUNCH256
  256 FORMAT(22X36HTHE FOLLOWING JOBS ARE NOT DELIVERED)
      PUNCH268,(JDEL(J),J=1,L)
  268 FORMAT(24X20I3)
      PAUSE
 2050 PRINT288
  288 FORMAT(40HLOAD MATERIAL COST,PRICE AND PROFIT DATA)
      PUNCH462
  462 FORMAT(18X45HJOB NO    MAT COST    PROCESSING HRS    PRICE)
      KTCST=0
      KSVNU=0
      DO460I=1,NEUJB
      KSUM=0
      READ800,KOUNT,MCOST
  800 FORMAT(2XI2,11H           I3)
      IF(KOUNT-I)999,1000,999
  999 PRINT1001
 1001 FORMAT(39HMATERIAL COST CARDS ARE OUT OF SEQUENCE)
 1000 KTCST=KTCST+MCOST
      DO461J=1,8
      KSUM=KSUM+NPT(I,J)
  461 CONTINUE
      KOJB=(KSUM*11)/2+MCOST+(800/NEUJB)
      KRVNU=KOJB*2
```

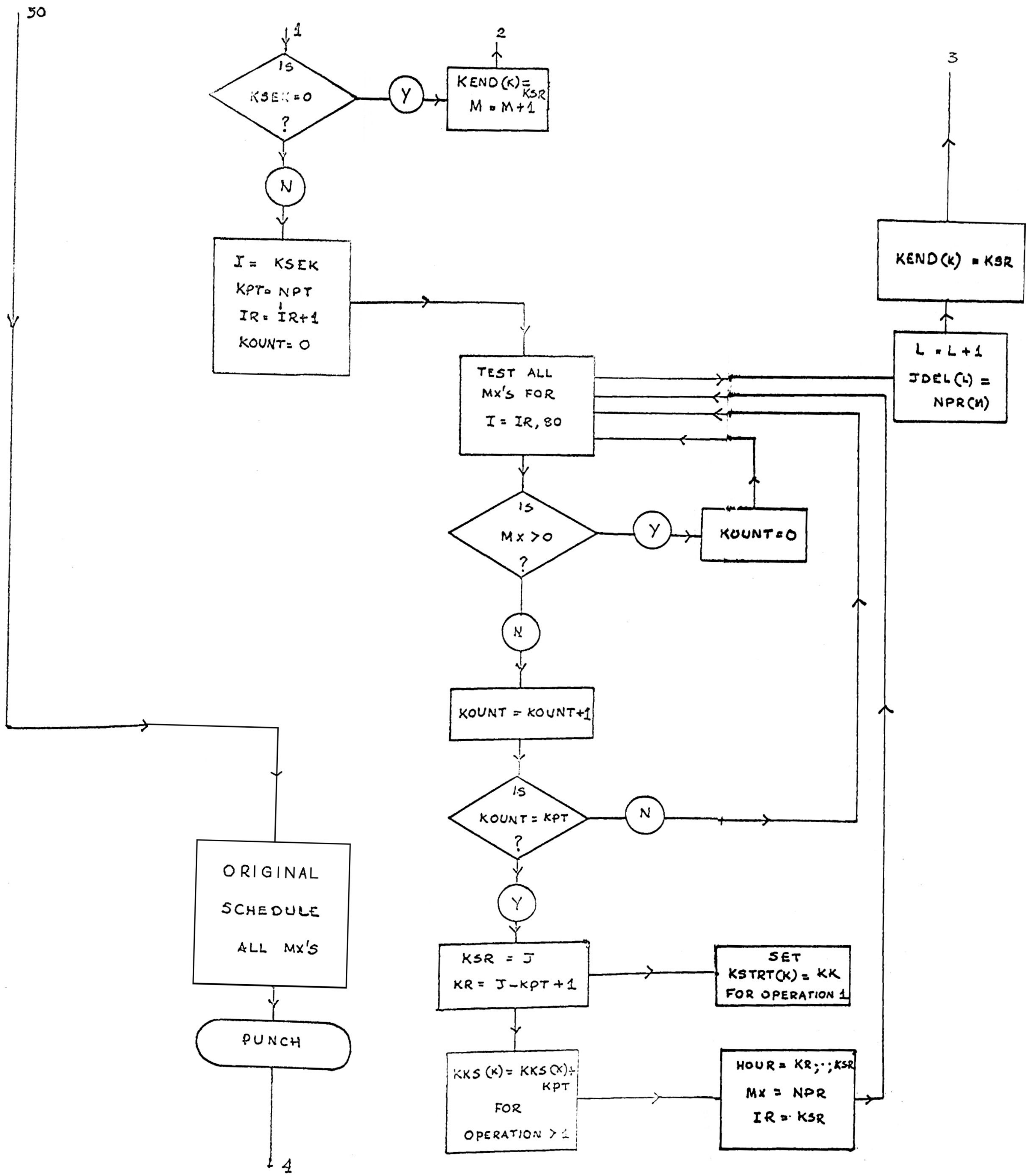
```
    DO464K=1,L
    IF(I-JDEL(K))903,904,903
465 KRVNU=KOJB
    GOTO469
464 CONTINUE
    KSVNU=KSVNU+KRVNU
469 PUNCH463,I,MCOST,KSUM,KRVNU
463 FORMAT(20XI3,7XI3,11XI3,12XI4)
460 CONTINUE
    IF(JOBS-NEUJB)320,605,610
610 PUNCH406
406 FORMAT(30X27HDELAYED JOBS FROM LAST WEEK)
    KK=NEUJB+1
    DO466I=KK,JOBS
    READ808,KRVNU
808 FORMAT(47X12H            I3)
    DO404K=1,L
    IF(I-JDEL(K))903,904,903
405 KRVNU=(KRVNU*3)/4
    GOTO409
404 CONTINUE
    KSVNU=KSVNU+KRVNU
409 PUNCH407,I,KRVNU
407 FORMAT(20XI3,36XI3)
466 CONTINUE
605 KCINV=CINV
    KCOST=800+(8*280)+2*(KSFT1+KSFT2)+KTCST+KCINV
    LCOST=(3*KSFT1)+4*KSFT2
```

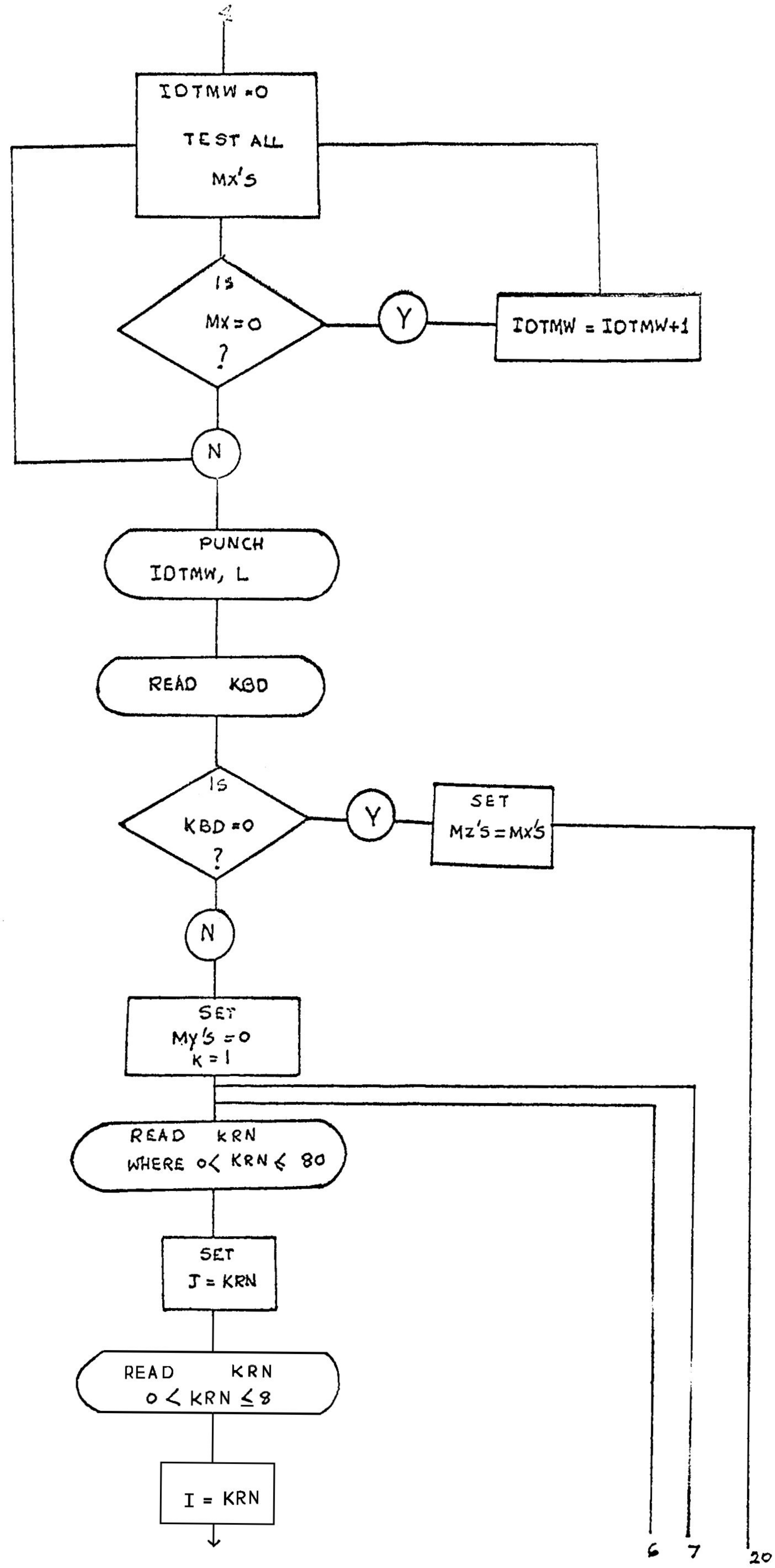
```
      LCOST=8*280-LCOST
      PUNCH906
  906 FORMAT(16X49HINPROCESS INVENTORY COST     UNUTILIZED LABOR COST)
      PUNCH910,KCINV,LCOST
  910 FORMAT(26XI4,20XI4)
      NPRFT=KSVNU-KCOST
      READ801,KPRFT
  801 FORMAT(42X20H                   I5)
      KPRFT=NPRFT+KPRFT
      PUNCH408
  408 FORMAT(19X43HTOTAL REVENUE      TOTAL COST      NET PROFIT)
      PUNCH410,KSVNU,KCOST,NPRFT
  410 FORMAT(21XI5,11XI5,12XI5)
      PUNCH810,KPRFT
  810 FORMAT(42X20HCUMULATIVE PROFIT   I5)
      PRINT890
  890 FORMAT(39HLOAD DATA FOR THE NEXT TEAM,PRESS START)
  320 PAUSE
      GOTO891
  903 DIFF=KEND(I)-KSTRT(I)-KKS(I)+1
      CINV=CINV+.10*DIFF
      IF(I-NEUJB)464,464,404
  904 DIFF=80-KSTRT(I)-KSS(I)+1
      CINV=CINV+.10*DIFF
      IF(I-NEUJB)465,465,405
      END
```

Flow Diagram for Program II

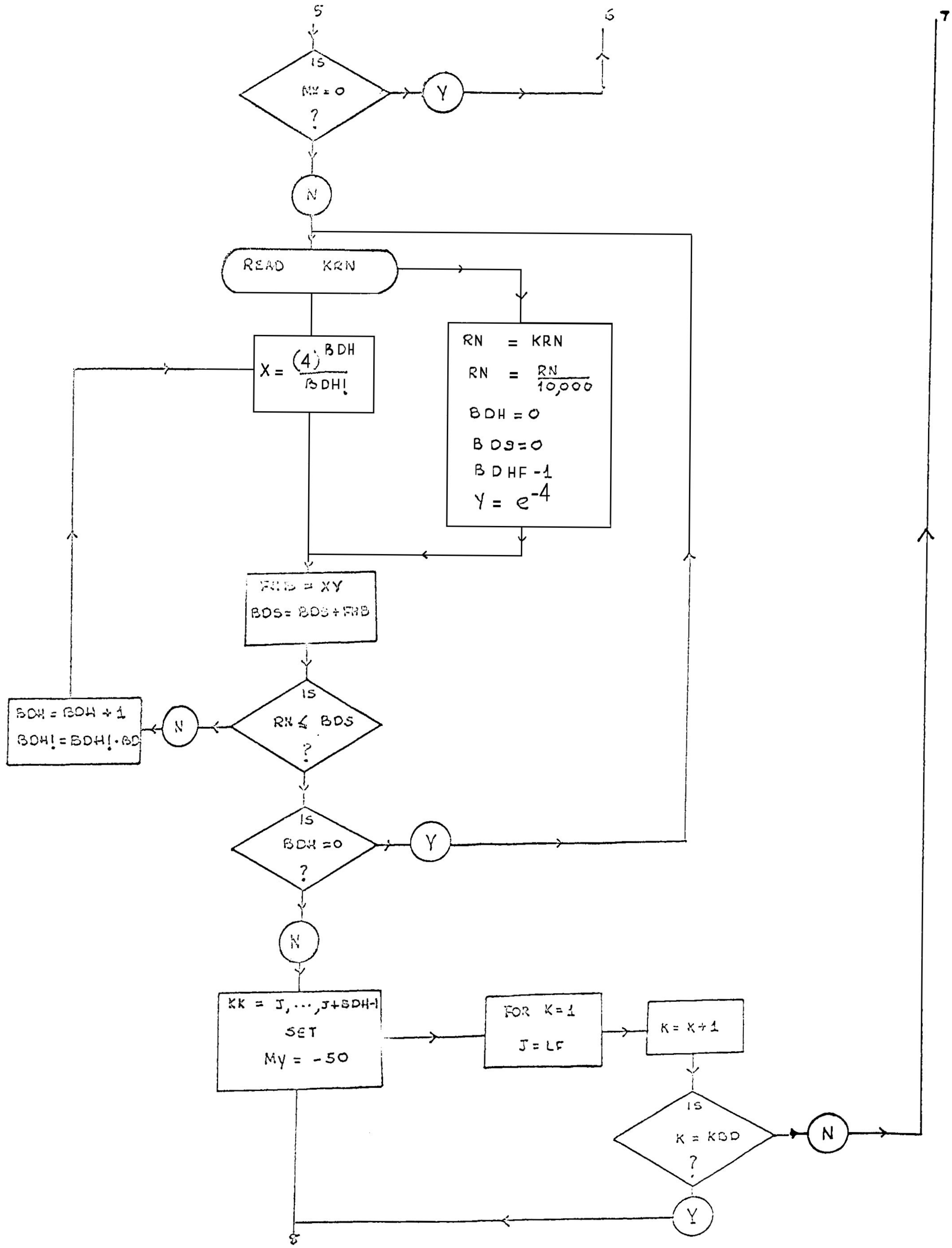
Part (a)

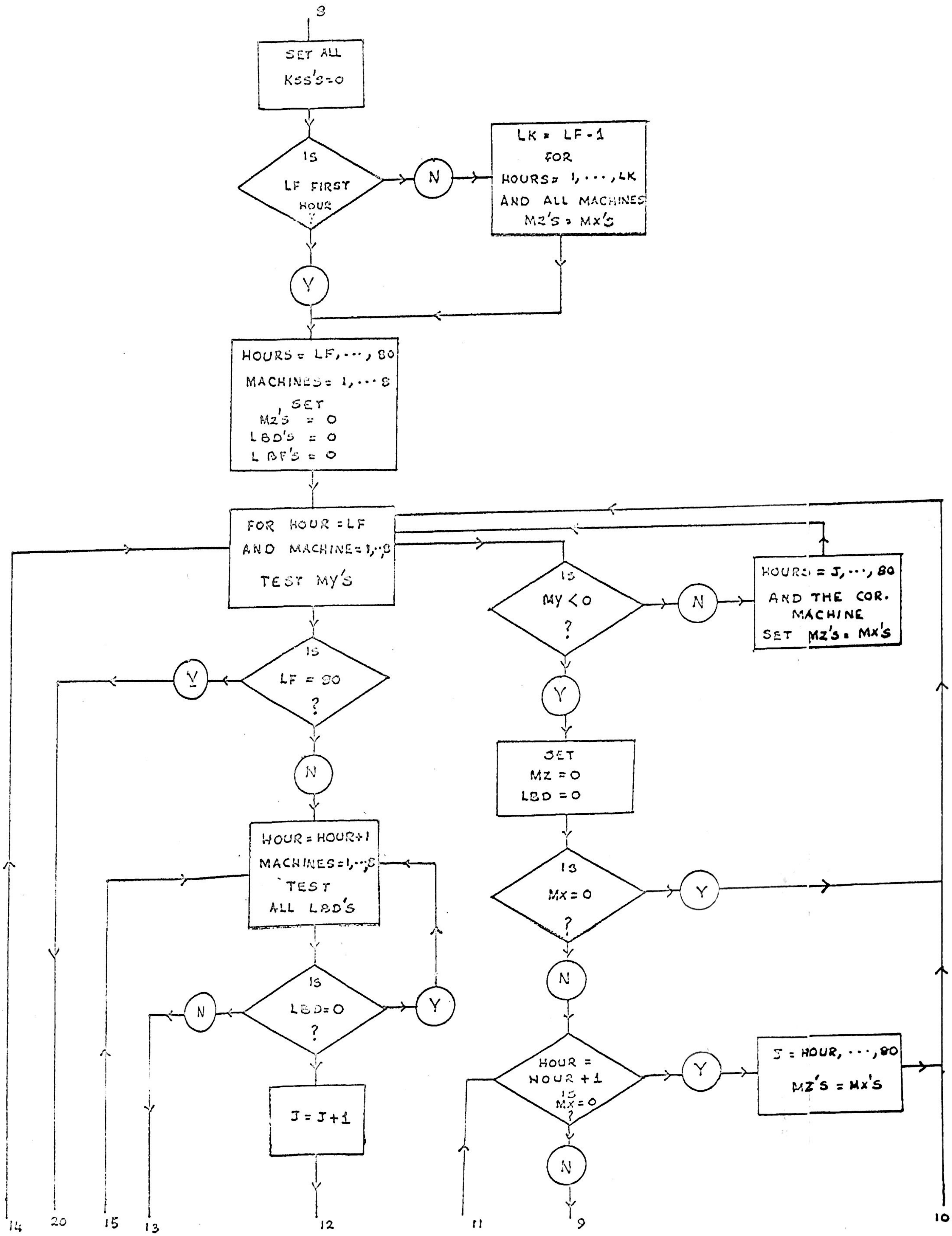


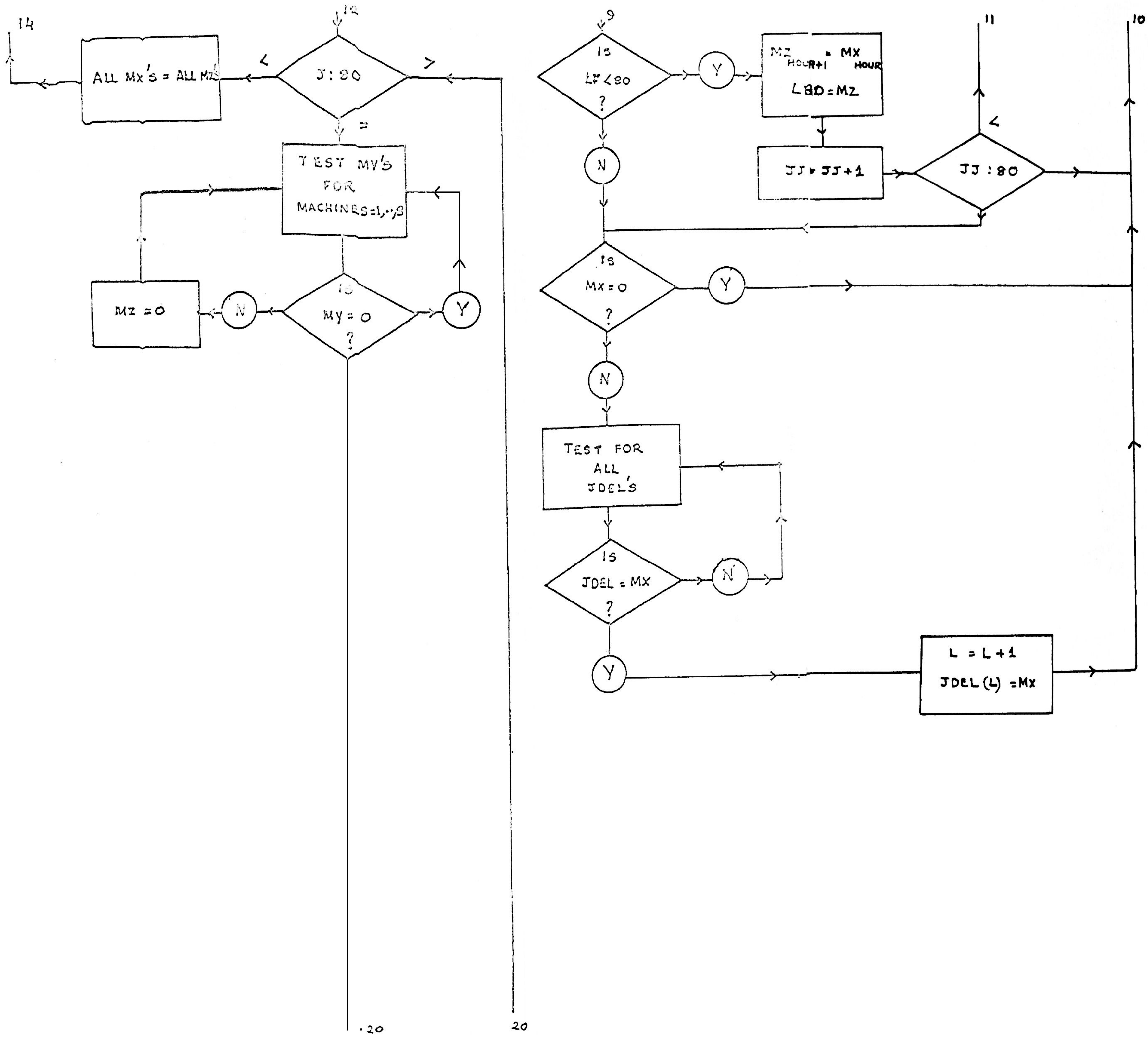

50
1
IS KSEK=0 ?
Y
KEND(K)=KSR
M = M+1
2
N
I = KSEK
KPT= NPT
IR= IR+1
KOUNT= O
3
KEND(K) = KSR
L = L+1
JDEL(L) =
NPR(M)
TEST ALL
MX'S FOR
I = IR, 80
IS
MX >0
?
Y
KOUNT=0
N
KOUNT = KOUNT+1
IS
KOUNT = KPT
?
N
Y
ORIGINAL
SCHEDULE
ALL MX'S
PUNCH
4
KSR = J
KR= J-KPT+1
SET
KSTRT(K)= KK
FOR OPERATION 1
KKS (K)= KKS (K)+
KPT
FOR
OPERATION > 1
HOUR= KR,···,KSR
MX = NPR
IR = KSR

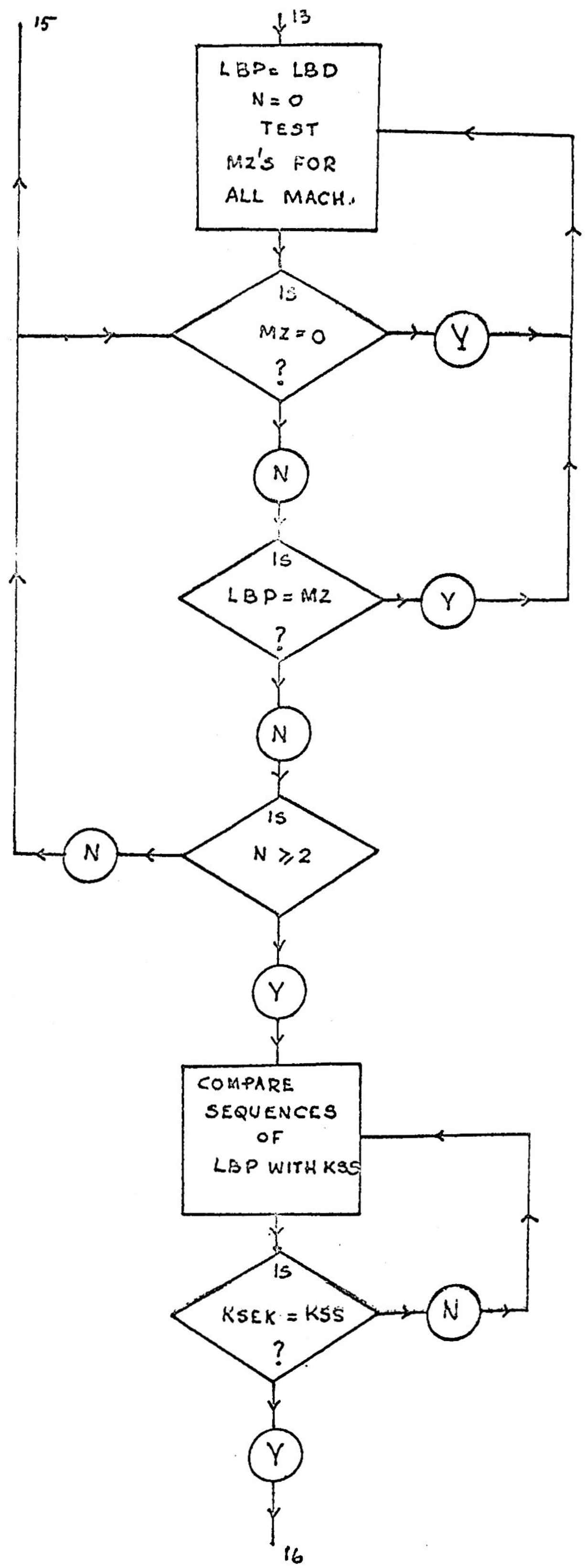

4
IDTMW =0
TEST ALL
Mx'S
IS
MX=0
?
Y
IDTMW = IDTMW+1
N
PUNCH
IDTMW, L
READ KBD
IS
KBD =0
?
Y
SET
Mz'S = Mx'S
N
SET
My'S =0
K = 1
READ KRN
WHERE 0 < KRN ≤ 80
SET
J = KRN
READ KRN
0 < KRN ≤ 8
I = KRN
6
7
20

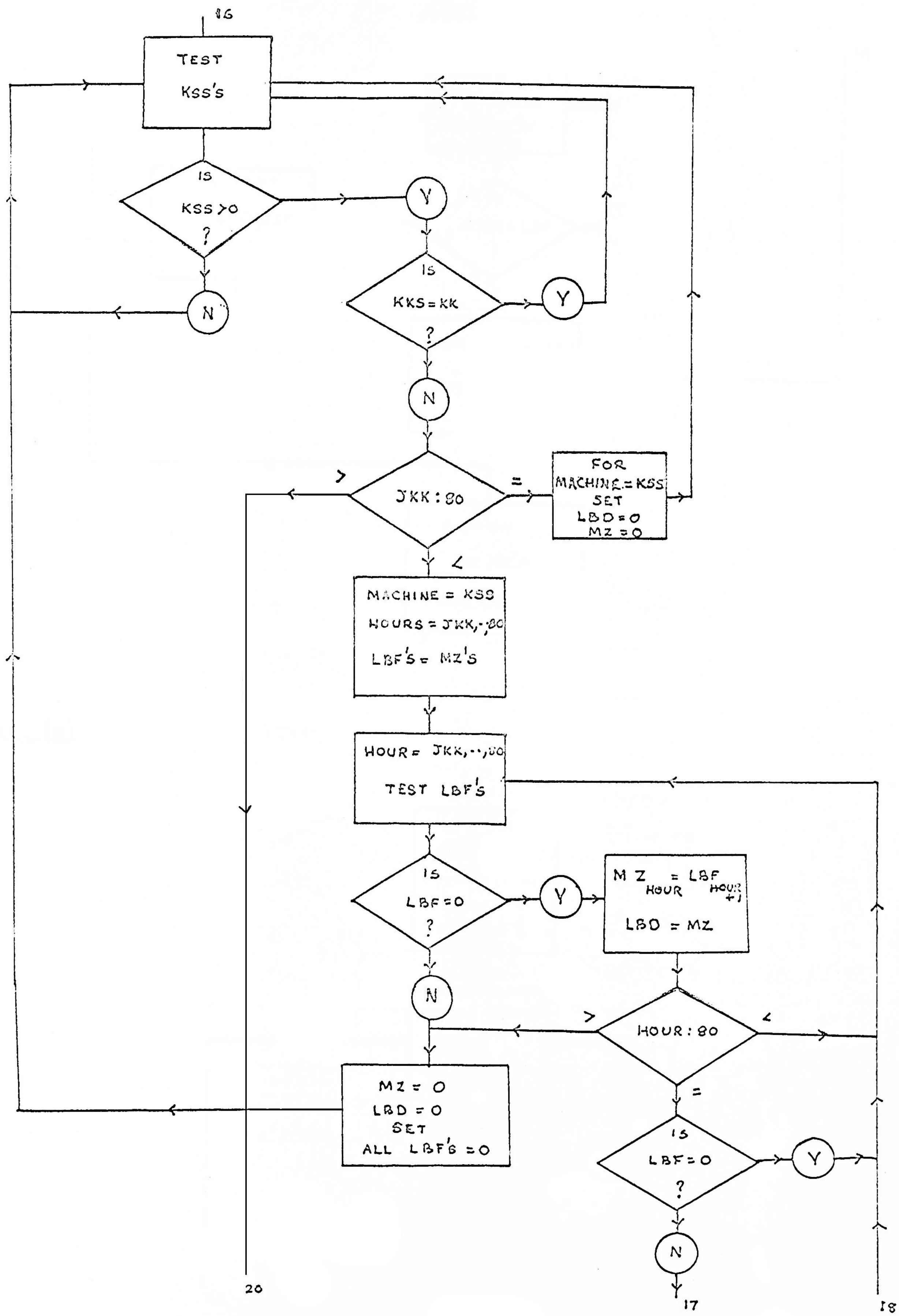
Part (b)

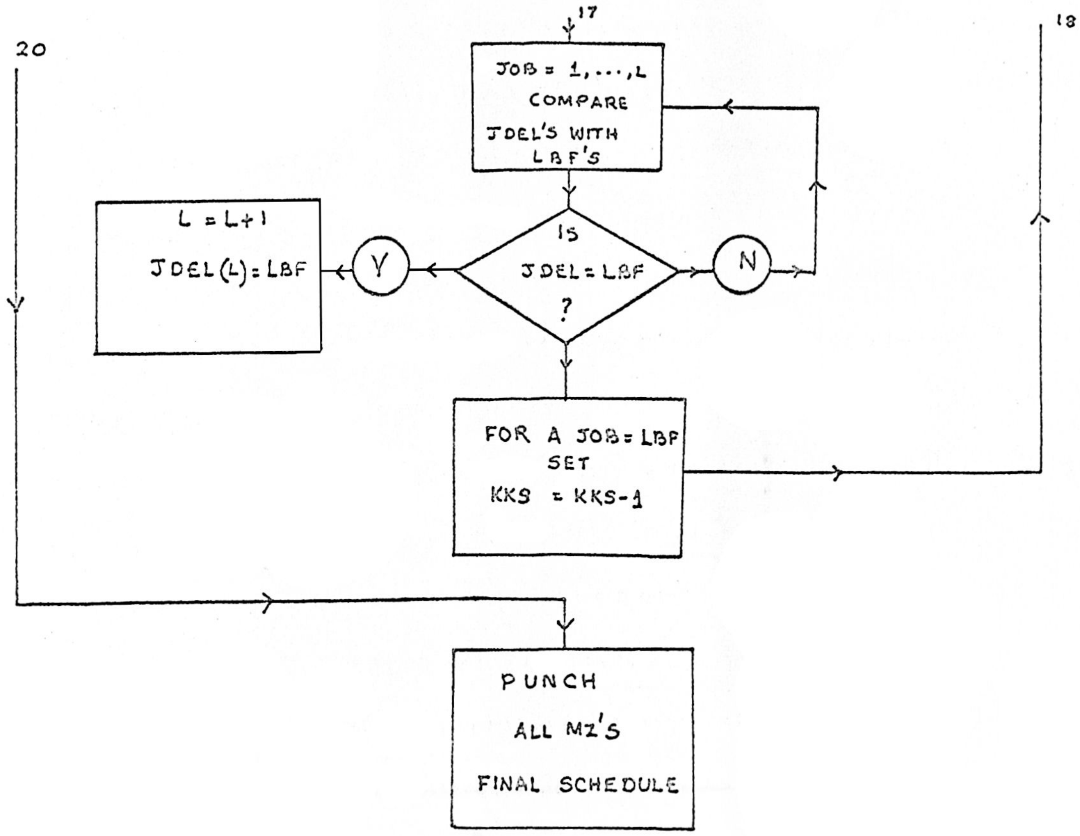

5
6
7
IS
MY = 0
?
Y
N
READ KRN
RN = KRN
RN = RN / 10,000
BDH = 0
BDS = 0
BDHF - 1
Y = e^-4
X = (4)^BDH / BDH!
FHB = XY
BDS = BDS + FHB
IS
RN ≤ BDS
?
N
BDH = BDH + 1
BDH! = BDH! · BD
IS
BDH = 0
?
Y
N
KK = J, ..., J+BDH-1
SET
My = -50
FOR K = 1
J = LF
K = K + 1
IS
K = KBD
?
N
Y
8


8
SET ALL KSS'S=0
IS LF FIRST HOUR ?
N
Y
LK = LF-1 FOR HOURS= 1,···,LK AND ALL MACHINES MZ'S = MX'S
HOURS = LF,···, 80
MACHINES= 1,··· 8
SET
MZ'S = 0
LBD'S = 0
LBF'S = 0
FOR HOUR = LF AND MACHINE=1,··8 TEST MY'S
IS MY < 0 ?
HOURS = J,···, 80 AND THE COR. MACHINE SET MZ'S = MX'S
IS LF = 80 ?
SET MZ = 0 LBD = 0
HOUR = HOUR+1
MACHINES=1,··,8
TEST ALL LBD'S
IS MX = 0 ?
IS LBD=0 ?
HOUR = HOUR + 1 IS MX = 0 ?
J = HOUR, ···, 80
MZ'S = MX'S
J = J+1
14
20
15
13
12
11
9
10


14
12
9
11
10
ALL MX'S = ALL MZ'S
<
J : 80
>
=
TEST MY'S FOR MACHINES=1,..,8
IS MY = 0 ?
N
Y
MZ = 0
IS LF < 80 ?
Y
N
MZ HOUR+1 = MX HOUR
LBD = MZ
JJ = JJ + 1
JJ : 80
<
IS MX = 0 ?
Y
N
TEST FOR ALL JDEL'S
IS JDEL = MX ?
N
Y
L = L + 1
JDEL (L) = MX
20
20

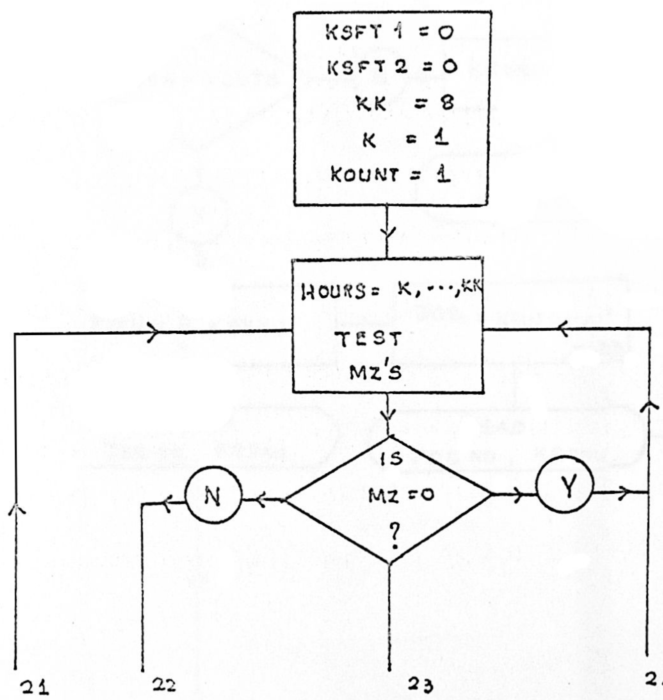

15
13
LBP = LBD
N = 0
TEST
MZ'S FOR
ALL MACH.
IS
MZ = 0
?
Y
N
IS
LBP = MZ
?
Y
N
IS
N ≥ 2
N
Y
COMPARE
SEQUENCES
OF
LBP WITH KSS
IS
KSEK = KSS
?
N
Y
16


16
TEST
KSS'S
IS
KSS > 0
?
Y
N
IS
KKS = KK
?
Y
N
JKK : 80
>
=
<
FOR
MACHINE = KSS
SET
LBD = 0
MZ = 0
MACHINE = KSS
HOURS = JKK,..,80
LBF'S = MZ'S
HOUR = JKK,..,80
TEST LBF'S
IS
LBF = 0
?
Y
N
MZ HOUR = LBF HOUR+1
LBD = MZ
HOUR : 80
>
<
=
MZ = 0
LBD = 0
SET
ALL LBF'S = 0
IS
LBF = 0
?
Y
N
20
17
18

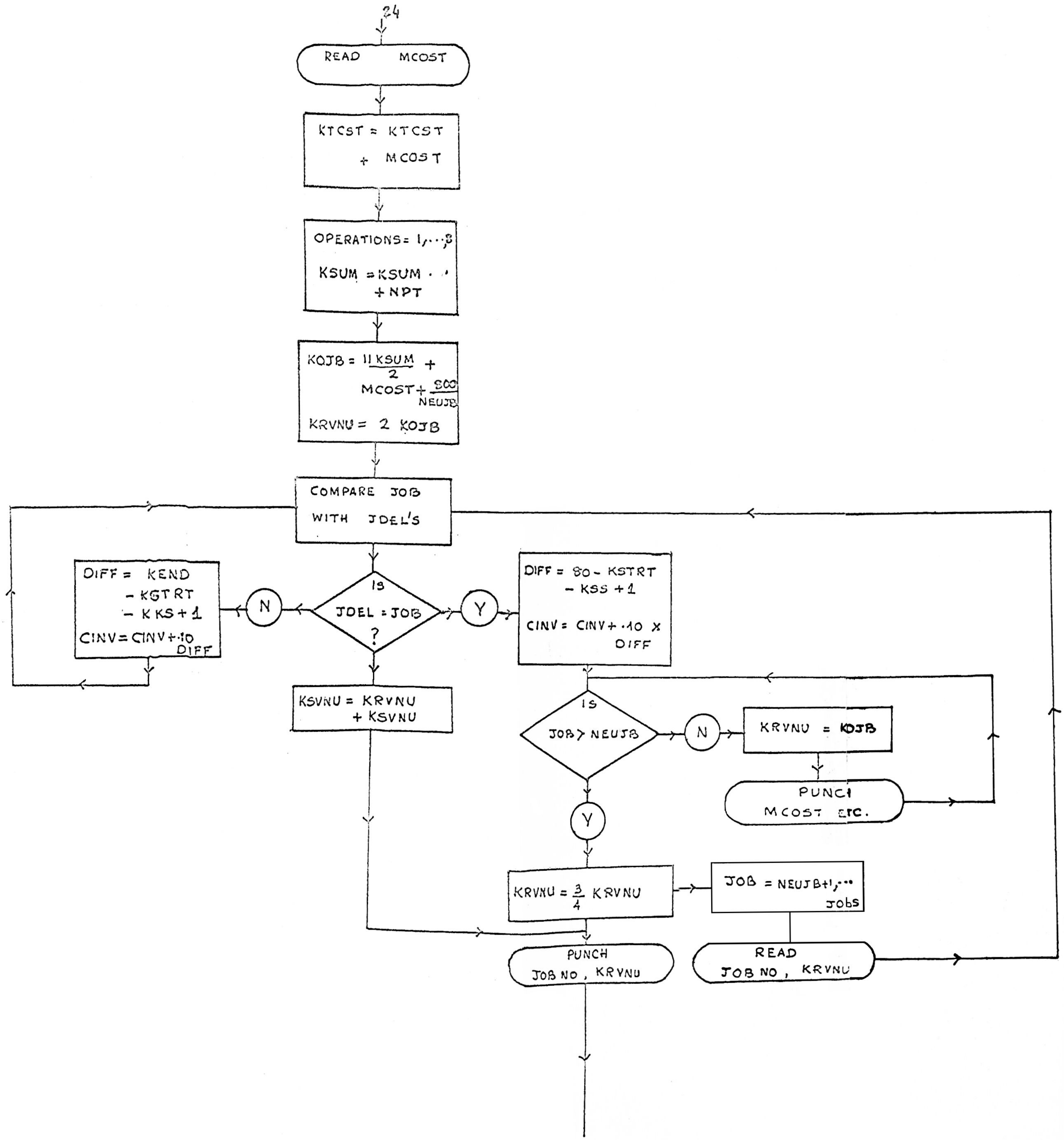


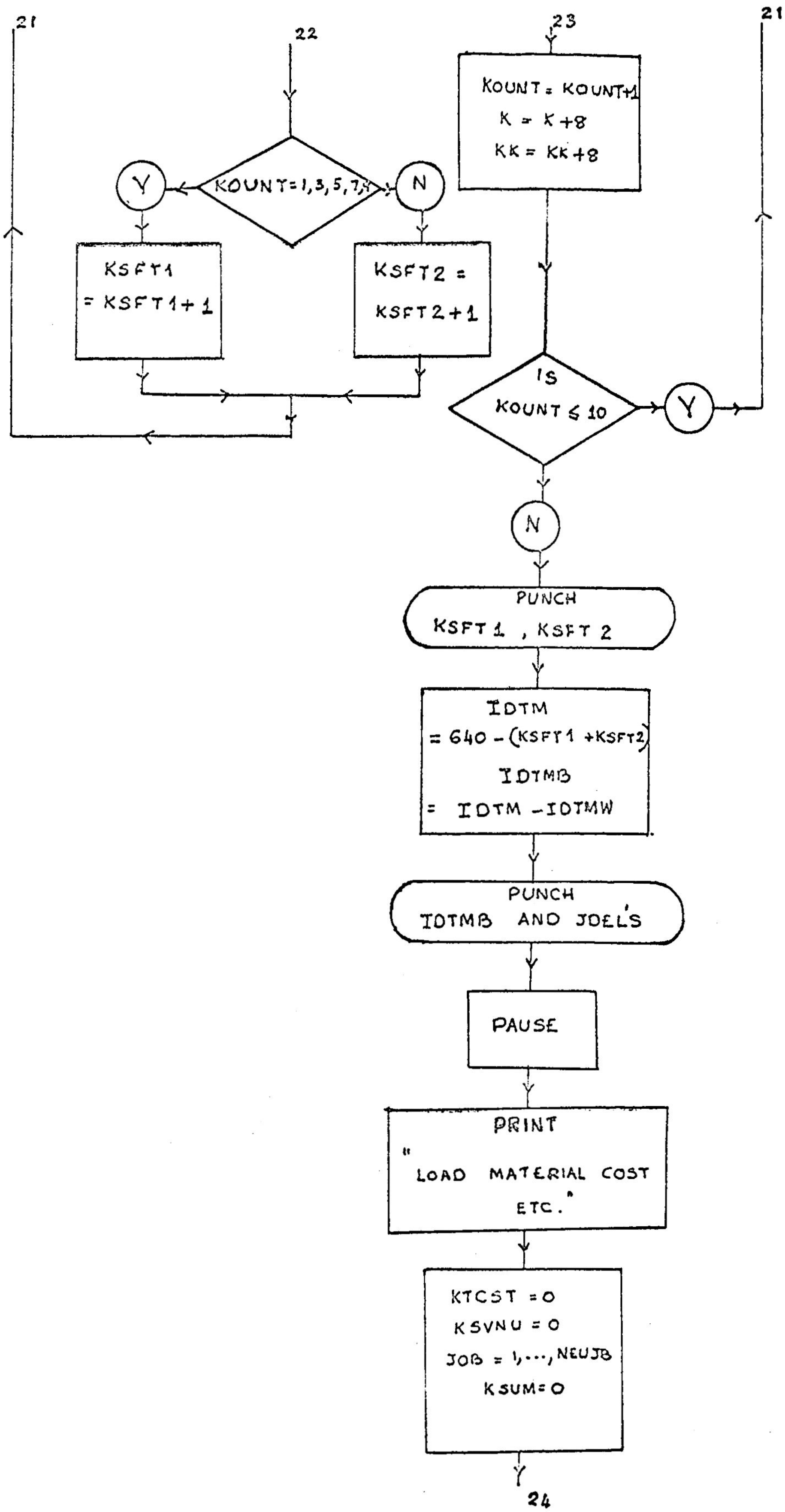
17
18
20
JOB = 1,...,L
COMPARE
JDEL'S WITH
LBF'S
L = L+1
JDEL(L) = LBF
Y
IS
JDEL = LBF
?
N
FOR A JOB = LBF
SET
KKS = KKS-1
PUNCH
ALL MZ'S
FINAL SCHEDULE

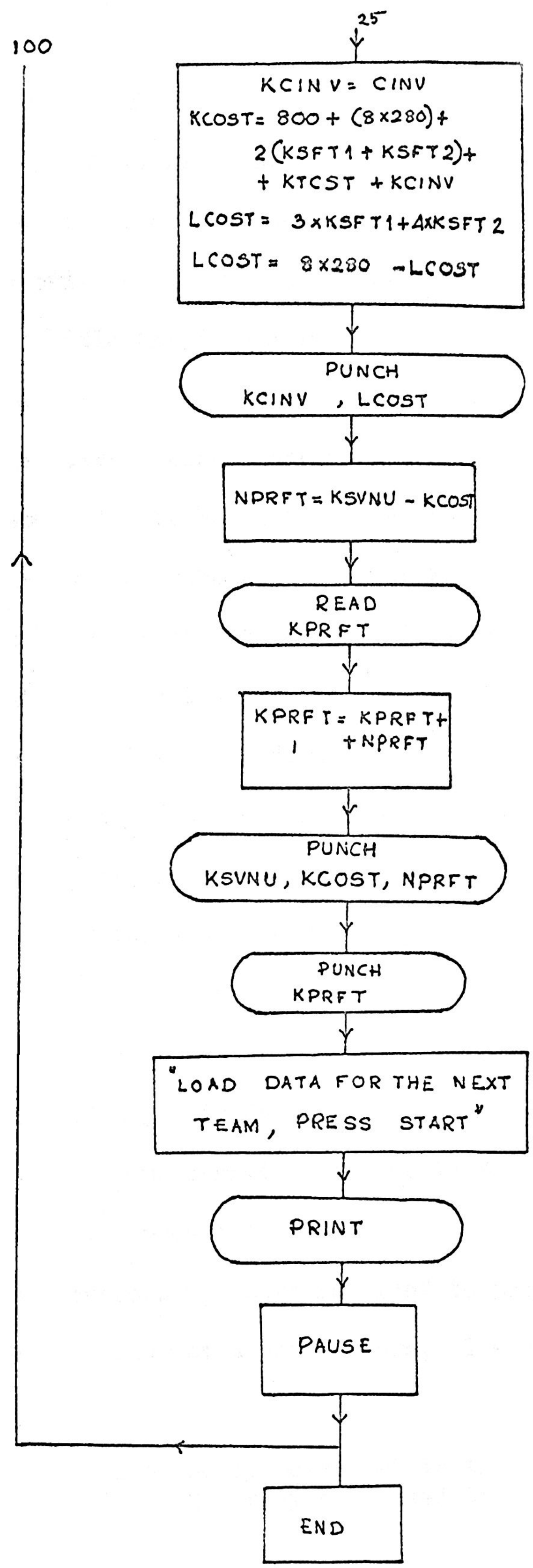
Part (c)



KSFT 1 = 0
KSFT 2 = 0
KK = 8
K = 1
KOUNT = 1
HOURS = K,...,KK
TEST
MZ'S
IS
MZ = 0
?
N
Y
21
22
23
24


24
READ MCOST
KTCST = KTCST + MCOST
OPERATIONS= 1,...,8
KSUM = KSUM + NPT
KOJB = 11 KSUM / 2 + MCOST + 800 / NEUJB
KRVNU = 2 KOJB
COMPARE JOB WITH JDEL'S
IS JDEL = JOB ?
N
Y
DIFF = KEND - KSTRT - KKS + 1
CINV = CINV + .10 DIFF
DIFF = 80 - KSTRT - KSS + 1
CINV = CINV + .10 x DIFF
KSVNU = KRVNU + KSVNU
IS JOB > NEUJB
KRVNU = KOJB
PUNCH MCOST ETC.
KRVNU = 3/4 KRVNU
JOB = NEUJB+1,... Jobs
PUNCH JOB NO, KRVNU
READ JOB NO, KRVNU


21
22
23
21
KOUNT = KOUNT+1
K = K+8
KK = KK+8
Y
KOUNT=1,3,5,7,9
N
KSFT1
= KSFT1+1
KSFT2 =
KSFT2+1
IS
KOUNT ≤ 10
Y
N
PUNCH
KSFT1 , KSFT 2
IDTM
= 640 - (KSFT1 +KSFT2)
IDTMB
= IDTM -IDTMW
PUNCH
IDTMB AND JDEL'S
PAUSE
PRINT
"LOAD MATERIAL COST
ETC."
KTCST =0
KSVNU = 0
JOB = 1,...,NEUJB
KSUM=0
24


25
100
KCINV = CINV
KCOST = 800 + (8x280) + 2(KSFT1 + KSFT2) + + KTCST + KCINV
LCOST = 3xKSFT1+4xKSFT2
LCOST = 8x280 - LCOST
PUNCH KCINV , LCOST
NPRFT = KSVNU - KCOST
READ KPRFT
KPRFT = KPRFT + 1 + NPRFT
PUNCH KSVNU, KCOST, NPRFT
PUNCH KPRFT
"LOAD DATA FOR THE NEXT TEAM, PRESS START"
PRINT
PAUSE
END

A Glossary of Symbols*

BD	Number of break downs
BDF	Factorial value of BD
BDH	Duration of a break down in hours
BDHF	Factorial value of the duration of a break down in hours.
IDTM	Total idle hours in a week
IDTMB	Idle hours caused due to break downs
IDTMW	Idle hours without considering break downs
JBPRV	Number of jobs handled a week previous to the week of play
JDEL (I)	Delayed job, where, I = Job No.
JOBS	Total number of jobs handled in a week
KBD	Number of break downs
KBDH	Duration of a break down
KCINV	Total cost of the in-process inventory
KCOST	Sum of all the material costs
KEND (I)	Last hour of a job in the shop, where, I = Job No.
KOJB	Estimated cost of a job
KRN	A random number
KRVNU	Price of a job in dollars
KSVNU	Total revenue earned in a week in dollars
KPRFT	Cumulative profit in dollars
KSUM	Total processing hours required to produce a job
KSTRT (I)	Starting hour of a job, where, I = Job No.

---

* This section covers the symbols used in the computer programs to define the main variables. Minor symbols used to redefine these variables have been ignored.

A Glossary of Symbols (continued)

KSEK (I, J)	Operational sequence of a job, where, I = Job No., and J = Operation No.
KKS (I)	Total processing hours consumed by a job in a week, where I = Job No.
KSS (I)	Temporary locations used in Program II, where, I = 1, ..., 8
L	Number of delayed jobs
LBD (I, J)	Temporary locations used in Program II, where, I = 1,..., 8 and J = 1,..., 80
LBF (I)	Temporary locations used in Program II, where, I = 1,..., 80
LF	First hour of break down
LCOST	Total labor cost
MX	A cell value of the matrix for the original schedule
MY	A cell value of a matrix to locate the break downs
MZ	A cell value of the matrix for Final Schedule
NEUJB	New orders received in a week
NOP	Number of operations required to produce a job
NPRFT	Net profit of a team in a week

A PRODUCTION SCHEDULING SIMULATOR

by

HARIPADA NANDA

B. Sc. (Engineering) Mechanical, Utkal University, India, 1961

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1963

Approved by:

____

Major Professor

The object of this thesis was to develop a computer algorithm to be used in playing a job-shop scheduling game aimed at developing the intuition of the participants in realizing the relative importance of jobs in a job-shop.

The job-shop conceived is one with eight machines scheduled for a five-day week satisfying a varying demand of twelve to fifteen jobs a week.

The participants of the game are supposed to decide upon the relative importance of jobs by assigning priority numbers to different jobs from their individual point of view. The job-shop is scheduled separately on the basis of the decisions made by each player and the schedules developed are evaluated by a number of measures of effectiveness. The results are handed over to the players at the end of each decision interval. The players are supposed to gain experience from the results of their previous decisions to make priority decisions in a more effective way so as to receive better results.

The computer program was written in two parts; one to generate orders for each week and another to simulate the production shop. Second part of the program consisted of three subsections. The first subsection was meant to schedule the job-shop, the second was to generate random break downs over such a schedule and the last was to compute the results of the simulation and report them to the players.

The program was written in Fortran II machine language for the IBM 1620 Digital Computer with 60 K storage.