Computers ind. Engng Vol. 14, No. 2, pp. 193–200, 1988
Printed in Great Britain

0360-8352/88 $3.00+0.00
Pergamon Press plc

DETERMINATION OF AGGREGATE PREVENTIVE MAINTENANCE PROGRAMS USING PRODUCTION SCHEDULES

S. Sridhar
Administrative Staff College of India, Bella Vista, Hyderabad 500049, India

(Received for publication 6 April 1987)

Abstract—Scheduling preventive maintenance in a multi-period and multi-product situation has been dealt with in this paper. An aggregate preventive maintenance programme is determined based on the time to failure distribution, machine load in every period and varying cost of breakdown in different periods; using a steppingstone algorithm. Sample results from the computer programme written for solving this problem are given. This algorithm can be used to solve similar problems.

INTRODUCTION

The scheduling of preventive maintenance has been extensively dealt with by many researchers [1]. A recent paper [2] has however given a new look to the problem. The problem has been posed in the Joshi and Gupta paper as aggregate planning of preventive maintenance (PM) over a number of periods given that the breakdown cost in each period varies as a function of the load in the period and the profitability of that period's product-mix.

Joshi and Gupta have used a complete search technique for solving the problem. A more efficient method of solving this problem is described in this paper. This method uses a steppingstone scheme which can be adopted to solve similar problems.

PROBLEM DESCRIPTION

Performing routine or preventive maintenance has always been a source of conflict generation between executives managing the production and maintenance functions. A machine breakdown in peak load periods or in highly profitable periods causes heartburns. Medium to long term planning of routine maintenance is very much necessary in this case in order to account and plan for the interactions between machine load, product-mix profitability and machine breakdowns.

Joshi and Gupta [2], provide for the interaction by varying the cost of breakdown by using a weighted average cost. The formula is given below:

$$\text{Average cost of downtime in period } j(B_j) = \frac{\sum_i t_i \times Q_{ij} \times C_i \times t_D}{H_{\max}} \times \frac{\sum_i Q_{ij}}{\sum_i Q_{ij} \times t_i}$$

$$= \left(\textit{Average value of production in period } j \times \text{average downtime in hours} \times \frac{1}{\text{Average time required per unit of production}}\right)$$

where C_i is the "value" of product i being produced on the machine in period j, Q_{ij} is the quantity of product i produced on the machine in period j, t_i is the time taken by product i on the machine in hours/unit, $H_{\max}$ the maximum permissible load on the machine and t_D the mean downtime in hours from historical data.

In this paper it is assumed that the breakdown costs, in each period, have been computed using this or a similar formula. An efficient solution method is proposed to solve the problem of minimizing the total cost of maintenance.

PROBLEM DEFINITION

Given that:

N = number of periods in the planning horizon
t = time in hours
$f(t)$ = probability density function of failure (after repair) of the machine for which the PM schedule is being generated
B_i = cost of breakdown per hour of the machine in period i
t_D = average total waiting plus breakdown repair hours per breakdown
L_i = hours of load on the machine in period i
P = average cost of one preventive maintenance
X_i = 0, if no preventive maintenance is scheduled in period i
= 1, if preventive maintenance is scheduled in period i
M = number of preventive maintenance to be performed within N, the planning horizon
H_{ij} = expected number of failures in period j given the last PM was in period i
$K_n(S_r)$ = cost of scheduling the r_{th} P_m in the S^{th} period when $M = n$
$H(t)$ = expected number of failures in time t
J = set of periods in which PM is performed where $j(n)$ is the period in which the n^{th} PM is scheduled.

The problem can be defined as:

$$\operatorname{Min}_{M=0 \text{ to } N}\left(\operatorname{Min}_{J=\{j(r),\, r=1 \text{ to } M\}}\left\{\sum_{i=1}^{N} t_D \times B_i \times H_{j(s)i} + X_i \times P\right\}\right)$$

Where:

$$j(s) \in J \text{ and } j(s) \geqslant j(t);\ j(t) \leqslant i \text{ and } j(t) \in J$$
$$t = 1 \text{ to } M$$

The problem defined has two dimensions, the first of determining the optimal schedule J for a given value of M, and the second of optimising over the value of M.

It is assumed that B_i, t_D, and P are determined from historic data. L_i is computed from the production schedule for the N periods, for the machine. It is assumed that any PM will be scheduled at the beginning of a period.

H_{ij} as used in the computation is determined by first computing the function $H(t)$, through the recursion equation given below [3]:

$$H(t) = \sum_{u=0}^{t-1} \int_{u}^{u+1} f(x) \, . \, (1 + H(t-u-1) \, . \, dx$$

$$\text{with } H(0) = 0 \qquad \text{(i)}$$

SOLUTION

Consider the case of $M = 1$. Without loss of generality we can assume that a newly repaired (or new machine) is available at the beginning of the first period.

This will imply that if the scheduling of the PM in the n^{th} period is locally optimum, then the cost will increase on shifting the PM to either of the adjacent periods, $(n-1)$ or $(n+1)$.

Thus:

$$K_1(n+1) \geqslant K_1(n)$$
$$K_1(n-1) \geqslant K_1(n) \quad (n \neq 1)$$

or:

$$\sum_{j=1}^{n} H_{1j} \times B_j + \sum_{j=n+1}^{N} H_{n+1j} \times B_j \geqslant \sum_{j=1}^{n-1} H_{1j} \times B_j + \sum_{j=n}^{N} H_{nj} \times B_j$$

and

$$\sum_{j=1}^{n-2} H_{1j} \times B_j + \sum_{j=n-1}^{N} H_{n+1j} \times B_j \geqslant \sum_{j=1}^{n-1} H_{1j} \times B_j + \sum_{j=n}^{N} H_{nj} \times B_j \qquad \text{(ii)}$$

or

$$H_{1n} \times B_n - H_{nn} \times B_n + \sum_{j=n+1}^{N} (H_{n+1j} - H_{nj}) \times B_n \geqslant 0$$

and

$$H_{n-1n-1} \times B_{n-1} - H_{1n-1} \times B_{n-1} + \sum_{j=n}^{N} (H_{n-1j} - H_{nj}) \times B_n \geqslant 0 \qquad \text{(iii)}$$

All n satisfying (iii) are locally optimum. Because the cost function can take any shape a complete search is required. But due to the compactness of the formula, if the totals $\sum_{j=1}^{r} (H_{ij} \times B_j)$ are computed and stored in a square upper triangular matrix, then the computations can be performed quite fast, using (ii) itself.

To extend this to other values of M, consider the case of M=2. The schematic representation is given in Fig. 1.

Fig. 1

If the PM[s] are done in periods S and R then:

(i) R must be optimum for the span S to N

(ii) S must be optimum for the span 1 to $(R-1)$

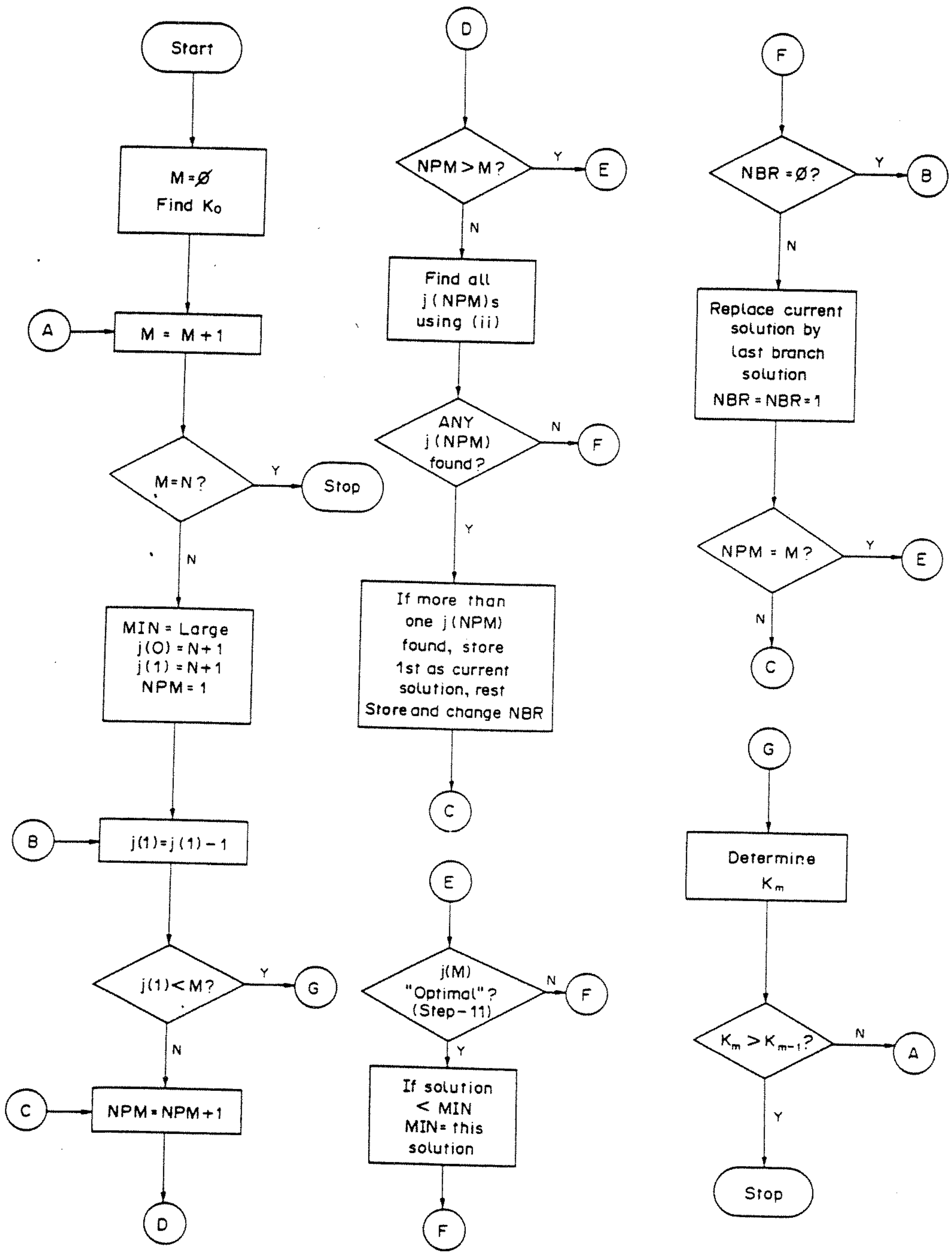

Fig. 2

The steppingstone algorithm therefore assumes the value of R, one-by-one decreasing it from N to 3. S is found from (ii) for the span S to N, for each R. If multiple values of S are there, then these are stored.

After S has been determined, (ii) is used to check the local optimality between 1 to $(R-1)$. If it is not "optimal" then the stored solutions for the same value of R are checked. If they are not optimal, these solutions are deleted.

If S and R are "optimal" then this solution is compared with the previous minimum "optimal" solution (if any). The minimum solution is stored.

After searching through all values of R from N to 3, the minimum solution is obtained. If there is no such solution, then two PMs are not required.

For the general case, it is a repetitive application of the steppingstone method until the last PM has been scheduled. The last span is checked for "optimality". The flow chart of the computations is given in Fig. 2 and the step by step description given below.

THE STEPPINGSTONE SEARCH ALGORITHM

1. Set $M = 0$. Find K_0, the cost of having no PM^s.
2. $M = M + 1$. If $M = N$ STOP
3. Set, MIN = large number.
4. NBR, No. of branches = zero
   Set the first PM in the $(N + 1)^{st}$ period
5. $j(1) = N + 1$; (note: $j(0) = N + 1$ in Step 9)
6. Set the No. of PM^s scheduled NPM = 1
7. $j(1) = j(1) - 1$. If $j(1)$ is less than M go to step 16
8. Increase NPM by one
   If NPM $> M$, go to step 11.
9. Find the next PM, j(NPM).
   Here (ii) is used within the specified span j(NPM) to j(NPM−2), with j(NPM−1) as the "optimal" PM in the span. If there is more than one solution, store these and increase NBR correspondingly. Go to step 8.
10. If the next PM cannot be found in step 9, go to step 14.
11. Check "optimality" using $j(M)$ in the span 1 to $j(M-1)$, using (ii).
12. If not "optimal" go to step 14.
13. If "optimal" compare the solution with previous MIN and store the least solution.
14. If NBR = 0, go to step 7.
15. Replace the current solution by the last branch solution. Reduce NBR by one. If NPM of the current solution is equal to M directly go to step 11. Else go to step 8.
16. Determine K_M. If $K_M > K_{M-1}$, STOP. Otherwise go to step 2.

COMPUTER PROGRAM

A computer program was written in FORTRAN and run on the VAX 11/730 at the Administrative Staff College of India, Hyderabad. The program reads the value of H_j computed by another program. This enables different density functions to be tried out. (The Weibull distribution was used in the runs). There are two sub-routines, one to compute the next PM, the other to check the final span's optimality. The output for some of the values are given in Tables 1–6.

Table 1. The function $H(t)$: selected values

Hours since last PM	Expected no. of breakdowns (rounded off)
0	0.00000
50	0.00960
100	0.03870
200	0.15042
300	0.31949
400	0.52581
500	0.75093
600	0.98188
700	1.21204
800	1.43961
900	1.66522
1000	1.89007

Note: $f(t) = (b/a).\ [(t-c)/a]^{b-1}.\ \text{Exp} -[(t-c)/a]^b$ for $t > C$ (Weibull density function) using $a = 500$ hours; $b = 2$; $c = 0$ hr.

Table 2

	Period i											
	1	2	3	4	5	6	7	8	9	10	11	12
B_i breakdown cost	40	70	80	70	69	57	30	99	74	70	75	77
L_i, load in hours	50	80	60	80	90.7	60.4	80	90.5	70.6	80.6	80.6	70.8

Table 3. The function $H_{ij} B_j$ (rounded off) expected cost of breakdown in period (assuming, t_D = 1 hr, as the total waiting plus repair time per breakdown)

		1	2	3	4	5	6	7	8	9	10	11	12
Last PM in	1	0.383	3.887	5.700	8.930	12.155	7.396	5.433	20.575	11.952	12.817	13.610	12.156
period	2		1.733	4.046	7.346	10.937	6.977	5.274	20.407	11.979	12.890	13.684	12.190
	3			1.109	4.303	8.327	5.960	4.819	19.644	11.871	12.948	13.811	12.300
	4				1.733	5.884	4.901	4.283	18.517	11.582	12.874	13.870	12.387
	5					2.163	3.128	3.293	16.079	10.778	12.478	13.793	12.463
	6						0.790	1.844	11.947	9.137	11.412	13.282	12.352
	7							0.743	8.442	7.565	10.231	12.555	12.051
	8								3.104	4.918	8.012	10.962	11.215
	9									1.400	4.712	8.233	9.508
	10										1.733	5.479	7.562
	11											1.857	4.734
	12												1.457

Table 4. Expected cost of breakdowns from period i to period j if last PM is in period i

1	2	3	4	5	6	7	8	9	10	11	12
0.383	4.270	9.970	18.900	31.055	38.451	43.884	64.459	76.411	89.228	102.838	114.994
—	1.733	5.779	13.125	24.062	31.039	36.313	56.720	68.699	87.589	95.273	107.463
	—	1.109	5.412	13.739	19.699	24.578	44.162	56.033	68.987	82.792	95.092
		—	1.733	7.617	12.578	16.801	35.318	46.900	59.774	73.644	86.031
			—	2.163	5.291	8.584	24.663	35.441	47.919	61.712	74.175
				—	0.790	2.634	14.581	23.718	35.130	48.412	60.764
					—	0.743	9.185	16.750	26.981	39.536	51.587
						—	3.104	8.022	16.034	26.996	38.211
							—	1.400	6.112	14.335	28.843
								—	1.733	7.212	14.774
									—	1.857	6.591
										—	1.457

The expected number of breakdowns (see Table 1) are computed using the recursion relation (i) and assuming the parameters of the Weibull distribution, $f(t) = (b/a) [(t-c)/a]^{b-1}$. Exp. $-[(t-c/a)^b]$, to be a = 500 hr, b = 2 and c = 0. Table 2 gives the expected machine load and the cost of breakdown for 12 periods. Table 3 shows the cost of breakdowns in period j if the last preventive maintenance is performed at the beginning of period i. Here it is assumed that the average time for waiting and repair per breakdown, t_D, is one hour. Table 3 is converted into Table 4, to give the expected cost of breakdowns from period i to period j, where i is the last period in which a PM is performed, prior to period j+1. Table 5 gives the minimum expected cost of maintenance when there are one to four set-ups for preventive maintenance. Here it is assumed that the average cost per set-up for preventive maintenance is 15.0. The optimal policy is found to have three set-ups for preventive maintenance in the periods 4, 8 and 11.

In the actual program, the processing does not stop at $K_M > K_{M-1}$ but proceeds till M = 11. This way the breakeven value for the PM cost in order to have one more PM, is computed as the difference between the optimal breakdown costs for successive number of PM^s. (See Table 6).

APPLICATION TO THE DYNAMIC LOT SIZE PROBLEM

This algorithm can be used to solve similar problems such as the problem of determining dynamic lot sizes [5]. The entries of Table 4 can be visualized as the cost of carrying inventory to meet the demand for periods i to j, when the last set-up for production is at the beginning of period i. The cost of one preventive maintenance could be equated to the cost of a setup for production. In case the cost for a set-up is different in different periods, then this too can be incorporated in Table 4 itself by adding the cost of set-up to each row. The only difference in the approach for computing dynamic lot sizes would be that there may be carry-over inventory from periods prior to the planning horizon. However it can be assumed without loss of generality that, the first setup will be in the first period of the planning horizon; with the demand in the first period suitably adjusted to take care of partial fulfilment of that period's demand from the carry-over inventory.

Table 5. Solution for average cost of PM = 15.00

No. of PM^s	Period in which PM^s are to be performed J	Expected total cost of policy
0	—	114.994
1	(8)	97.094
2	(8, 4)	94.981
3	(11, 8, 4)	94.395
4	(11, 8, 5, 3)	100.891

Optimal policy is three PM^s in (11, 8, 4).

Table 6. Breakeven PM cost for having 'M' PM^s

No. of PM^s M	Periods in which PM^s are to be performed, J	Breakdown cost for policy (M, J) K_M	Breakeven cost of one PM for having this PM $K_M - K_{M-1}$
0		114.994	—
1	(8)	82.094	32.900
2	(8,4)	64.981	17.113
3	(11,8,4)	49.395	15.586
4	(11,8,5,3)	40.891	8.504
5	(11,9,8,5,3)	34.073	6.818
6	(11,9,8,6,5,3)	30.285	3.788
7	(12,11,9,8,6,5,3)	27.009	2.277
8	(12,11,10,9,8,6,5,3)	24.030	2.979
9	(12,11,10,9,8,6,5,4,3)	21.459	2.570
10	(12,11,10,9,8,6,5,4,3,2)	19.305	2.155
11	(12,11,10,9,8,7,6,5,4,3,2)	18.203	1.101

This will imply that a constant will get added to all the cost figures of the first row in Table 4, due to the setup in the first period.

The minimum cost for having different number of setups in the planning horizon can be determined using the same search algorithm. The optimal policy is to have 4 setups for production in periods 1, 4, 8 and 11 (from Table 5). The incremental cost for having more or less than 4 setups for production in the planning horizon can be found from Table 6. For example, the marginal increase in cost for having 3 setups in periods 1, 4 and 8 is 0.586; whereas the increase in cost for 5 setups is 6.496. This is a valuable input to the decision maker if there are different products competing for the same set of production facilities.

CONCLUSION

The algorithm described is quite fast taking about 2–4 sec of CPU time, after the H_{ij} matrix has been generated. This algorithm can be used for solving similar problems, an example being the dynamic lot size problem of Wagner and Whitin.

REFERENCES

1. I. B. Gertsbakh. *Models of Preventive Maintenance*. North-Holland, The Netherlands (1977).
2. S. Joshi and R. Gupta. Scheduling of routine maintenance using production schedules and equipment failure history. *Computers ind. Engng* **10**, 11–20 (1986).
3. A. K. S. Jardine. Equipment replacement strategies. In *Operational Research in Maintenance* (Edited by Jardine A. K. S.). Barnes & Noble, New York (1970).
4. D. R. Cox. *Renewal Theory*. Methuen, London (1962).
5. H. M. Wagner and T. H. Whitin. Dynamic version of the economic lot size model. *Mgmt Sci.* **5**, 89–96 (1958).