

CELLULAR SCHEDULLING OF BATCH PRODUCTION USING A COMPUTER

by

E. A. Onyeagoro

Department of Mechanical Engineering

University of Nigeria, Nsukka

(Manuscript received May I, 1980 and in revised form November 2, 1980)

ABSTRACT

This paper describes a simple computer programme developed for a medium-sized engineering firm manufacturing various types and sizes of machine pinions. The programme is capable of assigning operators to their appropriate machines for the purpose of machining components in a manufacturing system designed on the Group Technology principles. The particular machine or machines an operator can operate were already defined as part of an input data. A number of test runs were made with different batch sizes, firstly, for the purpose of ensuring the proper operation of the computer programme, and secondly to obtain sufficient data on man/machine interaction which is necessary in the analysis in this study.

1. INTRODUCTION

The problems associated with production scheduling have existed for years in a large number of manufacturing industries. These problems differ greatly from one firm to another, with various types of constraints and with pay offs dependent upon differing criteria. Usually, these criteria are based on cost, and the solution that ensures the attainment of the objectives of the scheduling method at the lowest possible costs is considered the best one [1]. The principal elements of costs which are of interest in production scheduling are the: (i) Machine set-up costs (ii) Machine running costs (iii) Stock-holding costs.

To minimise the total machine set-up costs, where the set-up costs for any particular component depends on the preceding batch component in the schedule, the order or sequence of batches should then be arranged to achieve this objective. To minimise machine running costs, it would be necessary to decide-on the appropriate batch size taking into account+, technological restrictions and considerations and at the same time to fix the production schedule so as to

achieve maximum utilization of all machines involved in the production process. In order to minimise stock-holding costs, a low-inventory policy must. be applied. This requires that all raw materials, work-in-progress and finished goods inventories must be kept to a minimum level which would satisfy the demand requirements.

Even in a single-product situation, the combination of the above three minimisation problems would make the scheduling problem more complicated if an attempt is made to find a way of dealing with them all at the same time. This complication would be much greater in the scheduling problem is encountered in a multi-product situation. The reason being that the minimum solution of one problem usually conflicts with the solution of the other. For instance, the minimisation of the set-up costs causes a decrease in the number of cycles, hence an increase in batch size which would cause a corresponding increase in the stock-holding costs.

Traditionally, most production scheduling problems have been solved mathematically using such methods as linear programming, dynamic programming, heuristic

algorithm, queueing theory, etc. Besides the lengthy iterations these methods require to arrive at a solution, the discrete and sagacious manner in which batches of components are dispatched in a Group Technology system make the use of a computer based method more realistic and practical [2].

In the context of the present paper, "Group Technology" (GT) can be described in general as a complete manufacturing system where families of similar components are formed and processed on groups of non-identical machines (group being one machine upwards) in order to extend the technical and economical advantages of mass or flow production to those associated with batch production and jobbing. The formation of these components into families does not necessarily mean that they are similar in shape, but are so formed by virtue of their having all or some common machining operations.

A "Cell" here is taken to be a self-contained processing unit which is made up of one machine or a number of machines (identical or non-identical) required to complete the process of all the components assigned to it.

The company studied is a medium sized U.K. engineering concern, manufacturing different types and sizes of machine pinions. The machine shop is functionally laid out, with machine tools of a similar type being contained in the same bay (Fig. 1). As a result, groups of similar machine tools are duplicated in order to satisfy the requirement of two separate but basically similar products.

Components machined in this machine shop varied both in types and sizes with each component having a variety of operations performed upon it by one special purpose machines. The provision for machining these components demanded that the production level must be based on a one-year forecast of demand. Also, there were difficulties in meeting fluctuations of market demand, for customers' orders tended to be non-repetitive and the machine shop did not have the advantage of a buffer stock for helping production levelling.

A large part of the company's range of components is manufactured for stock and the assembly lines are supplied from finished component stocks. About 90% of components held in stock are used tip in this way, and the remaining 10% are held as spares.

2. STUDY BACKGROUND

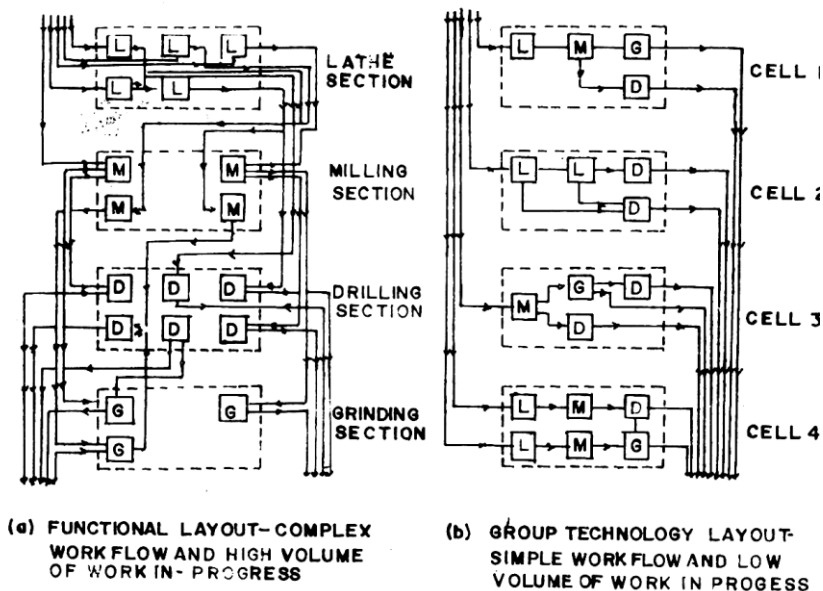


FIGURE 1. TYPES OF MACHINE LAYOUT

On average the company's products have a manufacturing lead time of approximately nine to eighteen months, and it is this fact that is recognised as being responsible for a number of problems. Therefore, one of the company's main objectives, and hence this study, is to introduce new manufacturing concepts which would reduce the manufacturing lead time to a period not more than six months. In order to achieve this objective it is proposed to adopt a philosophy which would apply the Group Technology (GT) techniques, and set up piece part manufacture in cells. The manufacturing cell used in this study was a simple one, comprising of six machines and three operators. The machines are arranged in a straight line, to allow for easy input and output of raw materials and finished components respectively (Figure 2).

3. SELECTING A BATCHING AND SEQUENCING METHOD.

The Economic Batch Quantity (EBQ) rule is used in obtaining the batch size to be issued for machining. Also, the simple techniques for solving the batch sequencing problem in a group flowline situation [3] is used in this study. Having obtained the required batch sizes and determined the order in which those batches of components are to be issued to the cells for machining, a computer programme, which is capable of assigning operators to their appropriate machines, was written. The particular machine or machines an operator can operate were already defined as part of an input data, and a number of test runs were made with different batch sizes, fundamentally for the purpose of ensuring the proper operation of the computer programme.

4. EXPLANATION OF FLOW CHART.

The flow chart for the computer programme written for this study is shown in Appendix 'A'. The numbers refer to operations in the chart.

The programme, CELLSCHED (an acronym for CELLULAR SCHEDULLING), is written in FORTRAN IV, and in three parts which must be run together

The first part of the programme deals mainly with the input/output routines, initialisation and earth procedures (from operation I to operation 18, with exception of operation 13). At operation 13, a check is made to determine whether or not the batch size is zero. If the answer to the question in operation 13 is no, control will be transferred through operation 14 to operation 17, where a second check is made. If the answer to the second check is no, the programme goes into loop, back

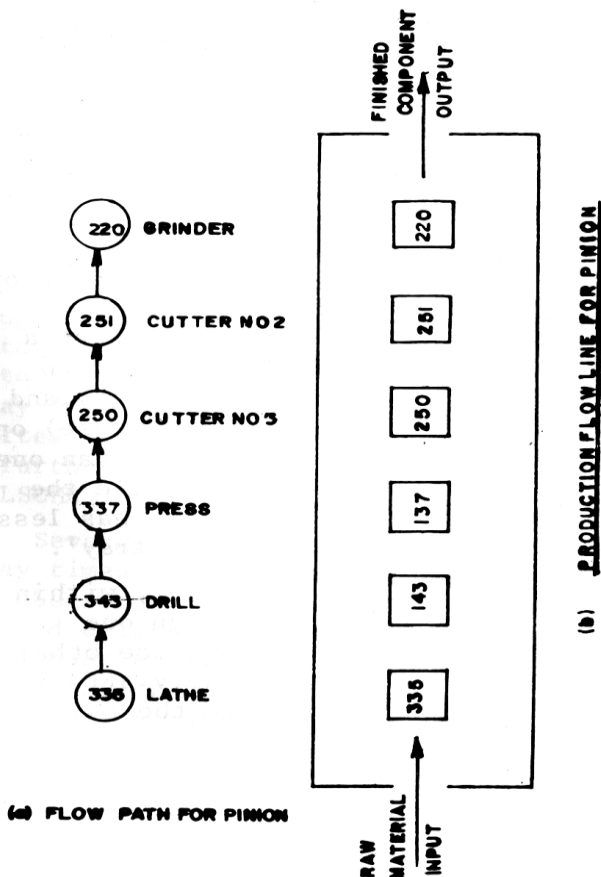


Fig. 2. One-Direction Flow System

to operation 12; if the answer is yes, subroutine Rule 1 is called in operation 19. If , - however, the answer to the check in operation 13 is yes, another check will be made in operation 25, a no answer to this check will cause the programme to terminate in operation 33 after printing the output in operation 32; but if the answer is yes, Subroutine Rule 2 would be called in operation 26.

The second part which is the Subroutine Rule 1 (from operation 20 to operation 24) is called whenever the production (or machining) rate of one or more machines (except the first machine) is equal to or greater than the quantity of components in the out tray, of the preceding machine.

It must be understood that components in the 'in tray' of a machine are ready to be machined by that machine, while those in the 'out tray' have already been machined and are ready to be transported to the 'in tray' of the next machine or to the stores as finished components.

The third and last part of the programme is the subroutine Rule 2 (from operation 27 to operation 31). This Subroutine is used as a 'mopping up operation'~ and is only called when there is some clearing work to be done. That is, Subroutine Rule 2 is only called either when the:

- (i) Batch size is equal to zero and still there are components remaining in the 'in tray' of one or more machines to be processed.
- (ii) Batch size is greater than zero but less than the production (or machining) rate of first machine,
- (iii) Quantity of components in the 'in tray' of one or more machines is greater than zero and, at the same time, those of the 'out tray' of every machine is equal to zero

- (iv) Quantity of components in the 'in tray' of one machine (or more) is less than its production rate.

At the Subroutine Rule 2 or 'mopping up operation' stage, when less than one hour's working is available on each machine, either all the remaining components in the batch size (in the case of the first machine) or all the components in each of the 'out trays' of the subsequent machines (in the case of all other machines except the last one) are moved into the 'in tray' of the forward machine, which already has an operator on it, moving from left to right.

Thus, the procedure for machining components under Subroutine Rule 1 is exactly the opposite to that of Subroutine

Rule 2; in the former, the components are first moved into the 'in tray' of a machine before an operator is assigned to it, while in the latter, the operator is first assigned to a machine before the component is moved into its 'in tray'.

It is essential that the above requirements are met, since the CELLSCHED programme reflects the production rat~ of each machine which is expressed in components per hour. For this reason, a facility is provided in the programme which recorded both an idle time and .a message about a tie on an operator, by placing an asterisk against an operator number and by printing a statement about a tie on an operator respectively, whenever these actions are necessary.

An operator is said to be idle if having been assigned to a machine, finds that the machine cannot be operated because of a constraint, e.g. waiting for work to arrive from the rear machine. Work here is taken to mean either raw material ready to go through

the first machining operation, or components which have already been machined on one machine and are about to be machined on the other (i.e. work - in - progress). Whilst a tie is said to have occurred on an operator if two or more machines to which the operator has been scheduled to operate have one or more hours work in their respective 'in trays', at the same time. In such a situation, the operator is assigned to the machine furthestmost to the right, and would return to that (or one of those) on the left if it has equal or more than one hour's work in its 'in tray', while the machine furthestmost to the right has less than one hour's work in its 'in tray'.

It must be emphasised that, within a cell, it is possible to assign one operator to three machines and the other to one machine, without drastically affecting their overall work balance in the cell.

5. DISCUSSION

The computer programme developed in this study is to help the management of a medium-sized U.K. manufacturing company to know the state of its manufacturing cell, hour by hour. Since the clock-time in the programme is updated hourly, to equal the arrival time of the next batch of components.

The programme facilitates rapid recording of and easy reference to all the information which are produced in a single presentation concerning the performance of both operators and machines. The programme is easily understood, does not require extensive training' of personnel, is inexpensive and produces practical solutions at short notice.

For each demand schedule, the CELLSCHED programme was used to produce a simulation run of individual batch sizes which make up the batched schedule of a

particular size of component~ as shown in Appendix "B". By matching individual operators' number with the appropriate machine number(s) within a cell, the CELLSCHED programme was able to assign each operator to the correct machine.

A time delay mechanism was written into the CELLSCHED programme so that, for any run, the second machine would start at a predetermined time, depending on the rate of production of the first machine. If the first machine had a much higher rate of production than the second machine, the latter was delayed for at least one hour before it was started, in order to build up sufficient components for it to start. On the other hand, if the production rate of the second machine was found to be greater than that of the first, the former was delayed sufficiently to allow enough components to be processed on the latter. So that, there was no fixed time up to which the second machine must be delayed after the first machine had started. This information was provided as further part of an input data to the CELLSCHED programme.

Several runs were made with varied delay times on the first machine, and it was observed at the end of each run that induced delays of between one and six hours made no significant difference in the overall total idle time recorded against each machine Appendix "C". This is due to the fact that earlier delay on the first machine is later spread out over the time horizon, as the machining operation continually progressed.

Therefore, it is important that the time delay is introduced earlier in the process rather than later to minimise the overall effect of the idle time. For example, a more effective labour utilisation would be achieved if there was a continuation of operation (even with

an initial delay' on the machine) than when the operation was intermittent and without initial machine delay.

Although by using the CELLSCHED programme in simulating the machine shop of the company studied, it was possible to obtain in a single presentation a number of useful information concerning the performance of both operators and machines, the following restrictions could reasonably be applied to it:

- i. the CELLSCHED programme handles only one batch of components at a time;
- ii. the operation sequence for each component and the sequencing order of batches of components will be pre-determined;
- iii. because the timing mechanism in the CELLSCHED programme is updated hourly, there may be an occasion when during the mopping up operation, the actual quantity of components to be machined may be less than an hour's work even though the time for machining them will be taken as one hour.

In practical terms however, the penalty for applying these restrictions is insignificant when compared with that for not having an effective control of the utilisation of operators and machines. Also since the above restrictions are part of the input data to the CELLSCHED programme, any costs that they may be incurred form part of the computer running cost, which is very small.

4.

6. CONCLUSION

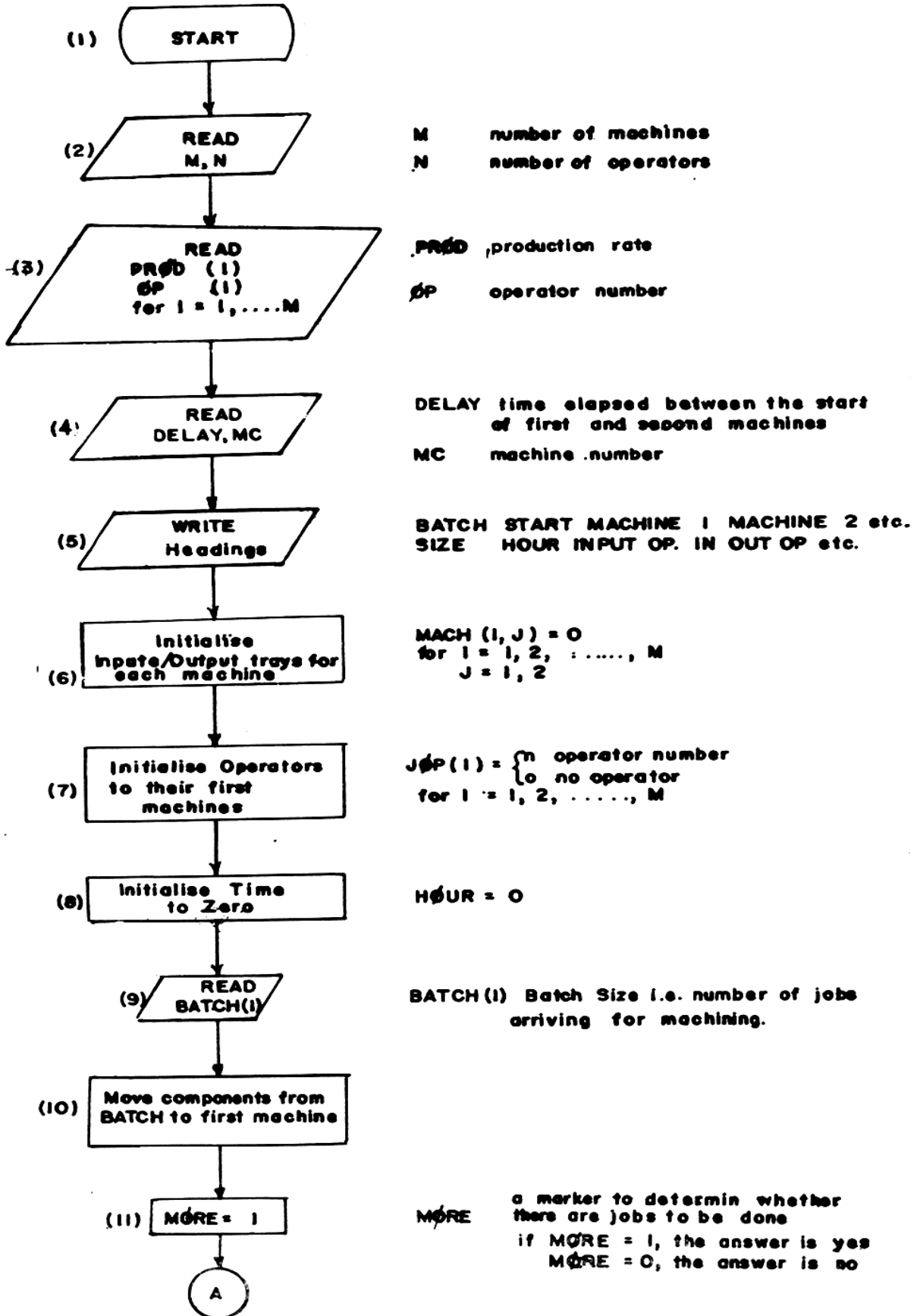
The CELLSCHED programme developed in this study will provide management with an hourly information, in a single presentation, concerning the state of the manufacturing cell. The possession of this information will enable management to have an effective control of the utilisation of operators and machines. It is highly unlikely that the above information could be obtained in a single solution if any one of the other methods (e.g. Linear Programming, Dynamic Programming, Heuristic Algorithm, Queueing Theory, etc.) of assigning operators to machines was used.

It is acknowledged that by using the CELLSCHED programme optimum is not obtained but procedure is relatively simple, easy to apply, does not require extensive training of personnel, is inexpensive and provides necessary information for obtaining practical solutions at short notice.

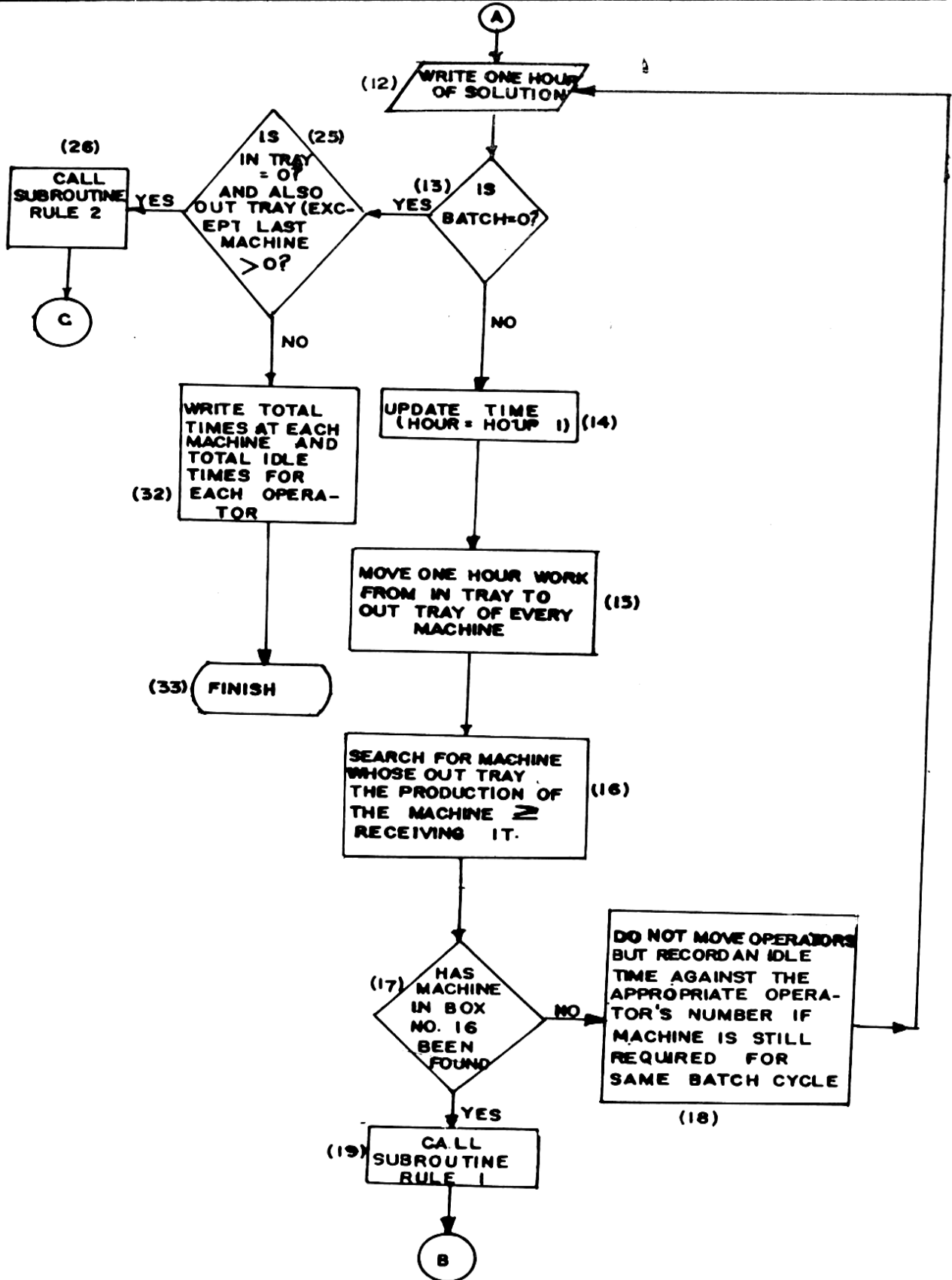
REFERENCES

1. Eilon, S., Elements of Production Planning and Control. The Macmillan Company, 1969.
2. Onyeagoro, E.A. "Cellular Scheduling for Batch Production", Ph.D. Thesis, University of Aston in Birmingham, 1977.
3. Petrov, V.A. "Flowline Group Production Planning. English Translation Published by Business Publications Ltd., 1968.

APPENDIX "A" INPUT/OUTPUT/ INITIALISATION ROUTINE

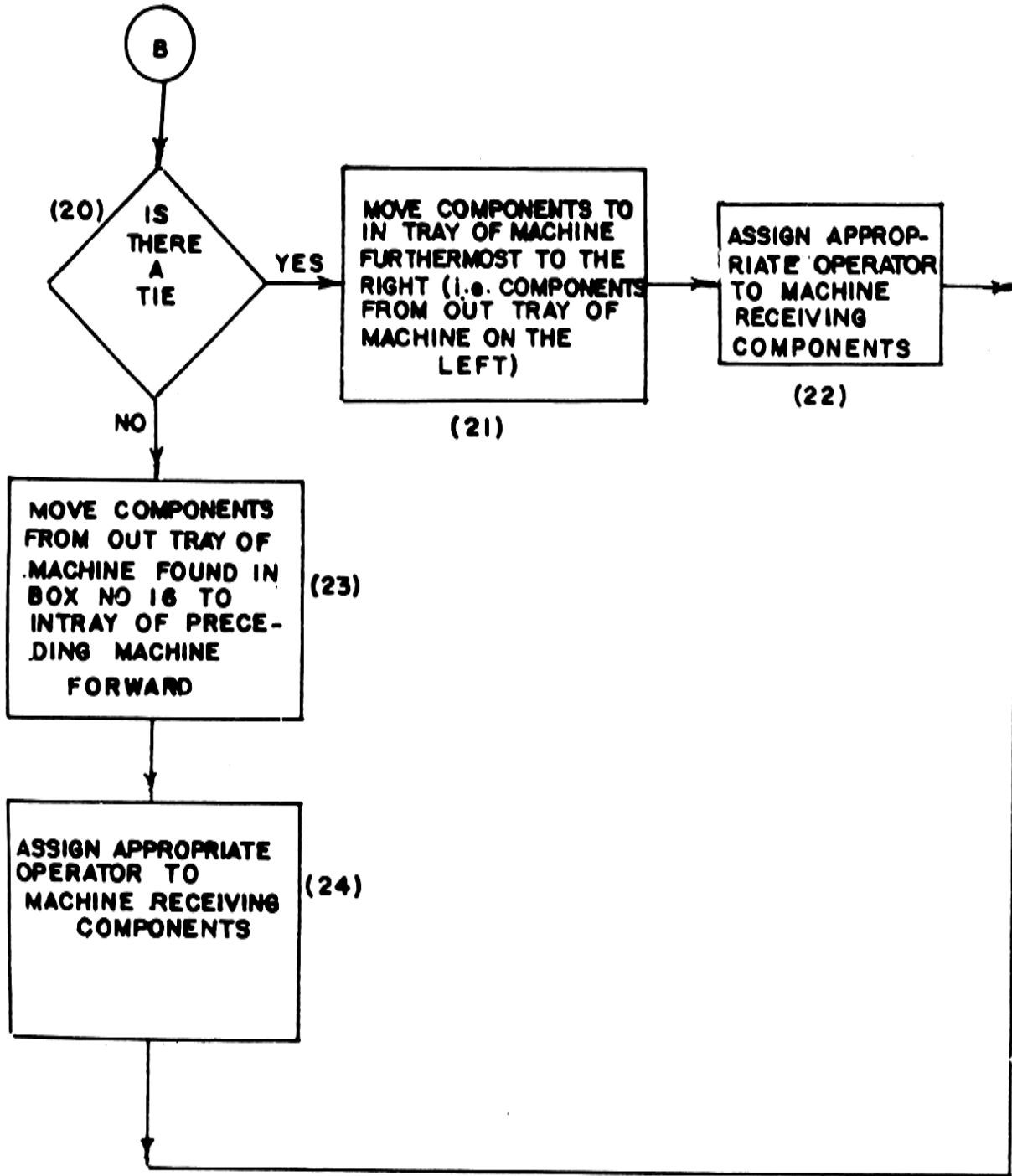


SCHEDULED OUTPUT LINE OF SOLUTION AND TEST FOR FINISH APPENDIX "A" Contd.

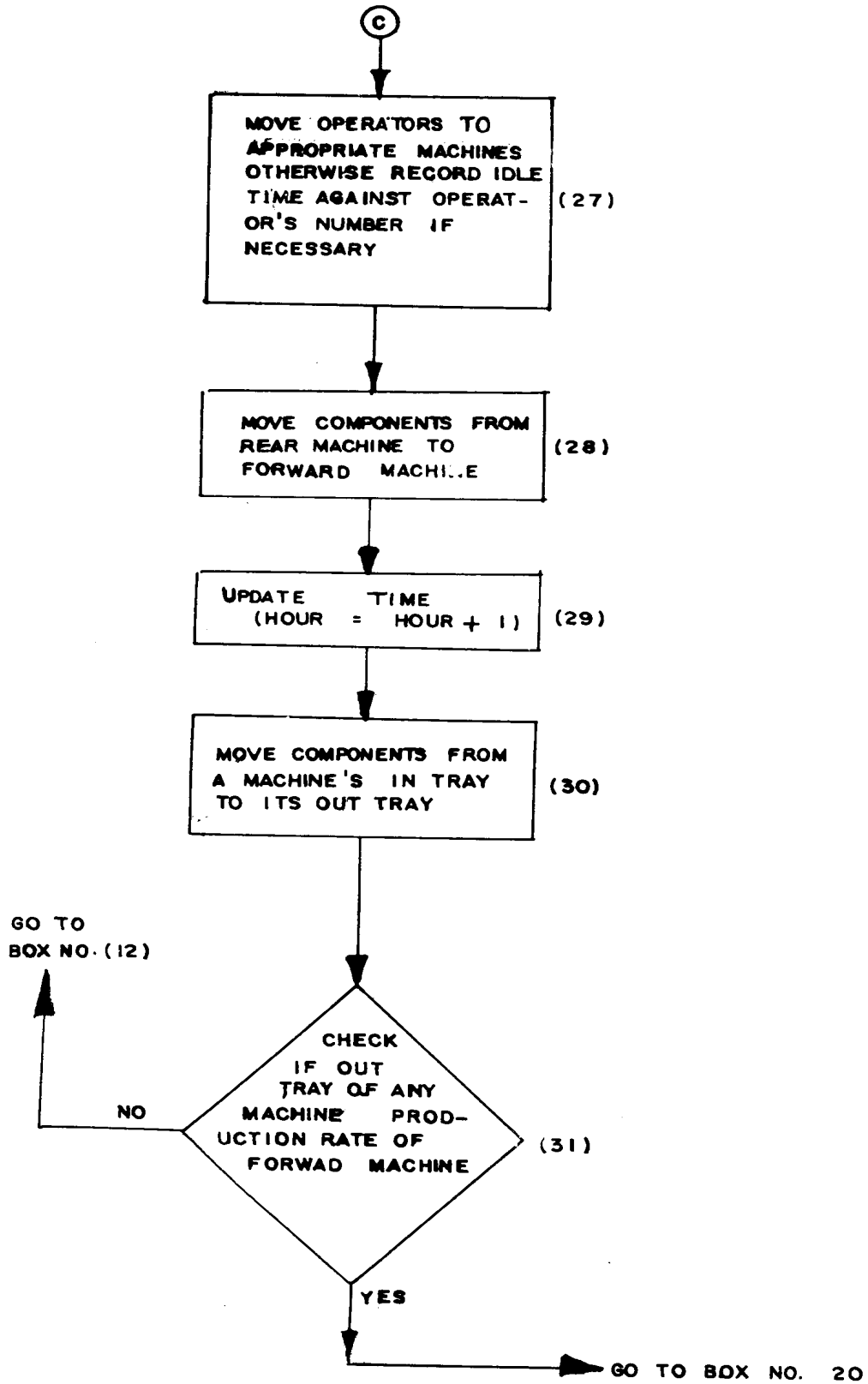


APPENDIX "A" Contd. CELLSCHED SUBROUTINE RULE 1

GC TO
BOX NO



APPENDIX "A" Contd: CELLSCHED SUBROUTINE RULE 2



A P P E N D I X B

BATCH SIZE	START HOUR	MACHINE 1			MACHINE 2			MACHINE 3			MACHINE 4			MACHINE 5			MACHINE 6						
		IN	OUT	OP	IN	OUT	OP	IN	OUT	OP	IN	OUT	OP	IN	OUT	OP	IN	OUT	OP				
653	1	9	0	1*	27	0	0	2*	41	0	0	0	12	0	0	0	20	0	0	29	0	0	0
644	2	9	9	1	0	0	2*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
635	3	9	18	1	0	0	2*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
626	4	9	27	1	0	0	2*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
617	5	9	36	1	0	0	2*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
608	6	9	45	1	0	0	2*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
599	7	9	27	1	0	27	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
590	8	9	9	1	0	54	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
581	9	9	18	1	0	13	0	0	41	2	0	0	0	0	0	0	0	0	0	0	0	0	0
572	10	9	27	1	0	13	0	0	29	2*	0	12	3	0	0	0	0	0	0	0	0	0	0
563	11	9	9	1	0	40	2	0	17	0	0	24	3	0	0	0	0	0	0	0	0	0	0
554	12	9	18	1	0	40	0	0	5	0	0	16	3	0	20	2	0	0	0	0	0	0	0
545	13	9	27	1	0	40	0	0	5	0	0	16	3*	0	20	2*	0	0	0	0	0	0	0
536	14	9	9	1	0	67	2	0	5	0	0	16	3*	0	20	0	0	0	0	0	0	0	0
527	15	9	18	1	0	26	0	0	46	2	0	16	3*	0	20	0	0	0	0	0	0	0	0
518	16	9	27	1	0	26	0	0	34	2*	0	28	3	0	20	0	0	0	0	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					2																		
509	17	9	36	1	0	26	0	0	22	0	0	20	3	0	40	2	0	0	0	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					2																		
A TIE HAS OCCURED WITH OPERATOR					3																		
500	18	9	45	1	0	26	0	0	22	0	0	0	0	0	31	2	0	29	3	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					3																		
491	19	9	27	1	0	53	2	0	22	0	0	0	0	0	2	0	0	58	3	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					2																		
482	20	9	36	1	0	12	0	0	31	2	0	12	3	0	2	0	0	58	0	0	0	0	0
473	21	9	18	1	0	39	2	0	39	0	0	24	3	0	2	0	0	58	0	0	0	0	0
464	22	9	27	1	0	39	0	0	27	0	0	16	3	0	22	2	0	58	0	0	0	0	0
455	23	9	9	1	0	66	2	0	15	0	0	28	3	0	22	0	0	58	0	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					2																		
446	24	9	18	1	0	66	0	0	3	0	0	20	3	0	42	2	0	58	0	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					2																		
437	25	9	27	1	0	56	0	0	3	0	0	0	0	0	33	2	0	87	3	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					2																		
428	26	9	36	1	0	25	0	0	44	2	0	0	0	0	4	0	0	116	3	0	0	0	0
419	27	9	18	1	0	52	2	0	32	0	0	12	3	0	4	0	0	116	0	0	0	0	0
410	28	9	27	1	0	11	0	0	61	2	0	24	3	0	4	0	0	116	0	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					2																		
401	29	9	36	1	0	11	0	0	49	0	0	16	3	0	24	2	0	116	0	0	0	0	0
392	30	9	18	1	0	38	2	0	37	0	0	28	3	0	24	0	0	116	0	0	0	0	0
383	31	9	27	1	0	38	0	0	25	0	0	20	3	0	44	2	0	116	0	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					2																		
A TIE HAS OCCURED WITH OPERATOR					3																		
374	32	9	36	1	0	38	0	0	23	0	0	0	0	0	35	2	0	145	3	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					3																		
365	33	9	18	1	0	65	2	0	25	0	0	0	0	0	6	0	0	174	3	0	0	0	0
356	34	9	27	1	0	24	0	0	34	2	0	12	3	0	6	0	0	174	0	0	0	0	0
347	35	9	9	1	0	51	2	0	42	0	0	24	3	0	6	0	0	174	0	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					2																		
338	36	9	18	1	0	51	0	0	30	0	0	16	3	0	26	2	0	174	0	0	0	0	0
329	37	9	27	1	0	10	0	0	59	2	0	28	3	0	26	0	0	174	0	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					2																		
320	38	9	36	1	0	10	0	0	47	0	0	20	3	0	46	2	0	174	0	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					2																		
A TIE HAS OCCURED WITH OPERATOR					3																		
311	39	9	45	1	0	10	0	0	47	0	0	0	0	0	37	2	0	203	3	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					3																		
302	40	9	27	1	0	37	2	0	47	0	0	0	0	0	8	0	0	232	3	0	0	0	0
293	41	9	9	1	0	64	2	0	35	0	0	12	3	0	8	0	0	232	0	0	0	0	0
284	42	9	18	1	0	23	0	0	64	2	0	24	3	0	8	0	0	232	0	0	0	0	0
275	43	9	27	1	0	23	0	0	52	0	0	16	3	0	28	2	0	232	0	0	0	0	0
266	44	9	9	1	0	50	2	0	40	0	0	28	3	0	28	0	0	232	0	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					2																		
257	45	9	18	1	0	50	0	0	28	0	0	20	3	0	48	2	0	232	0	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					2																		
A TIE HAS OCCURED WITH OPERATOR					3																		
248	46	9	27	1	0	50	0	0	28	0	0	0	0	0	39	2	0	461	3	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					2																		
A TIE HAS OCCURED WITH OPERATOR					3																		
239	47	9	36	1	0	9	0	0	69	2	0	0	0	0	10	0	0	290	3	0	0	0	0
230	48	9	18	1	0	36	2	0	57	0	0	12	3	0	10	0	0	290	0	0	0	0	0
221	49	9	27	1	0	36	2*	0	45	0	0	24	3	0	10	0	0	290	0	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					2																		
212	50	9	36	1	0	36	0	0	33	0	0	16	3	0	39	2	0	290	0	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					3																		
203	51	9	18	1	0	63	2	0	33	0	0	16	0	0	1	0	0	319	3	0	0	0	0
194	52	9	27	1	0	22	0	0	62	2	0	28	3	0	1	0	0	319	0	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					2																		
185	53	9	36	1	0	22	0	0	50	0	0	20	3	0	21	2	0	319	0	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					2																		
176	54	9	45	1	0	22	0	0	38	0	0	12	3	0	41	2	0	319	0	0	0	0	0
A TIE HAS OCCURED WITH OPERATOR					3																		
167																							

APPENDIX B (CONTD.)

149	37	9 45 1	0 8 0	0 55 0	0 16 3	0 32 2	0 348 0
A TIE HAS OCCURED WITH OPERATOR 3							
140	58	9 27 1	0 35 2	0 55 0	0 16 0	0 3 0	0 377 3
131	59	9 9 1	0 62 2	0 43 0	0 28 3	0 3 0	0 377 0
A TIE HAS OCCURED WITH OPERATOR 2							
122	60	9 18 1	0 62 0	0 31 0	0 20 3	0 23 2	0 377 0
A TIE HAS OCCURED WITH OPERATOR 2							
113	61	9 27 1	0 62 0	0 19 0	0 12 3	0 43 2	0 377 0
A TIE HAS OCCURED WITH OPERATOR 2							
A TIE HAS OCCURED WITH OPERATOR 3							
104	62	9 36 1	0 21 0	0 60 2	0 12 0	0 14 0	0 406 3
95	63	9 18 1	0 48 2	0 48 0	0 24 3	0 14 0	0 406 0
A TIE HAS OCCURED WITH OPERATOR 2							
86	64	9 27 1	0 48 0	0 36 0	0 16 3	0 34 2	0 406 0
A TIE HAS OCCURED WITH OPERATOR 2							
A TIE HAS OCCURED WITH OPERATOR 3							
77	65	9 36 1	0 7 0	0 77 2	0 16 0	0 5 0	0 435 3
68	66	9 18 1	0 34 2	0 65 0	0 28 3	0 5 0	0 435 0
59	67	9 27 1	0 34 0	0 53 0	0 20 3	0 25 2	0 435 0
A TIE HAS OCCURED WITH OPERATOR 2							
50	68	9 36 1	0 34 0	0 41 0	0 12 3	0 45 2	0 435 0
A TIE HAS OCCURED WITH OPERATOR 3							
41	69	9 18 1	0 61 2	0 41 0	0 12 0	0 16 0	0 464 3
32	70	9 27 1	0 20 0	0 70 2	0 24 3	0 16 0	0 464 0
A TIE HAS OCCURED WITH OPERATOR 2							
23	71	9 36 1	0 20 0	0 58 0	0 16 3	0 36 2	0 464 0
A TIE HAS OCCURED WITH OPERATOR 3							
14	72	9 18 1	0 47 2	0 58 0	0 16 0	0 7 0	0 493 3
3	73	9 27 1	0 6 0	0 87 2	0 28 3	0 7 0	0 493 0
A TIE HAS OCCURED WITH OPERATOR 2							
0	74	9 36 1	0 6 0	0 75 0	0 20 3	0 27 2	0 493 0
A TIE HAS OCCURED WITH OPERATOR 2							
0	75	9 41 1	0 6 0	0 63 0	0 12 3	0 47 2	0 493 0
A TIE HAS OCCURED WITH OPERATOR 3							
0	76	9 14 1*	0 33 2	0 63 0	0 12 0	0 18 0	0 522 3
0	77	9 14 1*	0 33 2*	0 51 0	0 24 3	0 18 0	0 522 0
0	78	9 14 1*	0 33 0	0 39 0	0 16 3	0 38 2	0 522 0
A TIE HAS OCCURED WITH OPERATOR 3							
0	79	9 14 1*	0 33 0	0 39 0	0 16 0	0 9 2*	0 551 3
0	80	9 14 1*	0 33 0	0 27 0	0 28 3	0 9 2*	0 551 0
0	81	9 14 1*	0 33 0	0 15 0	0 20 3	0 29 2	0 551 0
A TIE HAS OCCURED WITH OPERATOR 3							
0	82	9 14 1*	0 33 0	0 15 0	0 0 0	0 20 2	0 580 3
0	83	9 14 1*	0 33 0	0 3 0	0 12 3	0 20 2*	0 580 0
0	84	9 0 1*	0 47 2	0 0 0	0 13 3	0 20 0	0 580 0
0	85	9 0 1*	0 6 0	0 41 2	0 15 3*	0 20 0	0 580 0
0	86	9 0 1*	0 6 0	0 29 2*	0 27 3	0 20 0	0 580 0
0	87	9 0 1*	0 6 0	0 17 0	0 19 3	0 40 2	0 580 0
A TIE HAS OCCURED WITH OPERATOR 3							
0	88	9 0 1*	0 6 0	0 17 0	0 19 0	0 11 2*	0 609 3
0	89	9 0 1*	0 6 0	0 5 0	0 31 3	0 11 2*	0 609 0
0	90	9 0 1*	0 6 0	0 5 0	0 11 3*	0 31 2	0 609 0
0	91	9 0 1*	0 6 0	0 5 0	0 11 0	0 2 2*	0 638 3
0	92	9 0 1*	0 0 0	0 6 2	0 16 3	0 2 0	0 638 0
0	93	9 0 1*	0 0 0	0 0 0	0 6 3	0 18 2	0 638 0
0	94	9 0 1*	0 0 0	0 0 0	0 0 0	0 6 2	0 638 3
0	95	9 0 1*	0 0 0	0 0 0	0 0 0	0 0 2*	0 662 3

TOTAL TIMES AT EACH MACHINE

74 25 17 57 34 26

TOTAL IDLE TIMES AT EACH MACHINE

20 2 3 5 8 0