Comput. & Indus. Engng Vol. 6, No. 2, pp. 131–148, 1982
Printed in Great Britain.

0360–8352/82/020131–18$03.00/0
Pergamon Press Ltd.

A NEW METHOD FOR ASSEMBLY LINE BALANCING HAVING STOCHASTIC WORK ELEMENTS

A. RAOUF and C. L. TSUI
Department of Industrial Engineering, University of Windsor, Windsor, Ontario, Canada N9B 3P4

(*Received for publication* 13 *October* 1981)

Abstract—The main feature of the method suggested in this paper is the assignment of priority to elements and priority elements are preferred to non-priority elements when assigning elements to stations. It accepts element times from known or unknown symmetrical distributions, minimizes the variation within a station and allows assignments of elements to stations such that each station time does not exceed the probability confidence levels set by management.

INTRODUCTION

Assembly line balancing refers to the procedure of assigning tasks to work stations in such a manner as to approtion the assembly work among the stations as evenly and compactly as possible without violating any precedence requirements. The assignment of elements to work stations is subject to the following conditions:

(1) Each element task is assigned to one and only one work station.

(2) The sum of the time of all elemental tasks within a work station does not exceed the cycle time.

(3) The stations thus formed can be ordered so that the partial orderings among elemental tasks are not violated.

Considerable effort has been devoted to the assembly line balancing problem in order to obtain a better balance in terms of minimum number of stations and minimum total idle time. Early work in assembly line balancing by Bryton[8], Salveson[37], Bowman[5], Jackson[18], Held *et al.*[14], Mitchell[27], Burgeson and Daun[9], Tonge[39, 40], Arcus[2, 3], Kilbridge and Wester[19, 20], Hoffman[16], Helgeson and Birnie[15], Klein[22], Gutjahr and Nemhausher[13] and quite recently work by Bennett *et al.*[4] and Raouf *et al.*[34], considered deterministic element times only. However, production lines are manned by human beings who seldom, if ever, maintain exactly the same working pace throughout the work period. Besides, no two individuals have the same working pace. Thus, the human variability complicates the system and converts the assembly line balancing problem from a deterministic into a stochastic problem with which production engineers and shop floor management must cope. Presently, methods developed by Moodie and Young[28], Mansoor[25], Brennecke[7], Ransing and Downing[32], Rao[33], Reeve[34, 35], Mansoor and Tuvia[26], and Dar-El[10, 11] solving the assembly line balancing problem under variable element times assume that element time is normally distributed. Kottas and Lau[23, 24] developed a new heuristic method based on cost consideration for balancing a paced assembly line which exhibits normal element time variations. There is ample evidence in the literature that element time values are not necessarily normally distributed. Sphicas and Silverman[38] showed that for certain classes of distribution of element times, the stochastic formulation is equivalent to the deterministic approach. A review of early work on assembly line balancing problems is given by Ingall[17] and Kilbridge and Wester[21]. From the existing techniques, it is observed that none of the researchers seem to consider minimizing the variations in station times as the elements are assigned to the work stations.

A new heuristic method which is based on Omar's[30] approach, has been developed. The new method is likely to minimize the variations in station times and accept element times which come from any symmetrical distribution.

COEFFICIENT OF VARIATION

The coefficient of variation (CV) of a distribution is defined as the ratio of the standard deviation to the mean. Assuming no correlation between elements, Brady and Drury[6] showed that "if a new element is added to a group of elements, the coefficient of variation of the new group will be less than that of the old group, provided that the CV^2 of the new element is less than $(1 + 2*r)$ times the CV^2 of the old group, where "r" is the ratio of the mean of the old group to the mean of the new element".

By incorporating this concept into the assembly line balancing method, it is likely that the variation of station times will be reduced. From a list of available elements for assignment, a set of elements is chosen which can be placed into the current station provided that its inclusion will not cause the station time to exceed the cycle time and that the probability that the station time will exceed the cycle time is within a predetermined value. From this set of elements, a subset of elements is chosen such that its inclusion into the current station will satisfy Brady and Drury's theorem. The one with the lowest CV among the elements in the subset is chosen to be added to the current station. This criterion is bypassed when the subset is empty. In this way, it is assured that the addition of an element to the work station will decrease the variation in station time. Thus, the probability that the station time exceeds the cycle time is reduced. As a result, the overall performance of the assembly line will likely be improved.

CHEBYSHEV'S INEQUALITY

Netter[29] anD Dudley[12] showed that the element time values are not necessarily normally distributed. Since there is no evidence that elemental time is precisely normally distributed, the normality assumption on element times and station times can be relaxed by using Chevyshev's inequality:

If X is a random variable with mean μ and variance σ^2, then for any positive number K the following inequality, known as Chebyshev's inequality holds.

$$P(|x - \mu| \geq K\sigma) \leq 1/K^2. \tag{1}$$

By assuming independence between element times and considering station time (ST) as a random variable having a symmetric distribution, with mean C, according to Chebyskev's inequality, we have:

$$P(|ST - C| \geq K\sigma_{ST}) \leq 1/K^2 \tag{2}$$

where C is the cycle time and σ_{ST} is the standard deviation of the station time. We are only interested in the case when $(ST - C) \geq 0$. Thus, the last inequality can be written as

$$P[(ST - C) \geq K_{\sigma ST}] \leq 1/2K^2. \tag{3}$$

The above inequality simply means that the probability of $(ST - C)$ exceeds $K_{\sigma ST}$ is always less than or equal to $1/2K^2$. If the mean and standard deviation of ST are known, it is possible to estimate the probability that each of the operators will be able to accomplish the assigned task within a predetermined C value.

If we are given a certain probability confidence level, say P_c, that represents the probability a ST can exceed the given C time, we then can calculate the value for K from the following expression:

$$K = \sqrt{(1/2P_c)}. \tag{4}$$

Assuming no correlation between element times, we can find the station time for station $j(ST_j)$ at $P_c = 1/2K^2$ from:

$$ST_j = \sum_{i \in j} U_i + K\sqrt{\left(\sum_{i \in j} \sigma_i^2\right)} \tag{5}$$

where U_i is the mean and σ_i^2 is the variance of element time for element i in station j.

THE HEURISTIC METHOD

Omar's heuristic method[30] is simple to use and provides good results for deterministic element times for almost all problems mentioned in assembly line balancing literature. We shall incorporate the proposed ideas into Omar's method in handling stochastic element times. A brief description of the method for deterministic element time values is given. A complete description of this method and a computer program has been provided in Ref. [34]. A modification of the method in dealing with stochastic time values will then be given in the next section.

It is assumed that a cycle time, uncorrelated element times and a precedence diagram are given. The heuristic method consists of two phases:

Phase I is the determination of the critical path and the assignment or priority to the work elements. The steps employed in phase I are as follows:

(1) Obtain the critical path of the given problems for each end element that requires at least two stages before its completion.

(2) Assign priority to the critical elements which fall in the critical path.

(3) Assign priority to those elements that have immediate successors and the cumulative time values sum of the immediate successors is greater than or equal to the cycle time.

(4) Assign priority to those elements which are direct predecessors of a critical element and immediate successors of another element and whose cumulative sum is greater than the cycle time.

(5) Assign priority to those elements which are direct predecessors of critical elements and have the lowest elemental time value among them.

Phase II is the assignment of elements to the work stations on the assembly line. The assignment of elements is done by the "largest candidate priority rule" which entails considering first the priority elements in declining order of the element time values and then the non-priority elements in the same manner for the purpose of assigning them to the work stations.

For treating the element times as stochastic variables, the coefficient of variation (*CV*) and Chebyshev's inequality are incorporated into the heuristic method as outlined above. In addition to the cycle times, mean element times and the precedence diagram, elemental time variance and the maximum probability confidence levels are assumed to be given.

Phase I of the new method will be the same as in the deterministic case.

In Phase II, the elements are assigned to consecutive work stations along the assembly line by the "least *CV* priority rule". This entails considering first the priority elements which cause the least increase in *CV* of the station and the non-priority elements which follow in the same manner.

The algorithm employed in phase II is as follows:

(1) From the *P*-matrix†, all elements whose rows contain all zeros are noted. These are called "available elements".

(2) From the available elements, the priority and the non-priority elements are placed in the priority list and the non-priority list respectively.

(3) From the priority list obtained in step 2, a set of priority elements is formed. Each element in the set satisfied the following two conditions: (a) its inclusion into the station will not cause the station time to exceed the cycle time and, (b) the probability that the resulting station time exceeds the cycle time is less than or equal to a predetermined value.

(4) A subset of elements from the set obtained in step 3 is formed, consisting of elements that satisfy Brady and Drury's theorem[6].

(5) If this subset is empty, steps 3 and 4 are repeated for elements in the non-priority list. Otherwise, from this subset the ones which cause the least increase in CV of the station are assigned to the current station. Brady and Drury's theorem is by passed when the subset of non-priority element is also empty.

In this case, the choice of elements assigned to station is based upon the ones which cause the least increases in CV of the station, with preference for the priority elements.

†This shows the immediate predecessors of each element (see Fig. 1).

(6) Elements in the row of the assigned element in the F-matrix† are noted. In the corresponding rows in the P-matirx indicated by the elements just noted, the assigned element is replaced by a zero. If the assignment of a zero causes the row in the P-matrix to become all zeros, the corresponding element becomes an available element.

(7) The elements are assigned according to steps 2–6 until the priority list and the non-priority list are empty. When this is the case, the problem is solved.

AN ILLUSTRATED PROBLEM

A 14-element problem (1) from industry is chosen to illustrate the new method. Precedence diagram, P-matrix and F-matrix are shown in Fig. 1. Mean element times and element variances are shown in Table 2.

For this problem Omar's method does not seem to produce good results in deterministic element times because there are many starting elements to choose. Conduct phase I in the backward direction (right to left) for the precedence diagram and phase II in the forward direction (left to right), a better result is obtained. A comparison of the results for cycle time 39 sec is shown in Table 3.

Table 1. Dual precedence matrices for 14-element problem

Element No.	P-Matrix		F-Matrix	
1	0	0	0	0
2	1	0	6	0
3	0	0	5	0
4	0	0	5	0
5	3	4	14	0
6	2	0	9	0
7	0	0	8	0
8	7	0	9	0
9	6	8	10	0
10	9	0	13	0
11	0	0	12	0
12	11	0	13	0
13	10	12	14	0
14	5	13	0	0

Table 2. Mean and variance of element time for 14-element problem

Element No.	Element Time	
	Mean	Variance
1	7	0.36
2	25	19.36
3	12	15.21
4	6	2.89
5	11	1.69
6	13	4.84
7	16	5.29
8	11	6.25
9	6	8.41
10	12	1.69
11	19	5.29
12	13	6.79
13	9	3.61
14	13	28.09

Table 3. Comparison of proposed backward assignment method with other methods for cycle time 39

	Omar (30)	New Method	Moodie & Young(28)	Hoffman (16)
# of stations	6	5	5	5
Total idle time	61	22	22	22
Smoothness index	31.54	9.64	17.32	17.32
Efficiency	74 %	88.7%	88.7%	88.7%

†This shows the immediate followers of each element (see Table 1).

We shall apply the improved method to the 14-element problem for stochastic element time. Further, we assume that the cycle time is fixed at 39 sec abd that management wishes that the maximum probability for each station to exceed the cycle time to be not more than 5%. From eqn (4) we can calculate the value of K.

$$K = \sqrt{(1/0.05)} = 3.162.$$

In phase I, all fourteen elements are priority elements. In phase II, element 1, 3, 4, 7 and 11 are the available elements. To begin with, the one with the least CV^2 is chosen to be assigned to work station 1. From Table 3, element 1 is assigned to station 1 and the station time is increased to 8.14 sec. From the P-matrix, and the F-matrix, element 2 is added to the available element list.

Only elements 3, 4, 7 and 11 are capable of being assigned to station 1 because the inclusion of element 2 in station 1 will cause the station time to exceed the given cycle time. The Brady and Drury theorem is bypassed since none of the elements 3, 4, 7 and 11 satisfied its' hypothesis (Table 4). Element 11 is assigned to station 1 because it causes the least increase in CV of station 1. The station time now is increased to 33.52 sec. Element 12 is added to the available list. No further elements can be added to station 1 since its inclusion will cause the station time to exceed the cycle time of 39 sec. Now a new station is opened for assignment. Continuing the same procedure, the final results of assignment are shown in Table 6.

Table 4. CV^2 of available elements

Element	CV^2
1	0.0073
3	0.1056
4	0.0803
7	0.0207
11	0.0147

Table 5. Resultant station CV^2 with element i included

Element i	Element CV^2	(1+2r) Station CV^2
3	0.0156	0.0160
4	0.0803	0.0245
7	0.0207	0.0138
11	0.0147	0.0128

r = mean of station/mean of new element

Table 6. Balance for the 14-element problem for cycle time 39 sec

Work Station	Elements	Station Time (ST)	Idle Time	P(ST > CT)
1	1,11	33.52	5.48	0.017
2	7,8	37.74	1.26	0.04
3	2	38.91	0.09	0.049
4	6,12	36.91	2.22	0.034
5	4,3	31.45	7.55	0.021
6	5,9	27.05	11.95	0.010
7	10,13	28.28	10.72	0.008
8	14	29.76	9.24	0.027

Smoothness index = 20.70 output efficiency = 84.5%

ZONING CONSTRAINTS

Besides the precedence requirements imposed upon the grouping of elements into work stations, zoning constraints are common in assembly lines because of the layout of production facilities. An example is the non-assignment of tasks to be done on the front and the back of a product to an operator where those tasks require him to cross the conveyor line.

Modifications are made in the heuristic method to handle zoning constraints as well as deterministic and variable element times values. Instead of placing the available elements into priority and non-priority lists, we now place them in one of the following four lists:

(1) Priority zone list that contains elements that can only be placed in the current zone.

(2) Non-priority zone list that contains elements that can only be placed in the current zone.

(3) Priority list that contains elements that can be placed in the current zone or some other zones.

(4) Non-priority list that contains elements that can be placed in the current zone or some other zones.

Priority zone list has the highest priority and the non-priority list has the lowest priority. The 21-element problem from Tonge[40] with zoning constraints (Table 7) is solved for varriable element time values and the balance is given in Table 8. Many researchers did mention the zoning constraints but they did not solve the assembly line balancing zone constraints problem using their proposed methods especially for variable element times.

RESULTS

The new heuristic method for assembly line balancing is programmed in FORTRAN language. A flow chart which describes details of the procedure is depicted in Appendix A. The program listing is given in Appendix B. This program can be used for cases where work elements are assumed to be deterministic or stochastic in nature.

Three sample problems are used in testing the new heuristic method. They are:

(1) 14-element problem from industry[1]

(2) 21-element problem from Moodie and Young[28]

(3) 11-element problem from Kottas and Lau[23].

A summary of the solutions under stochastic element times at various probability confidence levels for the three chosen problems and comparisons with other methods are given in Table 8. Further, a summary of the comparisons between the CV^2 of each station obtained by the new heuristic method and the three chosen problems obtained by using other methods are given in Table 9.

Table 7. Zoning constraints for the 21-element from Tonge[40]

Zone	Elements
1	1, 2, 3, 4, 5, 7, 8, 21
2	3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 18
3	13, 14, 15, 16, 17, 18, 19, 20

Table 8. Balance for the 21-element problem for cycle time 20 under variable element time with zoning constraints with 95% probability confidence level

Work Station	Elements	Station Time (ST)	Idle Time	P(ST>CT)
1	1,2	11.47	8.53	0.006
2	3	13.47	6.53	0.008
3	4,21	18.32	1.68	0.031
4	5	14.48	5.52	0.012
5	7,6	18.32	1.68	0.031
6	8,9	19.07	0.93	0.039
7	11,10,12,15,16	20.07	0.00	0.051
8	13,14,18	18.48	1.52	0.031
9	17	20.07	0.00	0.051
10	20,19	9.47	10.53	0.004

Smoothness index = 16.27 Output efficiency = 81.55%

Table 9. Comparison of various methods under variable element times for different cycle times and probability confidence levels

Pc	# of elements	Method	Cycle Time	# of Stat	Idle Time	S.I.	Eff. %
95%	14	(1)	39	8	47.51	20.13	84.77
		(2)		8	48.52	20.7	84.46
		(3)		8	49.40	20.0	84.17
95%	14	(1)	40	7	22.03	11.5	92.13
		(2)		7	23.3	13.3	91.68
		(3)		8	57.41	20.0	82.0
90%	14	(1)	35	7	12.65	7.0	93.24
		(2)		7	13.9	8.84	94.32
		(3)		8	43.68	17.41	84.40
85%	14	(1)	34	7	16.56	7.0	93.04
		(2)		7	17.16	7.86	92.79
		(3)		8	49.21	18.3	82.0
95%	21	(1)	21	9	30.0	16.53	84.13
		(2)		9	29.0	16.73	84.64
		(3)		10	47.2	20.91	77.52
90%	11	(1)	10	7	14.3	6.17	82.1
		(2)		7	12.63	6.79	81.96
		(3)		7	13.48	7.47	80.74

S.I. Smoothness index
Eff. Output efficiency
Eff = {1-Idle time/(Cycle time * Station Time)}*100%
(1) Moodie & Young method (28)
(2) Proposed method
(3) Helgeson and Birnie method (15)
(4) Proposed method under normality assumption
(5) Kottas & Lau (23) under normality

Table 10. Comparison of CV^2 of various methods at different probability confidence levels

Work Station	$\frac{CV_1^2}{CV_2^2}$ of the 14-element			$\frac{CV_1^2}{CV_2^2}$ of the 21-element	$\frac{CV_3^2}{CV_4^2}$ of the 11-element
	95%	90%	85%	95%	90%
1	5.13	1.0	1.0	1.0	1.0
2	1.0	1.0	1.0	1.0	1.0
3	0.82	1.0	1.0	1.0	0.76
4	0.71	1.0	1.0	0.91	1.0
5	0.27	1.58	1.58	0.95	1.40
6	1.45	1.16	1.16	1.33	1.0
7	1.0	1.0	1.0	1.27	1.0
8	1.0			1.0	
9				0.66	

CV_1^2 : CV^2 from Moodie and Young (28).
CV_2^2 : CV^2 from the new method.
CV_3^2 : CV^2 from Kottas and Lau (23).

In Table 8, we see that the proposed method is far more superior than the Helgeson and Birdnie RPW method and it gives almost similar results when compared to Moodie and Young's method. From Table 9, the proposed method minimizes the variability of the station for better than any of the methods. Moreover, results from the new heuristic method are more realistic when we take into consideration station variability and all symmetrical distributions of work elements. The new heuristic method provides the maximum allowable balance to the assembly line balancing problem. When the distribution of work element is known, we can simply replace Chebyshevyshev's inequality by the limits imposed by that distribution.

CONCLUSIONS

The incorporation of the coefficient of variation and Chevyshev's inequality into the critical path heuristic method [30] for the assembly line balancing problem do provide good and realistic results. The new heuristic method has the following characteristics:

(a) It can handle variable element time values.

(b) It can handle zoning constraints.

(c) In the case of stochastic assembly line balancing; (1) It accepts element time from known or unknown symmetrical distribution. (2) It minimizes the variation within a station. (3) It allows assignment of elements to station such that each station time does not exceed the probability confidence levels set by management.

REFERENCES

1. B. Amin, *An Algorithm to Solve Stochastic Line Balancing Problem Using Moodie and Young's and Chevyshev's Inequality*, Research paper, Dept. of Industrial Engineering, Rutgers University, New Jersey, 1979.
2. A. L. Arcus, Assembly line balancing by computer. Graduate School of Business, University of California, Berkley, June 1962.
3. A. L. Arcus, COMOSAL—A computer method of sequencing operations for assembly lines. *Int. J. Production Res.* **4**(4), 259–277 (1966).
4. G. R. Bennette & Jack Byrd, Jr., A trainable heuriestic procedure for the assembly line balancing problem. *AIIE Trans.* **8**(2), 195–201 (1976).
5. E. H. Bowman, Assembly line balancing by linear programming. *Opers. Res.* **8**(3), 385–9, (1960).
6. W. Brady & C. G. Drury, The dependence of the coefficient of variation of a work time distribution of number of elements in the work task. *Int. J. Production Res.* **7**(4), 311, 1 (1969).
7. Brennecke, D., Two parameter assembly line balancing model. *Models and Analysis for Production Management* (Edited by M. Hottenstein). International Textbook Company, Scranton, Pennsylvania (1968).
8. B. Bryton, Balancing of a continuous production line. Unpublished M.S. Thesis, Northwestern University (1954).
9. J. W. Burgeson and T. E. Daum, Production line balancing. File No. 10.3.002, Akron, International Business Machines Corporation, Ohio, 1958.
10. E. M. Dar-El, MALB—A heuristic technique for balancing large single-model assembly lines. *AIIE Trans* **5**(5), 343–356 (1973).
11. E. M. Dar-El, Solving large single-model assembly line balancing problems—A comparative study. *AIIE Trans.* **7**(3), 302–310 (1975).
12. N. A. Dudley, *Work Measurement, Some Research Studies*, p. 43. MacMillan, London (1968).
13. A. Gutjahr & G. Nemhauser, An algorithm for the line balancing problem. *Management Sci.* **11**(2), 308–333 (1964).
14. M. Held, P. M. Karp & R. Shareshian, Assembly line balancing—dynamic programming with precedence constraints. *Opers Res.* **11**(3), 442–459 (1963).
15. W. P. Helgeson & D. P., Birnie, Assembly line balancing using the ranked positional weight technique. *J. Indus. Engng* **12**(6), 394–398 (1961).
16. T. R. Hoffman, Assembly line balancing with a precedence matrix. *Management Sci.* **9**(4), 551–563 (1963).
17. E. J. Ingall, A review of assembly line balancing. *J. Indus. Engng* **16**(4), 244–254 (1965).
18. J. R. Jackson, A computing procedure for a line balancing problem. *Management Sci.* **2**(3), 261–267 (1956).
19. M. D. Kilbridge & L. Wester, A heuristic method of assembly line balancing. *J. Indus. Engng* **12**(4), 292–298 (1961).
20. M. D. Kilbridge & L. Wester, Heuristic line balancing—A case. *J. Indus. Engng* **13**(3), 139–149 (1962).
21. M. D. Kilbridge & L. Wester, Review of analytical systems of line balancing. *Opers Res.* **10**(5), 626–638, (1962).
22. M. Klein, On assembly line balancing. *Opers Res.* **11**(2), 274–281 (1963).
23. J. F. Kottas & H. S. Lau, A cost-oriented approach to stochastic line balancing. *AIIE Trans.* **5**(2), 164–171 (1973).
24. J. F. Kottas & H. S. Lau, A total operating cost model for paced lines with stochastic tasks times. *AIIE Trans.* **8**(2), 234–240 (1976).
25. E. M. Mansoor, Assembly line balancing—An improvement on the ranked positional weight technique. *J. Indus. Engng* **15**(2), 73–77 (1964).
26. E. M. Mansoor & B. S. Tuvia, Optimizing balanced assembly lines. *J. Indus. Engng* **18**(3), 1966.
27. J. Metchell, A computational procedure for balancing zoned assembly lines. *Res. Reps. No.* 6–94801–R3, Pittsburgh: Westinghouse Research Laboratories, Feb. 1957.
28. C. L. Moodie & H. H. Young, A heuristic method of assembly line balancing for assumptions of constant or variable work element times. *J. Indus. Engng* **16** (1), 23–29, (1965).
29. M. A. Netter, Critical path analysis of repetitive man machine systems operations. Ph.D. Dissertation, University of Michigan, Ann Arbor, Michigan, 1969.
30. M. T. Omar, Development of a heuristic method for assembly line balancing, Unpublished M.A.Sc. Thesis, University of Windsor, 1975.
31. A. Papoulis, *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, New York (1965).
32. K. Ramsing & R. Downing, Assembly line balancing with variable element time. *Indus. Engng* 41–43(1970).
33. D. Rao, Single and mixed model assembly line balancing methods for both deterministic and normally distributed work element times. Unpublished M.S. Thesis, Oregon State University, June 1971.
34. A. Raouf, E. A. El-Sayed & C. L. Tsui, A new heuristic approach to assembly line balancing. *Comput. Indus Engng* **4**, 223–234 (1980).
35. N. R. Reeve, A heuristic technique for assembly line balancing, Unpublished M.S. Thesis, State University of New York at Buffalo, 1968.
36. N. R. Reeve, Balancing continuous stochastic assembly lines. Unpublished Ph.D. Thesis, State University of New York at Buffalo, 1971.
37. N. R. Reeve & W. H. Thomas, Balancing stochastic assembly lines. *AIIE Trans.* **5**(3), 223–229 (1973).
38. M. E. Salveson, The assembly line balancing problem. *J. Indus. Engng* **6**(3), 18–25 (1955).

39. G. P. Sphicas & F. N. Silverman, Deterministic equivalents for stochastic assembly line balancing. *AIIE Trans.* **8**(2), 280–282 (1976).
40. F. M. Tonge, Summary of a heuristic line balancing procedure. *Management Sci.* **7**(1), 21–42 (1960).
41. F. M. Tonge, *A heuristic program for assembly line balancing.* Prentice Hall, Englewood Cliffs, New Jersey (1961).
42. F. M. Tonge, Using probabilities combinations of heuristics. *Management Sci.* **11**, 727–735 (1965).
43. C. L. Tsui, Stochastic assembly line balancing using the coefficient of variation and Chebyshev's inequality, M.A.Sc. Major Paper, Dept. of Industrial Engineering, University of Windsor, Windsor, Ontario, 1980.

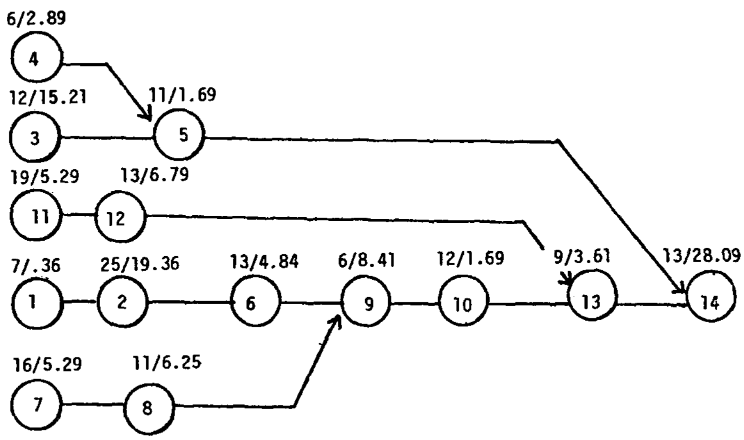
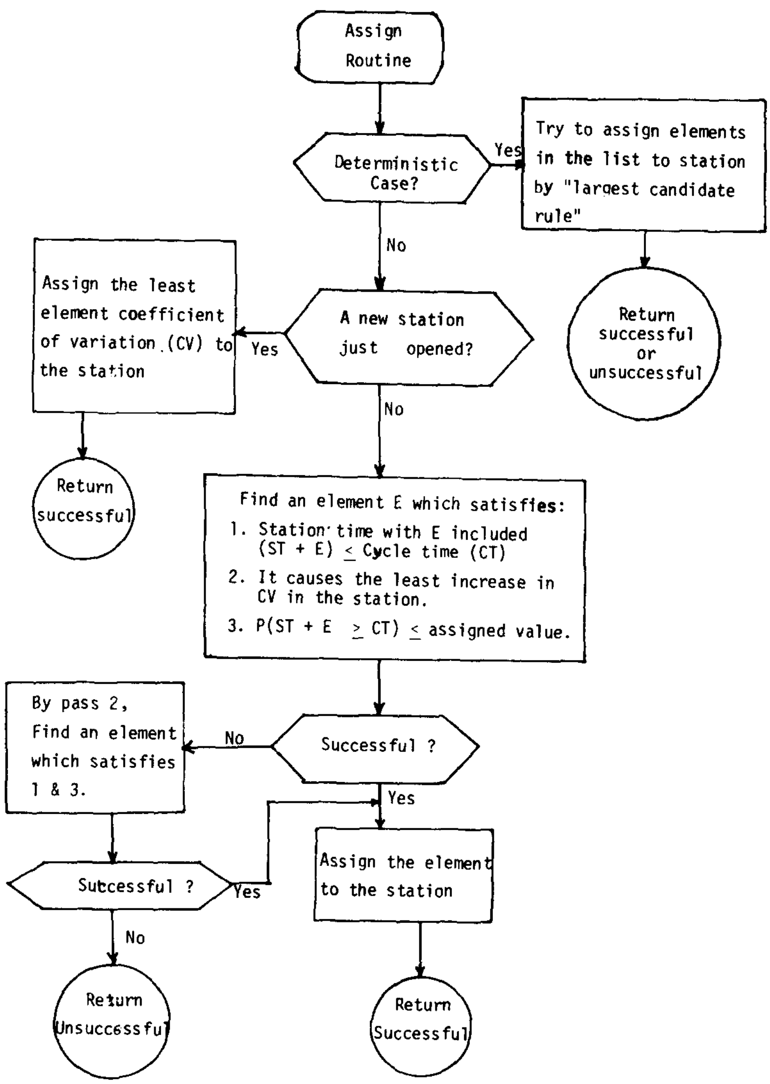
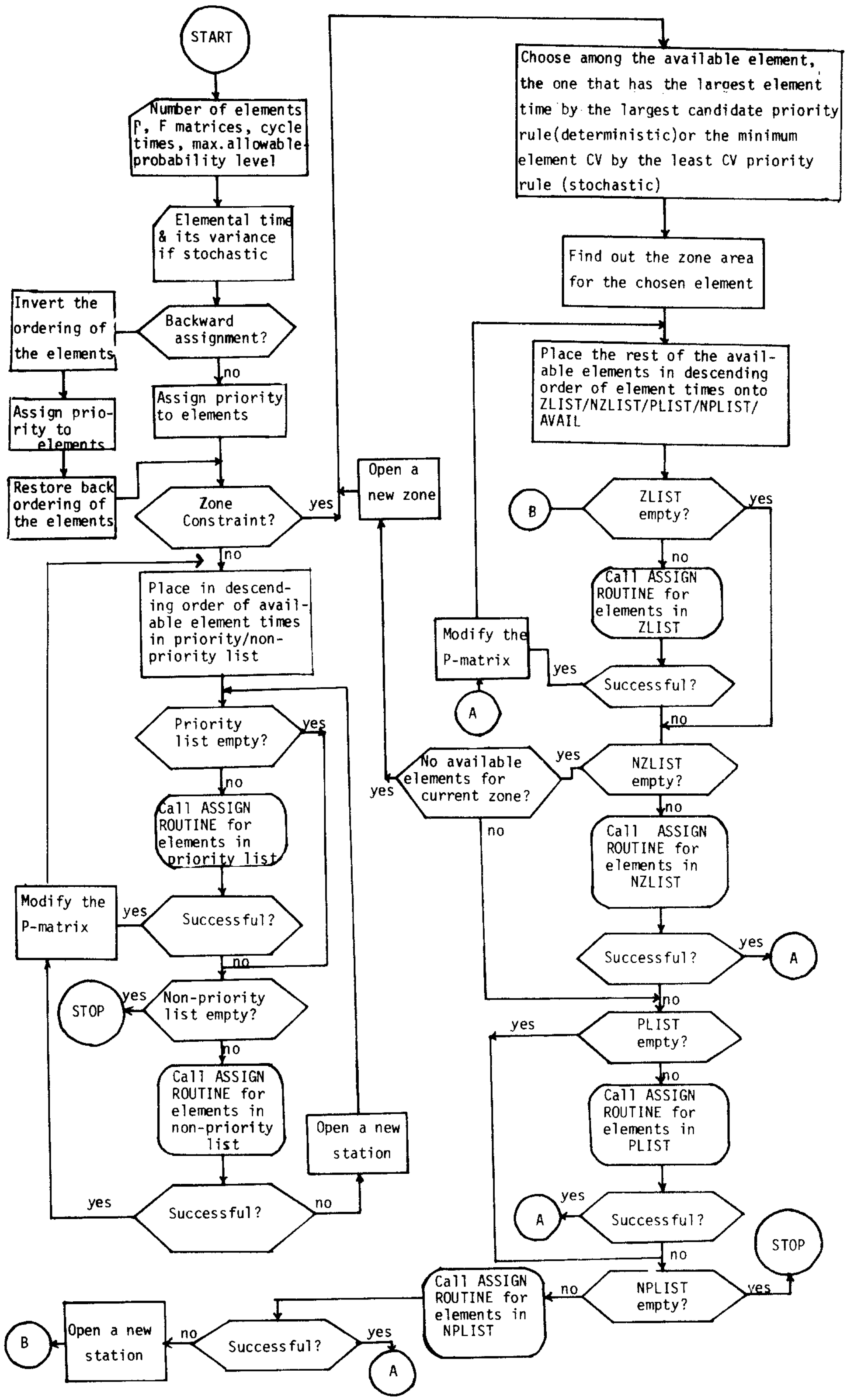


Fig. 1. Precedence diagram for 14 element problem.

APPENDIX A: FLOW CHART FOR ASSIGN ROUTINE




START
Number of elements P, F matrices, cycle times, max.allowable probability level
Elemental time & its variance if stochastic
Backward assignment?
Invert the ordering of the elements
Assign priority to elements
Assign priority to elements
Restore back ordering of the elements
Zone Constraint?
Open a new zone
Place in descending order of available element times in priority/non-priority list
Priority list empty?
Call ASSIGN ROUTINE for elements in priority list
Successful?
Modify the P-matrix
Non-priority list empty?
STOP
Call ASSIGN ROUTINE for elements in non-priority list
Successful?
Open a new station
Choose among the available element, the one that has the largest element time by the largest candidate priority rule(deterministic)or the minimum element CV by the least CV priority rule (stochastic)
Find out the zone area for the chosen element
Place the rest of the available elements in descending order of element times onto ZLIST/NZLIST/PLIST/NPLIST/AVAIL
ZLIST empty?
B
Call ASSIGN ROUTINE for elements in ZLIST
Successful?
Modify the P-matrix
A
NZLIST empty?
No available elements for current zone?
Call ASSIGN ROUTINE for elements in NZLIST
Successful?
A
PLIST empty?
Call ASSIGN ROUTINE for elements in PLIST
Successful?
A
STOP
NPLIST empty?
Call ASSIGN ROUTINE for elements in NPLIST
Successful?
A
Open a new station
B
yes
no

APPENDIX B:

```
C *****************************************************************************
C *   CRITICAL PATH APPROACH TO ASSEMBLY LINE BALANCING                       *
C *                     ZONE CONSTRAINTS              (IZ    = 1)             *
C *                                                                           *
C *   THIS PROGRAM IS WRITTEN IN FORTRAN AND CONSISTS OF TWO PHASES           *
C *                                                                           *
C *      PHASE 1-----THE DETERMINATION OF THE CRITICAL PATH FOR               *
C *                  EACH ELEMENT  WHICH HAS AT LEAST TWO ELEMENTS            *
C *                  BEFORE IT ALONG THE PATH & ASSIGNMENT OF PRIORITY        *
C *                  TO ELEMENTS                                              *
C *      PHASE 2-----ASSIGNING THE ELEMENTS TO STATIONS                       *
C *                                                                           *
C *   DEFINING THE VARIABLES USED IN THE PROGRAM                              *
C *     CT-----------CYCLE TIME                                               *
C *     NOE----------NO. OF ELEMENTS IN THE PROBLEM                           *
C *     NOCP---------NO. OF COLUMNS IN P MATRIX                               *
C *     NOCF---------NO. OF COLUMNS IN F MATRIX                               *
C *     IZ-----------INCORPORATES ZONE CONSTRAINTS (VALUE 1)                  *
C *                  OR NO CONSTRAINTS (VALUE 0)                              *
C *     NZ-----------NO. OF ZONE AREA  UNDER CONSIDERATION                    *
C *     IB-----------BACKWARD (VALUE 1) OR FORWARD (VALUE 0) ASSIGNMENT       *
C *                  OF PRIORITY FOR THE ELEMENTS                             *
C *     ALLOW--------MAXIMUM ALLOWABLE PROBABILITY THAT THE STATION           *
C *                  WILL NOT EXCEED THE CYCLE TIME                           *
C *     TIME(M)------ELEMENTAL TIME FOR ELEMENT M                             *
C *     VAR(M)-------TIME VARIANCE FOR ELEMENT M                              *
C *     CV2(M)-------SQUARE OF COEFFICIENT OF VARIATION FOR ELEMENT M         *
C *     SLACK(M)-----SLACK TIME FOR ELEMENT M                                 *
C *     CDP(M)-------CUMULATIVE TIME OF DIRECT PREDECESSORS FOR               *
C *                  ELEMENT M                                                *
C *     CIF(M)-------CUMULATIVE TIME OF IMMEDIATE SUCCESSORS OF               *
C *                  ELEMENT M                                                *
C *     PRT(M)-------PRIORITY OF ELEMENT M (VALUE 1)                          *
C *     ZLIST--------PRIORITY LIST OF AVAILABLE ELEMENTS THAT CAN BE          *
C *                  PLACED IN CURRENT ZONE ONLY                              *
C *     NZLIST-------NON-PRIORITY LIST OF AVAILABLE ELEMENTS THAT             *
C *                  CAN BE PLACED IN CURRENT ZONE ONLY                       *
C *     NPLIST-------NON-PRIORITY LIST OF AVAILABLE ELEMENTS OR               *
C *                  AVAILABLE ELEMENTS THAT CAN BE PLACED IN CURRENT         *
C *                  ZONE OR OTHERS                                           *
C *     AVAIL--------A LINKED LIST CONTAINS AVAILABLE ELEMENTS THAT           *
C *                  CANNOT BE PLACED IN CURRENT ZONE                         *
C *     ZONE(I,J)----VALUE 1 INDICATES ELEMENT I CAN BE PLACED IN             *
C *                  ZONE J AND 0 OTHERWISE                                   *
C *     SNO----------NO. OF STATIONS                                          *
C *     NWS(I)-------NO. OF ELEMENTS IN STATION I                             *
C *     WS(I,)-------ELEMENTS IN STATION I                                    *
C *     ST(I)--------STATION TIME FOR STATION I                               *
C *     IDLE(I)------IDLE TIME FOR STATION I                                  *
C *     STCV2(I)-----SQUARE OF COEFFICIENT OF VARIATION FOR STATION I         *
C *     PROB(I)------PROBABILITY THAT STATION I WILL EXCEED THE               *
C *                  CYCLE TIME                                               *
C *     FLAG---------SUCESS(VALUE 0) OR FAILURE(VALUE 1) OF ASSIGNMENT        *
C *                  OF AVAILABLE ELEMENTS TO CURRENT STATION                 *
C *                                                                           *
C *   THIS PROGRAM CAN HANDLE UP TO A 50 ELEMENT-PROBLEM  WITH A              *
C *   MAXIMUM OF 15 DIRECT PREDECESSORS OR IMMEDIATE SUCCESSORS FOR           *
C *   EACH ELEMENT                                                            *
C *                                                                           *
C *   DATA INPUT :  (FORMATTED)                                               *
C *                                                                           *
C *     1. FIRST SET OF DATA CONTAINS :                                       *
C *           COLUMN    1- 3     NOE       ( I3 )                             *
C *                     4- 6     NOCP      ( I3 )                             *
C *                     7- 9     NOCF      ( I3 )                             *
C *                    10-14     CT        (F5.0)                             *
C *                    15        IZ        ( I1 )                             *
C *                    16-17     NZ        ( I2 )                             *
C *                    18        IB        ( I1 )                             *
C *                    20-23     ALLOW     (F4.2)                             *
C *     2. SECOND SET OF DATA CONTAINS REAL VALUES OF ELEMENT TIMES           *
C *          IN F5.0 FORMAT                                                   *
C *     3. THIRD SET OF DATA CONTAINS NOE GROUPS OF DATA  IN I3 FORMAT        *
C *          ITH GROUP CONTAINS DIRECT PREDECESSORS & IMMEDIATE               *
C *                  SUCCESSORS OF ELEMENT I (ALL IN INTEGER FORM).           *
C *                  IF THE NUMBER OF DIRECT PREDECESSORS AND /OR             *
C *                  IMMEDIATE SUCCESSORS < NOCP AND/OR NOCF, FILLED          *
C *                  THE REMAINING WITH ZEROS OR BLANKS.                      *
C *     4. FOURTH SET OF DATA CONTAINS REAL VALUES OF ELEMENT TIME            *
C *          VARIANCE IN F6.3 FORMAT.                                         *
C *     5. FIFTH SET OF DATA CONTAINS NOE GROUPS OF ZONE INFORMATION          *
C *          IN I1 FORMAT IF IZ = 1                                           *
C *****************************************************************************
      DIMENSION TIME(50),SLACK(50),CIF(50),CDP(50),NP(10),NF(15)
      DIMENSION NPLIST(50),ST(25),NWS(25),VAR(50),CV2(50)
      DIMENSION STCV2(20),PROB(25),NZLIST(50)
      INTEGER P(50,15),F(50,15),PP,FF,PRT(50),DPCLT(10),POINT
      INTEGER DPIF(10),COUNT(100),PTR,PLIST(50),WS(25,20),SNO,CP,FLAG
      INTEGER DIGIT(15)/'1','2','3','4','5','6','7','8','9',
     $                  '10','11','12','13','14','15'/
      INTEGER VF(6)/'(1X,','T11,','I3, ','T24,','    ','I7)'/
      INTEGER RVF(5)/'(   ','    ','I3, ','    ','I3)'/
      INTEGER WVF(6)/'(11X','.I2,','6X, ','    ','I10,','/)'/
      INTEGER AVAIL(25,2),ZONE(50,7),SZ(50),ZLIST(50)
      INTEGER ZPTR,APTR,PPTR,UPTR,CPTR
      REAL LST(100),IDLE(25)
      CHARACTER C1*20/'DETERMINISTIC TIMES'/
      CHARACTER C2*20/'VARIABLE TIMES'/
      CHARACTER C3*20/'BACKWARD ASSIGNMENT'/
      CHARACTER C4*20/'ZONE CONSTRAINTS'/
      COMMON TIME,WS,CT,CUMST,AVAIL,ZONE,SZ,APTR,PPTR,NZ,PRT
      COMMON CMEAN,OVAR,CVO2,VAR,ALPHA,CUMVAR,STIME,CV2
      DATA LST/100*0./,CDP/50*0./,CIF/50*0./,PRT/50*0/,VAR/50*0./
      DATA SLACK/50*10E06/,TLARGE/0./,NLE/0/,NIEP/0/,NIE/0/
C******   READ IN THE INPUT VALUES
```

```
1111  READ(5,555,END=9999)NOE,NOCP,NOCF,CT,IZ,NZ,IB,ALLOW
 555  FORMAT(3I3,F5.0,I1,I2,I1,1X,F4.2)
      READ(5,5555)(TIME(I),I=1,NOE)
5555  FORMAT(16F5.0)
      RVF(2)=DIGIT(NOCP)
      RVF(4)=DIGIT(NOCF)
      DO 1 I=1,NOE
      READ(5,RVF)(P(I,J),J=1,NOCP),(F(I,J),J=1,NOCF)
1     IF(TIME(I).GT.TLARGE)TLARGE=TIME(I)
      TLARGE=TLARGE*NOE
      ALPHA=SQRT(1./(2*ALLOW))
      READ(5,5666)(VAR(I),I=1,NOE)
5666  FORMAT(13F6.3)
      IF(IB.EQ.0)GO TO 101
      CALL INVERT(NOE,NOCP,NOCF,P,F,TIME)
C******  PHASE 1
C******  FIND OUT ELEMENTS THAT DO NOT HAVE PREDECESSORS & ASSIGN THEM
C******  WITH THEIR ELEMENTAL TIME AS THEIR LATEST FINISH TIME.
101   DO 10 I=1,NOE
      IF(P(I,1).NE.0)GO  TO 100
         DO 11 J=2,NOCP
11       IF(P(I,J).NE.0)GO TO 100
         LST(I)=TIME(I)
         NIEP=NIEP+1
         NP(NIEP)=I
 100  IF(F(I,1).NE.0)GO TO 10
         DO 12 J=2,NOCF
12       IF(F(I,J).NE.0)GO TO 10
         NIE=NIE+1
         NF(NIE)=I
  10  CONTINUE
C******  ASSIGN PRIORITY TO THOSE END ELEMENTS THAT REQUIRE AT LEAST
C******  TWO ELEMENTS BEFORE IT ALONG THE PATH
      DO 120 JF=1,NIE
      FF=NF(JF)
      DO 122 J=1,NOCP
      IF(P(FF,J).EQ.0)GO TO 122
      PP=P(FF,J)
      DO 123 JP=1,NIEP
      IF(NP(JP).NE.PP)GO TO 123
      GO TO 122
 123  CONTINUE
      PRT(FF)=1
      SLACK(FF)=0.
      GO TO 120
 122  CONTINUE
 120  CONTINUE
C******  DETERMINE THE LATEST FINISH TIME (LST), CDP & CIF FOR EACH
C******  ELEMENT
      DO 13 I=1,NOE
      DO 133 J=1,NOCP
      IF(P(I,J).EQ.0)GO TO 133
      PP=P(I,J)
      CDP(I)=CDP(I)+TIME(PP)
 133  CONTINUE
      DO 14 J=1,NOCF
      IF(F(I,J).EQ.0)GO TO 14
      FF=F(I,J)
      CIF(I)=CIF(I)+TIME(FF)
      TMAX=LST(I)+TIME(FF)
      IF(LST(FF).LT.TMAX)LST(FF)=TMAX
  14  CONTINUE
C******  ASSIGN PRIORITY TO THOSE ELEMENTS WHOSE CIF > CT
      IF(CIF(I).GE.CT)PRT(I)=1
  13  CONTINUE
      DO 171 JF=1,NIE
      FF=NF(JF)
1777  POINT=0
C******  DETERMINE THE CRITICAL ELEMENT AND ASSIGNED WITH PRIORITY
      DO 172 JJ=1,NOCP
      PP=P(FF,JJ)
      IF(PP.EQ.0)GO TO 172
      SLACK(PP)=LST(FF)-LST(PP)-TIME(FF)
      IF(SLACK(PP).NE.0.)GO TO 172
         PRT(PP)=1
         POINT=PP
 172  CONTINUE
      FF=POINT
      DO 173 JP=1,NIEP
      IF(FF.NE.NP(JP)) GO TO 173
         PRT(FF)=1
         GO TO 171
 173  CONTINUE
      GO TO 1777
 171  CONTINUE
      TL=TLARGE
      DO 18 I=1,NOE
      IF(SLACK(I).NE.0.)GO TO 18
C******  ELEMENTS WHICH ARE DIRECT PREDECESSORS OF A CRITICAL ELEMENT
C******  AND HAVE THE LEAST  ELEMENTAL TIME AMONG THOSE WHICH ARE
C******  DIRECT PREDECESSORS OF CRITICAL ELEMENTS ARE ASSIGNED WITH
C******  PRIORITY
         DO 20 J=1,NOCP
         PP=P(I,J)
         IF(PP.EQ.0)GO TO 20
         IF(TL.LT.TIME(PP))GO TO 20
               IF(TL.NE.TIME(PP))GO  TO 181
                  NLE=NLE+1
                  DPCLT(NLE)=PP
             GO TO 20
 181              TL=TIME(PP)
                  NLE=1
                  DPCLT(NLE)=PP
  20     CONTINUE
```

```
   18   CONTINUE
        IF(NLE.EQ.0)GO TO 211
        DO 21 I=1,NLE
        J=DPCLT(I)
 21     PRT(J)=1
  211   DO 22 I=1,NOE
        IF(SLACK(I).NE.0.)GO TO 22
            IF(CCP(I).LT.CT)GO TO 22
                    DO 25 K=1,NOE
 25                 COUNT(K)=0
                    DO 23 J=1,NOCP
                    PP=P(I,J)
                    IF(PP.EQ.0)GO TO 23
                    DO 29 K=1,NOE
                    IF(SLACK(K).NE.0.)GO TO 29
                    DO 24 L=1,NOCF
                    IF(F(K,L).EQ.0)GO TO 24
                    IF(F(K,L).EQ.PP)COUNT(K)=COUNT(K)+1
   24               CONTINUE
   29               CONTINUE
   23               CONTINUE
                    DO 26 K=1,NOE
                    IF(COUNT(K).LT.1)GO TO 26
                          M=0
                          CET=0.
                          DO 27 J=1,NOCP
                          DO 27 L=1,NOCF
                          PP=P(I,J)
                          IF(PP.EQ.0)GO TO 27
                          IF(PP.NE.F(K,L))GO TO 27
                          CET=CET+TIME(PP)
                          M=M+1
                          DPIF(M)=PP
   27                     CONTINUE
                          IF(CET.LT.CT)GO TO 26
                                DO 30 J=1,M
                                MM=DPIF(J)
 30                             PRT(MM)=1
   26               CONTINUE
   22   CONTINUE
        IF(IB.EQ.0)GO TO 8999
        CALL INVERT(NOE,NOCP,NOCF,P,F,TIME)
        NUMIV=NOE+1
        DO 44 I=1,NIE
 44     NP(I)=NUMIV-NF(I)
        NIEP=NIE
        IH=NUMIV/2
        DO 43 I=1,IH
        IVN=NUMIV-I
        IPRT=PRT(IVN)
        PRT(IVN)=PRT(I)
 43     PRT(I)=IPRT
 8999   WRITE(6,9000)NOE
 9000   FORMAT('1',T8,'INPUT DATA FOR ',T24,I3,T27,'-ELEMENT PROBLEM',
       $/////,T8,'ELEMENT NO.',T27,'PRECEDENCE MATRIX')
        DO 2 I=1,NOE
        VF(5)=DIGIT(NOCP)
 2      WRITE(6,VF)I,(P(I,J),J=1,NOCP)
        WRITE(6,9002)
 9002   FORMAT(////,T8,'ELEMENT NO.',T27,'IMMEDIATE SUCCESSOR MATRIX')
        VF(5)=DIGIT(NOCF)
        DO 3 I=1,NOE
 3      WRITE(6,VF)I,(F(I,J),J=1,NOCF)
        WRITE(6,9003)
 9003   FORMAT(////,T8,'ELEMENT NO.',T25,'ELEMENT TIME',T40,'PRIORITY',
       $T53,'VARIANCE',T67,'CV2')
        DO 4 I=1,NOE
        CV2(I)=VAR(I)/TIME(I)**2
 4      WRITE(6,9004)I,TIME(I),PRT(I),VAR(I),CV2(I)
 9004   FORMAT(1X,T11,I3,T27,F7.2,T43,I1,T54,F6.3,T66,F6.4)
C******     PHASE  2
        IF(IZ.EQ.1)GO TO 7000
        PTR=0
        NPTR=0
        DO 40 J=1,NIEP
        PP=NP(J)
        IF(PRT(PP).NE.1)GO TO 401
            CALL SORT(PTR,PLIST,PP)
        GO TO 40
  401       CALL SORT(NPTR,NPLIST,PP)
   40   CONTINUE
        SNO=1
        EP=1
        CUMST=0.
        CUMVAR=0.
        SWP=0.
        SWNP=0.
        STMAX=0.
C******     TRY TO ASSIGN ELEMENTS IN PRIORITY LIST AND THEN ELEMENTS IN
C******     NON-PRIORITY LIST TO CURRENT STATION
  200   IF(PTR.LE.0)GO TO 300
  203   FLAG=0
        CALL ASSIGN(PTR,PLIST,SNO,EP,FLAG,LELE,SWP)
        IF(FLAG.EQ.0)GO TO 210
        IF(NPTR.LE.0.AND.SWP.EQ.1.)GO TO 203
        IF(NPTR.LE.0)GO TO 202
  201   FLAG=0
        CALL ASSIGN(NPTR,NPLIST,SNO,EP,FLAG,LELE,SWNP)
        IF(FLAG.EQ.0)GO TO 210
        IF(SWP.EQ.1..OR.SWNP.EQ.1.)GO TO 200
C******     ELEMENTS IN BOTH LISTS CANNOT BE ASSIGNED TO CURRENT STATION
  202   NWS(SNO)=EP-1
        STCV2(SNO)=CUMVAR/CUMST**2
        PROB(SNO)=CUMVAR/(2.*(CT-CUMST)**2)
        ST(SNO)=CUMST+ALPHA*SQRT(CUMVAR)
```

```
      CUMVAR=C.
      IF(ST(SNC).LE.CT)GC TC 2222
          STMAX=CT
          IDLE(SNC)=0.
      GO TO 2CCC
 2222     IF(ST(SNC).GT.STMAX)STMAX=ST(SNC)
          IDLE(SNC)=CT-ST(SNC)
 2000 EP=1
      SNO=SNC+1
      CUMST=C.
      IF(IZ.EG.1)GC TC 7299
      GO TO 200
C******    MODIFYING F-MATRIX
  210 DO 50 L=1,NCCF
      FF=F(LELE,L)
      IF(FF.EG.0)GC TC 50
      CZ=0
      DO 41 J=1,NCCP
      IF(P(FF,J).NE.0)GC TO 4000
          CZ=CZ+1
      GO TC 41
 4000     IF(P(FF,J).NE.LELE)GC TC 41
          P(FF,J)=0
          CZ=CZ+1
   41 CONTINUE
      IF(CZ.NE.NCCF)GC TC 50
      IF(PRT(FF).NE.1)GO TO 500
          CALL SCRT(PTR,PLIST,FF)
      GO TC 50
  500     CALL SCRT(NPTR,NPLIST,FF)
   50 CONTINUE
      SWP=0.
      SWNP=0.
      GO TO 200
C******     TRY TC ASSICN ELEMENTS IN NCN-PRICRITY LIST WHEN THE PRIORITY
C******     LIST IS EMPTY. IF BOTH LISTS ARE EMPTY STOP.
  300 IF(NPTR.GT.0.OR.SWNP.EC.1.)GO TC 201
 7100     ST(SNC)=CUMST+ALPHA*SCRT(CUMVAR)
          PROB(SNC)=CUMVAR/(2.*(CT-CUMST)**2)
      STCV2(SNC)=CUMVAR/CUMST**2
      IF(ST(SNO).GT.STMAX)STMAX=ST(SNC)
      IDLE(SNC)=CT-ST(SNC)
      NWS(SNO)=EP-1
      TIDL=0.
      TIDL2=0.
      WRITE(6,9006)NCE
 9006 FORMAT('1',///,T8,'CRITICAL PATH APPRCACH TO ASSEMBLY LINE',
     $T49,'BALANCING FCR',T63,I3,T66,'-ELEMENT PROBLEM')
 9100 FORMAT(T20,'WITH',T27,A20)
  650 WRITE(6,9101)C2,ALLOW
 9101 FORMAT(T20,'WITH',T27,A20,T50,' &     PRCBABILITY = ',
     $T72,F5.3)
      IF(IB.EG.0)GC TC 652
      WRITE(6,9100)C3
  652 IF(IZ.EG.0)GC TC 653
      WRITE(6,9100)C4
  653 WRITE(6,9001)
 9001 FORMAT('C',//,T8,'STATICN NO.',T30,'ELEMENT IN THE STATION',/)
      DO 60 I=1,SNC
      NWSE=NWS(I)
      TIDL=TIDL+IDLE(I)
      IF(STMAX.NE.CT)GC TO 600
          TIDL2=TIDL2+IDLE(I)**2
      GO TO 605
  600     TIDL2=TIDL2+(STMAX-ST(I))**2
  605 III=(NWSE/10*10+NWSE-NWSE/10*10+10)/10
      IIL1=III-1
      IF(IIL1.EG.0)GC TC 602
      DO 601 IJ=1,IIL1
      J1=(IJ-1)*10+1
      J2=IJ*10
      WVF(4)=DIGIT(10)
  601 WRITE(6,WVF)I,(WS(I,J),J=J1,J2)
  602 J1=IIL1*10+1
      NNN=NWSE-J1+1
      IF(NNN.EG.0)GO TC 60
      WVF(4)=DIGIT(NNN)
      WRITE(6,WVF)I,(WS(I,J),J=J1,NWSE)
   60 CONTINUE
      WRITE(6,9200)
 9200 FORMAT('C',///,T8,'STATICN NO.',T22,'CYCLE TIME',T35,
     $'STATICN TIME',T50,'ICLE TIME',T62,'# CF ELEMENTS',
     $T80,'STATICN CV2',T96,'P(ST > CT)',/)
      DO 80 I=1,SNC
 80   WRITE(6,9201)I,CT,ST(I),IDLE(I),NWS(I),STCV2(I),PROB(I)
 9201 FORMAT(1X,T11,I3,T23,F6.2,T38,F6.2,T50,F7.3,T65,I3,
     $T82,F7.5,T98,F5.3)
   75 E=(1.-TICL/(CT*SNC))*100
      SI=SQRT(TIDL2)
      WRITE(6,9009)TIDL,STMAX
 9009 FORMAT(1X,T50,7('-'),/,T50,F7.3,///,'C',
     $T30,'MAXIMUM STATICN TIME',T55,'=',T57,F6.2)
      WRITE(6,9012)E,SI
 9012 FORMAT(//,'0',T30,'OUTPUT EFFICIENCY',T55,'=',T57,F6.2,T63,'%',
     $    //,'0',T30,'SMOOTHNESS INDEX',T55,'=',T56,F7.2,/,'1')
      GO TO 1111
C******    CONSIDERING THE ZONING CONSTRAINTS
 7000 DO 7005 I=1,NOE
      READ(5,5000)(ZCNE(I,J),J=1,NZ)
 5000 FORMAT(10I1)
      SZ(I)=0
      DO 7005 J=1,NZ
 7005 SZ(I)=SZ(I)+ZCNE(I,J)
      WRITE(6,9105)
 9105 FORMAT(////,T11,'ZCNE',T27,'ELEMENTS',/)
```

```
      DO 7008 J=1,NZ
      IZC=0
      DO 7099 I=1,NCE
      IF(ZONE(I,J).EQ.0)GO TO 7099
      IZC=IZC+1
      NF(IZC)=I
7099  CONTINUE
      VF(5)=DIGIT(IZC)
7008  WRITE(6,VF)J,(NF(IJ),IJ=1,IZC)
      DO 7007 KK=1,24
7007  AVAIL(KK,2)=KK+1
      AVAIL(KK,2)=0
C******   CHOOSE AMONG THE AVAILABLE ELEMENTS THE ONE THAT HAS THE
C******   SMALLEST ELEMENT CV2.  FIND OUT THE ZONE AREA FOR THIS
C******   CHOSEN ELEMENT & PLACE THE REST OF THE AVAILABLE ELEMENTS
C******   ONTO THE ZLIST, NZLIST, PLIST, NPLIST & AVAIL
      ZPTR=0
      NZPTR=0
      PTR=0
      NPTR=0
      FIND=0.
      APTR=0
      PPTR=1
      SWZ=0.
      SWP=0.
      SWNP=0.
      SWD=0.
      TCV2=100.
      DO 7001 J=1,NIEF
      PP=NP(J)
      IF(PRT(PP).NE.1)GO TO 7010
      IF(CV2(PP).GE.TCV2)GO TO 7010
      TCV2=CV2(PP)
      IF(SWZ.EQ.0.)GO TO 7009
      CALL INSERT(MPE)
7009  MPE=PP
      SWZ=1.
      GO TO 7001
7010  CALL INSERT(PP)
7001  CONTINUE
      CALL SEF(NZONE,MPE,ZLIST,ZPTR,NZLIST,NZPTR,PLIST,PTR,NPLIST,NPTR)
      SNO=1
      EP=1
      CUMST=0.
      STMAX=0.
7299  IEMPTY=PTR+NPTR+ZPTR+NZPTR
C******   IF NO AVAILABLE ELEMENTS CAN BE PLACED INTO THE CURRENT
C******   ZONE AREA, FIND A NEW ZONE AREA
      IF(IEMPTY.EQ.0.AND.APTR.EQ.0)GO TO 7100
      IF(IEMPTY.EQ.0)GO TO 7200
      DO 7301 I=1,NCE
      IF(ZONE(I,NZONE).EQ.1.AND.SZ(I).EQ.1)GO TO 7302
7301  IF(ZONE(I,NZONE).EQ.1.AND.SZ(I).GT.1.AND.EP.GT.1)GO TO 7302
      GO TO 7200
C******   ASSIGNMENT OF ELEMENTS TO CURRENT STATION.
7302  IF(ZPTR.LE.0)GO TO 7300
7304  FLAG=0
      CALL ASSIGN(ZPTR,ZLIST,SNO,EP,FLAG,LELE,SWP)
      IF(FLAG.EQ.0)GO TO 7310
      IF(NZPTR.LE.0.AND.SWP.EQ.1.)GO TO 7304
7300  IF(NZPTR.LE.0)GO TO 7320
7305  FLAG=0
      CALL ASSIGN(NZPTR,NZLIST,SNO,EP,FLAG,LELE,SWNP)
      IF(FLAG.EQ.0)GO TO 7310
      IF(SWNP.EQ.1..AND.SWP.EQ.1.)GO TO 7304
      IF(SWNP.EQ.1.)GO TO 7305
7320  IF(PTR.LE.0)GO TO 7330
      FLAG=0
      CALL ASSIGN(PTR,PLIST,SNO,EP,FLAG,LELE,SWD)
      IF(FLAG.EQ.0)GO TO 7310
7330  IF(NPTR.LE.0)GO TO 202
      FLAG=0
      CALL ASSIGN(NPTR,NPLIST,SNO,EP,FLAG,LELE,SWD)
      IF(FLAG.EQ.0)GO TO 7310
      GO TO 202
C******   MODIFYING THE P-MATRIX UNDER ZONING CONSTRAINTS
7310  DO 7342 J=1,NZ
7342  ZONE(LELE,J)=0
      DO 7350 L=1,NOCF
      FF=F(LELE,L)
      IF(FF.EQ.0)GO TO 7350
      CZ=0
      DO 7341 J=1,NOCP
      IF(P(FF,J).NE.0)GO TO 7345
      CZ=CZ+1
      GO TO 7341
7345  IF(P(FF,J).NE.LELE)GO TO 7341
      P(FF,J)=0
      CZ=CZ+1
7341  CONTINUE
      IF(CZ.NE.NOCP)GO TO 7350
      IF(ZONE(FF,NZONE).EQ.1)GO TO 7800
      CALL INSERT(FF)
      GO TO 7350
7800  IF(PRT(FF).EQ.0)GO TO 7825
      IF(SZ(FF).GT.1)GO TO 7820
      CALL SORT(ZPTR,ZLIST,FF)
      GO TO 7350
7820  CALL SORT(PTR,PLIST,FF)
      GO TO 7350
7825  IF(SZ(FF).GT.1)GO TO 7826
      CALL SORT(NZPTR,NZLIST,FF)
      GO TO 7350
7826  CALL SORT(NPTR,NPLIST,FF)
```

```
 7350   CONTINUE
        SWP=0.
        SWNP=0.
        GO TO 7299
C******    START A NEW ZONE AREA. PUT ALL ELEMENTS IN THE PLIST &
C******    NPLIST ONTO THE LINKED LIST AVAIL.
 7200   IF(PTR.EQ.0)GO TO 7210
        DO 7201 J=1,PTR
        PP=PLIST(J)
        ZONE(PP,NZONE)=0
        SZ(PP)=SZ(PP)-1
 7201   CALL INSERT(PP)
        PTR=0
 7210   IF(NPTR.EQ.0)GO TO 7220
        DO 7202 J=1,NPTR
        PP=NPLIST(J)
        ZONE(PP,NZONE)=0
        SZ(PP)=SZ(PP)-1
 7202   CALL INSERT(PP)
        NPTR=0
C******    FROM THE LINKED LIST AVAIL, FIND THE ONE THAT HAS THE LARGEST
C******    ELEMENT TIME WITH PREFERENCE GIVEN TO PRIORITY ELEMENTS
 7220   IP=AVAIL(APTR,2)
        IF(IP.NE.0)GO TO 7221
        MPE=AVAIL(APTR,1)
        CALL DELETE(0,APTR)
        GO TO 7262
 7221   MPE=0
        SNPE=0.
 7222   SWTH=0.
        UPTR=0
        CPTR=APTR
        FIND=0.
        TCV2=100.
 7251   MVAIL=AVAIL(CPTR,1)
        IF(SNPE.EQ.1.)GO TO 7252
        IF(PRT(MVAIL).EQ.0)GO TO 7250
 7252   IF(CV2(MVAIL).GE.TCV2)GO TO 7250
        TCV2=CV2(MVAIL)
        IF(SWTH.EQ.0.)GO TO 7265
        CALL INSERT(MPE)
        IF(UPTR.EQ.0)UPTR=APTR
        GO TO 7266
 7265   SWTH=1.
 7266   MPE=MVAIL
        CALL DELETE(UPTR,CPTR)
        GO TO 7261
 7250   UPTR=CPTR
        CPTR=AVAIL(UPTR,2)
 7261   IF(CPTR.NE.0)GO TO 7251
        IF(MPE.NE.0)GO TO 7262
        SNPE=1.
        GO TO 7222
C******    FIND OUT THE NEW ZONE AREA FROM THE CHOSEN ELEMENT & PLACED
C******    THE REST OF THE ELEMENT IN AVAIL TO ZLIST, NZLIST, PLIST
C******    & NPLIST
 7262   CALL SEP(NZONE,MPE,ZLIST,ZPTR,NZLIST,NZPTR,PLIST,PTR,NPLIST,NPTR)
        IF(EP.GT.1)GO TO 202
        GO TO 7299
 9999   STOP
        END
        SUBROUTINE SORT(LPTR,LIST,LELE)
C******    THIS ROUTINE SORT THE AVAILABLE ELEMENTS IN PRIORITY OR
C******    NON-PRIORITY LIST USING LARGEST CANDIDATE RULE
        INTEGER WS(25,20),AVAIL(25,2),ZONE(50,7),SZ(50),APTR,PPTR
        INTEGER PRT(50)
        DIMENSION TIME(50),LIST(50),VAR(50),CV2(50)
        COMMON TIME,WS,CT,CUMST,AVAIL,ZONE,SZ,APTR,PPTR,NZ,PRT
        COMMON CMEAN,OVAR,CVO2,VAR,ALPHA,CUMVAR,STIME,CV2
        IF(LPTR.NE.0)GO TO 5
        LIST(1)=LELE
        GO TO 15
    5   DO 1 I=1,LPTR
        L=LIST(I)
        IF(TIME(LELE).LE.TIME(L))GO TO 1
        J=LPTR
    7   IF(J.LT.I)GO TO 10
        LIST(J+1)=LIST(J)
        J=J-1
        GO TO 7
    1   CONTINUE
   10   LIST(I)=LELE
   15   LPTR=LPTR+1
        RETURN
        END
        SUBROUTINE ASSIGN(LPTR,LIST,SNO,EP,FLAG,LELE,SWV)
C******    THIS ROUTINE ASSIGNS THE AVAILABLE ELEMENTS TO THE CURRENT
C******    STATION & DELETES THAT ASSIGNED ELEMENT FROM THE PRIORITY OR
C******    NON-PRIORITY LIST
        INTEGER SNO,EP,FLAG,WS(25,20)
        INTEGER AVAIL(25,2),ZONE(50,7),SZ(50),APTR,PPTR,PRT(50)
        DIMENSION TIME(50),LIST(50),VAR(50),CV2(50)
        COMMON TIME,WS,CT,CUMST,AVAIL,ZONE,SZ,APTR,PPTR,NZ,PRT
        COMMON CMEAN,OVAR,CVO2,VAR,ALPHA,CUMVAR,STIME,CV2
        SW=0.
        UL=-.05*ALPHA
        IF(EP.GT.1)GO TO 3
          DO 1 K=1,LPTR
          LELE=LIST(K)
          IF(K.EQ.1)GO TO 2
          IF(CV2(LELE).GT.CVO2)GO TO 1
    2     CVO2=CV2(LELE)
          II=K
    1     CONTINUE
          LELE=LIST(II)
```

```
          CUMST=TIME(LELE)
          CUMVAR=VAR(LELE)
          OMEAN=CUMST
          OVAR=CUMVAR
      GO TO 20
    3     SW1=0.
C******    BY PASS THE COEFFICIENT OF VARIATION CRITERION
          IF(SWV.EQ.0.)GO TO 12
                DO 18 K=1,LPTR
                LELE=LIST(K)
                CUMST=CUMST+TIME(LELE)
                CUMVAR=CUMVAR+VAR(LELE)
                Z=(CT-CUMST)/SQRT(CUMVAR)
                IF((Z-ALPHA).LT.UL)GO TO 19
                CVC2=(CVAR+VAR(LELE))/(OMEAN+TIME(LELE))**2
                IF(SW1.EQ.1.)GO TO 60
                      CV2MIN=CVC2
                      II=K
                      SW1=1.
                GO TO 19
   60                 IF(CVC2.GE.CV2MIN)GO TO 19
                      CV2MIN=CVC2
                      II=K
   19           CUMST=CUMST-TIME(LELE)
 18             CUMVAR=CUMVAR-VAR(LELE)
                LELE=LIST(II)
                CUMST=CUMST+TIME(LELE)
                CUMVAR=CUMVAR+VAR(LELE)
                OMEAN=OMEAN+TIME(LELE)
                OVAR=OVAR+VAR(LELE)
                CVO2=CV2MIN
                SWV=0.
                GO TO 20
C******     FIND OUT ELEMENTS THAT CAN BE ASSIGNED TO THE PRESENT STATION
   12           DO 15 K=1,LPTR
                LELE=LIST(K)
                CUMST=CUMST+TIME(LELE)
                CUMVAR=CUMVAR+VAR(LELE)
                Z=(CT-CUMST)/SQRT(CUMVAR)
                IF((Z-ALPHA).LT.UL)GO TO 13
                R=OMEAN/TIME(LELE)
                C=(1+2*R)*CVO2
                CVN2=CV2(LELE)
C******     FROM THESE CHOSEN ELEMENTS, THE ONE THAT GIVES THE MINIMUM
C******     STATION CV2 IS ASSIGNED
                IF(CVN2.LE.C)GO TO 70
                      SWV=1.
                      GO TO 13
   70           SWV=0.
                SW=0.
                IF(SW1.EQ.1.)GO TO 80
                      II=K
                      CV2MIN=CVN2
                      SW1=1.
                GO TO 14
   80                 IF(CVN2.GE.CV2MIN)GO TO 14
                      CV2MIN=CVN2
                      II=K
                GO TO 14
   13           SW=1.
   14           CUMST=CUMST-TIME(LELE)
 15             CUMVAR=CUMVAR-VAR(LELE)
                IF(SW.EQ.0.)GO TO 90
                      FLAG=1
                      RETURN
   90                 LELE=LIST(II)
                      CUMST=CUMST+TIME(LELE)
                      CUMVAR=CUMVAR+VAR(LELE)
                      OMEAN=OMEAN+TIME(LELE)
                      CVAR=CVAR+VAR(LELE)
                      CVC2=CVAR/OMEAN**2
   20 WS(SNO,EP)=LELE
      EP=EP+1
      LPTR=LPTR-1
      IF(II.LE.LPTR)GO TO 100
         LIST(II)=0
      RETURN
  100    IF(LPTR.LE.0)RETURN
                DO 10 J=II,LPTR
 10             LIST(J)=LIST(J+1)
                LIST(J)=0
      RETURN
      END
      SUBROUTINE SEP(NZE,MPE,ZL,ZPTR,NZL,NZPTR,PL,PTR,NPL,NPTR)
C******     THIS ROUTINE FINDS THE NEW ZONE AREA FROM THE CHOSEN ELEMENT
C******     (MPE) AND CALLS THE SORT ROUTINE TO SORT ELEMENTS IN THE
C******     LINKED LIST AVAIL TO ZLIST, NZLIST, PLIST, NPLIST.
      INTEGER AVAIL(25,2),ZONE(50,7),SZ(50),ZL(50),ZPTR,PTR,APTR
      INTEGER PL(50),PRT(50),UPTR,CPTR,PPTR,WS(25,20)
      DIMENSION NZL(50),NPL(50),TIME(50),VAR(50),CV2(50)
      COMMON TIME,WS,CT,CUMST,AVAIL,ZONE,SZ,APTR,PPTR,NZ,PRT
      COMMON OMEAN,OVAR,CVO2,VAR,ALPHA,CUMVAR,STIME,CV2
      DO 2 J=1,NZ
2     IF(ZONE(MPE,J).EQ.1)GO TO 11
   11 NZE=J
      IF(PRT(MPE).EQ.0)GO TO 13
      CALL SORT(ZPTR,ZL,MPE)
      GO TO 14
   13 CALL SORT(NZPTR,NZL,MPE)
   14 IF(APTR.EQ.0)RETURN
      UPTR=0
      CPTR=APTR
   12 MVAIL=AVAIL(CPTR,1)
      IF(ZONE(MVAIL,NZE).EQ.0)GO TO 33
      IF(PRT(MVAIL).EQ.0)GO TO 25
```

```
      IF(SZ(MVAIL).GT.1)GO TC 20
      CALL SCRT(ZPTR,ZL,MVAIL)
      GO TO 31
   20 CALL SCRT(PTR,PL,MVAIL)
      GO TC 31
   25 IF(SZ(MVAIL).GT.1)GO TC 26
      CALL SCRT(NZPTR,NZL,MVAIL)
      GO TO 31
   33 UPTR=CPTR
      CPTR=AVAIL(UPTR,2)
      GO TC 32
   26 CALL SCRT(NPTR,NPL,MVAIL)
   31 CALL DELETE(LPTR,CPTR)
   32 IF(CPTR.GT.C)GC TC 12
      RETURN
      END
      SUBROUTINE INSERT(MPE)
C****** THIS RCUTINE INSERT AN ELEMENT ONTC THE LINKED LIST AVAIL
      INTEGER WS(25,2C),AVAIL(25,2),ZCNE(50,7),SZ(50)
      INTEGER PRT(50),PPTR,APTR
      DIMENSICN TIME(50),VAR(50),CV2(50)
      COMMCN TIME,WS,CT,CUMST,AVAIL,ZONE,SZ,APTR,PPTR,NZ,PRT
      COMMCN CMEAN,OVAR,CVC2,VAR,ALPHA,CUMVAR,STIME,CV2
      IF(PPTR.EQ.0)GC TO 1
      IEMP=PPTR
      PPTR=AVAIL(IEMP,2)
      AVAIL(IEMP,2)=APTR
      AVAIL(IEMP,1)=MPE
      APTR=IEMP
      RETURN
    1 PRINT,'EMPTY PCCL'
      RETURN
      END
      SUBROUTINE CELETE(IPTR,JPTR)
C****** THIS RCUTINE DELETES AN ELEMENT FRCM THE LINKED LIST AVAIL
      INTEGER WS(25,20),AVAIL(25,2),ZCNE(50,7),SZ(50)
      INTEGER PRT(50),PPTR,APTR
      DIMENSICN TIME(5C),VAR(50),CV2(50)
      COMMCN TIME,WS,CT,CUMST,AVAIL,ZCNE,SZ,APTR,PPTR,NZ,PRT
      COMMCN CMEAN,OVAR,CVO2,VAR,ALPHA,CUMVAR,STIME,CV2
      IF(IPTR.NE.C)GC TC 1
      IEMP=APTR
      APTR=AVAIL(IEMP,2)
      JPTR=APTR
      AVAIL(IEMP,2)=PPTR
      PPTR=IEMP
      RETURN
    1 AVAIL(IPTR,2)=AVAIL(JPTR,2)
      AVAIL(JPTR,2)=PPTR
      PPTR=JPTR
      JPTR=AVAIL(IPTR,2)
      RETURN
      END
      SUBROUTINE INVERT(NCE,NCCP,NCCF,P,F,TIME)
C****** THIS RCUTINE INVERTS THE ELEMENT NUMBER IN P-MATRIX, F-MATRIX
C****** & TIME LIST FCR BACKWARD ASSIGNMENT
      INTEGER P(5C,15),F(50,15)
      DIMENSICN TIME(5C)
      IF(NOCF-NOCF)10,11,12
   10 JJ=NOCP+1
      DO 1 I=1,NCE
      DO 1 J=JJ,NCCF
    1 P(I,J)=0
      GO TC 11
   12 JJ=NCCF+1
      DO 2 I=1,NCE
      DO 2 J=JJ,NCCP
    2 F(I,J)=C
   11 NCOL=(NCCP+NCCF+1)/2
      NUM=NCE+1
      DO 3 I=1,NCE
      IVN=NUM-I
      DO 3 J=1,NCCL
      ITEMP=0
      IF(F(IVN,J).EQ.0)GC TC 2C
      ITEMP=NUM-F(IVN,J)
   20 IF(P(I,J).EC.C)GC TO 21
      F(IVN,J)=NUM-P(I,J)
      GO TC 22
   21 F(IVN,J)=0
   22 P(I,J)=ITEMP
    3 CONTINUE
      IHALF=NUM/2
      DO 5 I=1,IHALF
      IVN=NUM-I
      TEMP=TIME(IVN)
      TIME(IVN)=TIME(I)
    5 TIME(I)=TEMP
      ITEMP=NOCP
      NOCP=NCCF
      NOCF=ITEMP
      RETURN
      END
```