WORKING PAPER No.194

A Solution Procedure for the Multi-Component Deterministic Opportunistic Replacement Problem

by

A.K. Rao

&

M.R. Rao

July 2002

Please address all correspondence to:

A.K. Rao
Director & Professor
SDM Institute of Management Development
2254, Vinoba Road
Mysore 570 005
E-Mail : raoarza@yahoo.com or director@sdmimd.net
Phone : 0821-429722
Fax : 0821-425557

Prof. M. Rammohan Rao
Professor
Indian Institute of Management Bangalore
Bannerghatta Road
Bangalore – 560076, India
E-mail: mrao@iimb.ernet.in
Phone : 080 – 6993136
Fax : 080 - 6584050

A Solution Procedure for the Multi-Component Deterministic Opportunistic Replacement Problem

by

A.K. Rao*

M.R. Rao**

Abstract

Researchers in the past dealt with the optimization problem relating to deterministic opportunistic replacement problem. Complete solutions were obtained for a two component situation for both finite and infinite time horizon. For the multi-component opportunistic replacements with fixed time horizon, a mixed integer linear programming formulation is given in the literature. In this paper, a simplified alternative approach to solving the two-component problem is given.

A Dynamic Programming approach to solve the two-component problem which can be extended to K-component situation is also discussed. The mixed integer programming formulation is modified and computational advantages are discussed.

* Professor and Director, SDM Institute for Management Development, Mysore, INDIA

** Professor, Indian Institute of Management Bangalore, INDIA

A Solution Procedure for the Multi-Component Deterministic Opportunistic Replacement Problem

by

A.K. Rao
M.R. Rao

Scope and Purpose

Opportunistic replacement problem considers a system consisting of failing components which incur extensive maintenance costs upon failure. These costs relate to shutting down and disassembly of the entire system. When the system is disassembled for regular maintenance, it is possible to replace components at no additional maintenance cost. Mathematical techniques are used to find a replacement policy which trades off the remaining useful life of a still operational component in exchange for avoiding the high maintenance cost associated with component failure. The purpose of this paper is to develop a simple solution method for a two component deterministic opportunistic replacement problem.
For a finite horizon multicomponent deterministic situation, two different formulations, viz. a dynamic programming and a modified mixed integer programming formulation are suggested. Computational advantages and limitations are also discussed.

A Solution Procedure for the Multi-Component Deterministic Opportunistic Replacement Problem

by

A.K. Rao*
M.R. Rao**

1. INTRODUCTION

Earlier researchers dealt with the problem of Multi-Component Deterministic Opportunistic Replacement Problem [6]. The problem was originally introduced by Jorgenson and Radner [8] for stochastically failing components which incur extensive maintenance cost upon failure. An extension of the problem was studied by Epstein and Wilamowsky [3,4,5,6]. A new variation of the problem was introduced by George et.al [7] . They considered a purely deterministic opportunistic replacement problem. Epstein and Wilamowsky [6] made an analysis of the two component deterministic problem. They showed that for a two component problem with different life-limits, each individual scheduled replacement point offers potential opportunity for monetary saving. They proved that in this deterministic situation, only a limited number of the possible replacement points need be considered. An algorithm to generate these points was also given. Dickman, Epstein

* Professor and Director, SDM Institute for Management Development, Mysore, INDIA

** Professor, Indian Institute for Management Bangalore, INDIA

and Wilamowsky [1] presented a mixed integer linear programming formulation for any n-component system. However, the problem size becomes large.

In this paper, a simpler method for finding the optimal replacement point for the two component problem is given. A dynamic programming formulation of the problem for two component and three component situations are given. This can easily be extended to K-component situation.

2. PROBLEM FORMULATION FOR A 2 COMPONENT SITUATION

The formulation is as per [6] with some modification. Let us denote the two components by A and B.

Let

L_A : Assigned life limit for component A

L_B : Assigned life limit for component B

If $L_A = L_B$, then the solution is trivial.

Without loss of generality, assume that $L_A < L_B$.

It was observed by Epstein and Wilamowsky [6] that premature replacement, if warranted, is limited to scheduled B replacement points or scheduled A replacement points immediately preceding B replacement points. They also proved that only a fraction of the eligible points need be considered and presented an algorithm for their determination. In this paper the nature of the optimum solution is studied. It is shown that the time difference between a A-replacement point and the immediately following B-replacement point is crucial in the determination of the optimal joint replacement at a A-replacement point. Similarly it is shown that the time difference between a B-replacement point and the immediately following A-replacement point is crucial in the determination of the optimal joint replacement at a B-replacement point. This reduces to a comparison of a few ratios to determine the optimal replacement point if the joint replacement is at a A-replacement point. Similarly a comparison of few ratios will give the optimal replacement point if the joint replacement is at a B-replacement point. The least cost associated with these optimal points will give the optimal policy.

Let

L = L.C.M of (L_A , L_B) and

$N_A = L/L_B$ and $N_B = L/L_A$.

Then, N_A and N_B are relatively prime.

Also,

Total cycle time = L

No. of A's in cycle = N_B

No. of B's in cycle = N_A

Total No. of replacement points = $N_A + N_B - 1$.

It is assumed that both A and B are installed at time 0 and this is not included in the count of the number of replacements. This is because we are considering an infinite horizon problem.

If L_B is an integral multiple of L_A , then the solution is obviously trivial.

Let r be a positive integer such that

$$r\, L_A < L_B < (r+1)\, L_A.$$

Let

C_A : Cost of replacing a single A

C_B : Cost of replacing a single B

C : Cost of disassembly for single or double replacement

x : No. of A's used from the start of the cycle ($1 \leq x \leq N_B$). The component that we start at time 0 is not counted.

y : No. of B's used from the start of the cycle($1 \leq y \leq N_A$). The component that we start at time 0 is not counted.

ΔT_x : Time differential between the x^{th} A and the next B, ($0 \leq \Delta T_x < L_B$). Note that ΔT_x is 0 only when $x=N_B$

ΔT_y : Time differential between the y^{th} B and the next A, ($0 \leq \Delta T_y < L_A$). Note that ΔT_y is 0 only when $y=N_A$

$f_1(x)$: the y value that immediately follows x and equal to ($x L_A + \Delta T_x$) / L_B . Obviously

$f_1(x) = \lfloor x L_A / L_B \rfloor + 1$ for $x < N_B$ where

$\lfloor \ \rfloor$ indicates the integral part. For $x=N_B$ $f_1(x)=N_A$. Note that $\Delta T_x > 0$ for $x < N_B$ and $\Delta T_x = 0$ for $x = N_B$

$f_2(y)$: the x value that immediately follows y and equal to ($y L_B + \Delta T_y$) / L_A . Obviously

$f_2(y) = \lfloor y L_B / L_A \rfloor + 1$ for $y < N_A$ and $f_2(y) = N_B$ for $y = N_A$. Note that $\Delta T_y > 0$ for $y < N_A$ and $\Delta T_y = 0$ for $y = N_A$

C(x) : cost per unit time for a double removal at x^{th} A and equal to

$$(x C_A + f_1(x) C_B + [x + f_1(x) - 1] C)/(x L_A)$$

D(y) : cost per unit time for a double removal at y^{th} B and equal to

$$(y C_B + f_2(y) C_A + [y + f_2(y) - 1] C)/(y L_B)$$

Here we can consider two alternative objectives. The first one is to minimize the discounted cost of replacements over an infinite time horizon. The other one which was dealt by Epstein and Wilamowsky [6] is to minimize the average replacement cost per unit time without discounting. Here we consider the second alternative.

Since every double removal initiates an identical cycle, the replacement point yielding the minimum cost per unit time within a single cycle is the optimal replacement point and determines the overall cost per unit time for the entire system. Here we are not discounting the cost. The objective is to find the x or y that yields the minimum of all possible C(x) and D(y) values. Epstein and Wilamowsky [6] detailed a method of reducing the number of possible optimal points and arrive at the optimal by comparing the costs at these possible optimal points. An alternative method of solution which is simpler, is given below.

3. ALTERNATIVE METHOD OF SOLUTION

As already mentioned, if L_B is an integral multiple of L_A, then the solution is trivial. Hence we assume that L_B is not an integral multiple of L_A .

In section 3.1, we find the conditions for local optimum for C(x) and D(y). In section 3.2, we find the conditions for the global optimum for C(x) and D(y). Clearly, the minimum of these two global optimums will be the optimum solution for the problem.

The cost function $C(x)$ is

$$(x\ C_A + f_1(x)\ C_B + [x + f_1(x) - 1]\ C)\ /(x\ L_A)$$

$$= ((C+C_A)/L_A) + P(x) \qquad \text{where}$$

$$P(x) = ((C+C_B)/L_A)\ (f_1(x)/x) - (C/L_A)\ /x$$

The cost function $D(y)$ is

$$(y\ C_B + f_2(y)\ C_A + [y + f_2(y) - 1]\ C)\ /\ (y\ L_B)$$

$$= ((C+C_B)/L_B) + Q(y) \qquad \text{where}$$

$$Q(y) = ((C+C_A)/L_B)\ (f_2(y)/y) - (C/L_B)\ /y$$

3.1 CONDITIONS FOR LOCAL OPTIMUM

The necessary and sufficient conditions for a point x such that $1 < x < N_B$ to be a local optimum for $C(x)$ are

$$P(x+1) - P(x) \geq 0 \text{ and}$$

$$P(x-1) - P(x) \geq 0.$$

$$P(x+1)-P(x) = \frac{(C+C_B)}{L_A}\left[\frac{f_1(x+1)}{x+1} - \frac{f_1(x)}{x}\right] - \frac{C}{L_A}\left[\frac{1}{x+1} - \frac{1}{x}\right]$$

$$= \frac{C+C_B}{L_A x(x+1)}\left[x f_1(x+1) - (x+1) f_1(x) + \frac{C}{C+C_B}\right]$$

≥ 0 if and only if $x\ f_1(x+1) - (x+1)\ f_1(x) \geq 0$

since x and $f_1(x)$ are integers and $C/(C+C_B)$ is positive and less than 1 .

Similarly, it follows that

$$P(x\ 1)\ P(x)\ \ \frac{(C+C_B)}{L_A}\left[\frac{f_1(x\ 1)}{x\ 1}\ \ \frac{f_1(x)}{x}\right]\ \ \frac{C}{L_A}\left[\frac{1}{x\ 1}\ \ \frac{1}{x}\right]$$

$$\frac{C+C_B}{L_A x(x\ 1)}\left[x\ f_1(x\ 1)\ \ (x\ 1)\ f_1(x)\ \ \frac{C}{C+C_B}\right]$$

≥ 0 if and only if $x\ f_1(x\ 1) - (x-1)\ f_1(x) \geq 1$

Thus the necessary and sufficient conditions for the occurrence of a local optimum at x where $1 < x < N_B$ are

$$x\, f_1(x+1) - (x+1)\, f_1(x) \geq 0 \qquad (1)$$

$$x\, f_1(x-1) - (x-1)\, f_1(x) \geq 1. \qquad (2)$$

Note: Since $L_A < L_B$, there can be utmost one B-replacement point between two successive A-replacement points. Hence, $0 \leq f_1(x+1) - f_1(x) \leq 1$ and $0 \leq f_1(x) - f_1(x-1) \leq 1$.

Now we will find the conditions for local optimality for the end points $x = 1$ and $x = N_B$.

A local optimum at $x = 1$ occurs if and only if $P(2) - P(1) \geq 0$. This gives us the condition

$$1\, f_1(2) - 2\, f_1(1) \geq 0,$$

i.e. $f_1(2) \geq 2$ as $f_1(1) = 1$ for $L_A < L_B$.

If $L_A < (1/2)\, L_B$, then $f_1(2) = 1$ and hence a local optimum cannot occur at x=1. If $L_A > (1/2)\, L_B$, then a local optimum will occur at $x = 1$.

A local optimum at $x = N_B$ occurs if and only if $P(N_B-1) - P(N_B) \geq 0$.

This gives us the condition

$$N_B\, f_1(N_B-1) - (N_B-1) f_1(N_B) \geq 1.$$

This condition is always satisfied since

$$f_1(N_B-1)=f_1(N_B)=N_A \text{ for } L_A < L_B$$

Thus a local optimum always occurs at $x = N_B$.

Result 1:

For $1 < x < N_B$, necessary and sufficient conditions for a local optimum at x are

$$f_1(x+1) = f_1(x) + 1 \text{ and}$$

$$f_1(x-1) = f_1(x)$$

Proof: Conditions for local optimality are given in equations (1) and (2). Condition (1) is

$$f_1(x+1) \geq ((x+1)/x)\ f_1(x) > f_1(x) \text{ as } ((x+1)/x) > 1.$$

As $f_1(x+1)$ is either $f_1(x)$ or $f_1(x)+1$, necessarily

$$f_1(x+1) = f_1(x)+1 \qquad (3)$$

The second condition for local optimality given by equation (2) can be written as

$$x\ \{f_1(x-1) - f_1(x)\} + f_1(x) - 1 \geq 0. \qquad (3a)$$

Note that $f_1(x) = \lfloor x L_A / L_B \rfloor + 1 < x + 1$ as $L_A < L_B$.

Also, $f_1(x)$ is either $f_1(x-1)$ or $f_1(x-1) + 1$.

If $f_1(x)= f_1(x-1) + 1$, then the above inequality (3a) gives

$$- x + f_1(x) - 1 \geq 0,$$

i.e. $f_1(x) \geq x+1$ which is a contradiction. Hence,

$$f_1(x) = f_1(x-1) \qquad (4)$$

These conditions for optimal x are diagrammatically represented below:

```
(f1(x)-1)LB                      f1(x)LB                   (f1(x)+1)LB
-----|---|-------------|------|---------|-------------|----
       (x-1)LA          xLA              (x+1)LA
```

Between two B-replacement points, the last A-replacement point is a candidate provided there are at least two A-replacement points in that interval.

The global minimum for C(x) will be among the A-replacement points x which satisfy the above conditions.

A similar analysis for finding the conditions for local optimum of D(y) shows that

i. a local optimum at y=1 occurs if and only if

$(L_A/2) < L_B - r\, L_A$

ii. a local optimum always occurs at $y = N_A$

iii. For $1 < y < N_A$, a necessary and sufficient condition for a local optimum at y is

$f_2(y+1) = f_2(y) + r + 1$ and

$f_2(y-1) = f_2(y) - r$

For detailed derivation, the reader may refer to Rao and Rao [9].

The global minimum for D(y) will be among the B- replacement points y which satisfy the above conditions.

3.2 CONDITIONS FOR GLOBAL OPTIMUM

Conditions for Global Optimum for C(x):

Let x_1 and x_2 be two local optimal for C(x) such that

$$1 \leq x_1 < x_2 \leq N_B.$$

Then if $C(x_1) - C(x_2) \geq 0$, obviously we can drop point x_1 from consideration. This condition after simplification reduces to

$$\frac{x_2\, f_1(x_1) - x_1 f_1(x_2)}{x_2 - x_1} \geq \frac{C}{C + C_B}$$

Similarly, if

$$\frac{x_2\, f_1(x_1) - x_1 f_1(x_2)}{x_2 - x_1} < \frac{C}{C + C_B}$$

then, we can drop x_2 from consideration of global optimal. Thus, for any sequence of points x such that $1 < x < N_B$ which satisfy the condition

$$(f_1(x)-1)L_B < (x-1)\, L_A < x\, L_A < f_1(x)\, L_B < (x+1)\, L_A$$

and the end points $x = 1$ and $x = N_B$, we need to compare the successive points x which satisfy the condition for local optimal, the quantity,

$$(x_2\, f_1(x_1) - x_1\, f_1(x_2))/(x_2 - x_1) \text{ with } C/(C+C_B)$$

and then select one of the points. Note that $x=N_B$ will always be selected and $x=1$ will be dropped from consideration if $L_A < (1/2)\, L_B$. This will lead us to the global minimum of C(x). The comparison can still be simplified as shown below:

Let $C^* = C/(C+C_B)$. We drop x_1 from consideration if

$$\frac{x_2\, f_1(x_1) - x_1 f_1(x_2)}{x_2 - x_1} \geq C^*$$

As $f_1(x) = (\, x\, L_A + \Delta T_x\,)\, /\, L_B$, the above condition for dropping x_1 from consideration is

$$\frac{(\Delta T_{x1}/L_B) - C^*}{x_1} \geq \frac{(\Delta T_{x2}/L_B) - C^*}{x_2}$$

This reduces to finding the minimum of

$$\frac{(\Delta T_{x1}/L_B) - C^*}{x_1}$$

Suppose the minimum is attained at $x=x^*$. Then $C(x^*)$ is the global minimum for C(X).

Conditions for global optimum for D(y):

Similar analysis could be done to find the global optimum for D(y) as given in [9]. It is shown that to find the global minimum of D(y) we need to find the minimum of

$$\frac{(\Delta T_{yi}/L_A) - C^{**}}{y_i}$$

where $C^{**} = C/(C+C_A)$.

Suppose the minimum occurs at $y=y^*$. Then $D(y^*)$ is the minimum.

The minimum of $C(x^*)$ and $D(y^*)$ is the optimal double replacement point.

4. EXAMPLE

This method of solution is applied to the example given by Epstein and Wilamowsky [6] They illustrated the algorithm with four examples taking $L_A = 7$ and $L_B = 11$. The relevant costs taken for the 4 examples and the optimal solutions obtained are given below:

Example	C_A	C_B	C	Optimal
i	1	1	10	at 7 (x= 1)
ii	1	10	3	at 11 (y=11)
iii	10	2	1	at 21 (x= 3)
iv	2	10	1	at 77 (x=11,y=7)

Using the same values for the parameters , the above simple procedure is applied and the results obtained are tabulated below:

Example no.	C^*	C^{**}	x^*	y^*	$C(x^*)$	$C(y^*)$	Optimal
i	10/11	10/11	1	1	12/7	23/11	12/7 at $x^*=1$
ii	3/13	3/4	3	1	35/21	18/11	18/11 at $y^*=1$
iii	1/3	1/11	3	5	38/21	102/55	38/21 at $x^*=3$
iv	1/11	1/3	11	7	109/77	109/77	109/77 at $x^*=11$ and $y^*=7$

5. DYNAMIC PROGRAMMING FORMULATION

For a K component situation with a finite planning horizon T, the problem can be formulated as a dynamic programming problem. Assume that the revenue or cost accrued from components which still have a useful life at the end of the planning period T is 0. The approach to Dynamic Programming formulation essentially remains the same even if the revenue accrued is not 0.

Since the planning horizon T is fixed, minimizing the average cost per unit time is equivalent to minimizing the total cost for T periods. The Dynamic programming and Integer programming formulations given in this section and the next section minimize the total cost for T periods. The formulations in this section and the next section can easily be adapted to minimize the discounted cost over the planning horizon T.

Let the periods be numbered 1,2,3,...,T.

We will formulate a two component situation and indicate the changes necessary for the three component situation.

This can easily be extended to a K- component situation. For a two component situation, let n_1 and n_2 stand for the elapsed lives of components A and B at the end of a period. In the Dynamic Programming formulation, the stages are the periods and the states are the elapsed lives of components A and B. At the end of any period, if the elapsed lives of both the components are strictly less than their useful lives, then we do not replace any components. We replace one or both only when at least one of the components reaches the end of its useful life.

Let us define

$f_j(n_1, n_2) =$ minimum cost of the optimum policy when the system is in state (n_1, n_2) and there are j more periods to go; $j = 0, 1, 2, \ldots, T$.

The initial conditions are

$f_0(n_1, n_2) = 0$ for all n_1 and n_2.

The recursive equation for $j=1,2,\ldots,(T-1)$ is

$$
\begin{aligned}
f_{j+1}(n_1, n_2) &= f_j(n_1+1, n_2+1) && \text{if } n_1 < L_A \text{ and } n_2 < L_B \\
&= \text{Min}\{ C + C_A + f_j(1, n_2+1),\ C + C_A + C_B + f_j(1,1) \} && \text{if } n_1 = L_A \text{ and } n_2 < L_B \\
&= \text{Min}\{ C + C_B + f_j(n_1+1, 1),\ C + C_A + C_B + f_j(1,1) \} && \text{if } n_1 < L_A \text{ and } n_2 = L_B \\
&= C + C_A + C_B + f_j(1,1) && \text{if } n_1 = L_A \text{ and } n_2 = L_B
\end{aligned}
$$

$f_T(L_A, L_B)$ gives the minimum cost.

A three component situation can be formulated in a similar way. In this case we have to consider the following options:

At the end of any period we do not replace any components if the elapsed lives of all three components are strictly less than their useful lives. If exactly one of the components reach the end of its useful life at the end of a period, then we have to consider the following three options:

i. replace only that component which reached its useful life
ii. replace in addition, one more component which has not yet reached its useful life; there are two possibilities in this case depending on the component (that has not yet reached its useful life) chosen for replacement
iii. replace all three components

If exactly two components reach the end of their useful lives at the end of a period, then the options are replace only these two components, or all the three components. If all three components reach their useful lives at the end of a period, then we have to replace all three components.

For the K-component case also the number of stages is equal to T which is the number periods. However, the number of possible states at each stage is given by

$$\prod_{i=1}^{K} L_i \quad \text{where } L_i \text{ is the life of the component } i$$

For each possible state, the recursive equation will require the minimum of at most 2^K-1 values. Thus, if K is at most say 15, and if the number of states is not very large, the dynamic programming formulation can be applied. We expect this to be the case for most practical problems.

6. INTEGER PROGRAMMING FORMULATION

In order to formulate the problem as an integer programming problem, the following notation is used:

Let K = number of components

$T+1$ = number of periods

C_j = cost of replacing a single component j; $j = 1,2,\ldots,K$

C_o = maintenance cost for replacing one or more components

L_j = life of component j; $j = 1,2,\ldots,K$ assumed to be an integer

Define

X_{ij} = 1 if component j is replaced at the end of period i
= 0 otherwise

Y_i = 1 if there is any replacement at the end of period i
= 0 otherwise

Note that no replacement is required at the end of period $T+1$.

Now the integer programming formulation as given by Dickman et al [1] is

$$\text{Minimize} \sum_{j=1}^{K} \sum_{i=1}^{T} C_j X_{ij} + \sum_{i=1}^{T} C_o Y_i$$

subject to

$$\sum_{k=i}^{i+L_j-1} X_{kj} \geq 1 \quad ; \; i = 1,2,\ldots,T - L_j + 1, \quad j = 1,2,\ldots, K$$

$$\sum_{j=1}^{K} X_{ij} - K Y_i \leq 0 \; ; \; i = 1,2,\ldots, T \qquad (5)$$

$$Y_i = 0 \text{ or } 1 \; ; \; i = 1, 2,\ldots, T$$

$$0 \leq X_{ij} \leq 1 \qquad \text{for all } (i,j).$$

Some simplifications to this formulation are suggested in [1]. For instance, there will be no replacement at the end of periods which are not non-negative integer linear combinations of L_j ; $j = 1,2,\ldots,K$. For given instances of the problem, this may reduce the number of variables and constraints considerably. But as pointed out in [1], if $K = 3$, $L_1 = 3$, $L_2 = 4$ and $L_3 = 5$, then all periods from 3 to T are potential replacement periods. In this case, clearly $X_{ij} = 0$, $i = 1,2$ and $j = 1,2,3$; $X_{31} = 1$; $Y_i = 0$ for $i=1,2$ and $Y_3 = 1$. If $T = 100$, there will be , not including fixed variables, 293 continuous variables, 97 integer 0-1 variables. In this case, there will be 385 constraints, not counting the redundant constraints.

Constraint set (5), together with the objective function coefficient of Y_i , is a compact way of ensuring that the maintenance cost for replacement is incurred in period i if any one of the components is replaced in that period. But from a computational point of view, it is better to replace constraint set (5) by

$$X_{ij} - Y_i \leq 0 \; ; \quad i = 1,2,\ldots,T \qquad (6)$$
$$j = 1,2,\ldots,K$$

This increases the number of constraints by (K-1)T. But, these constraints are strong inequalities and the linear programming bound obtained by using constraints (6) is typically much better than the linear programming bound obtained by using (5).

7. COMPUTATIONAL RESULTS

Several finite time horizon problems were solved by dynamic programming as well as by integer linear programming. In

all 42 problems were solved using dynamic programming and 10 problems were solved using integer linear programming. Table 1 gives the objective function value and the time in seconds for a 3 component situation. C is the cost of disassembly for single, double and multiple replacement and C_i is the cost of replacing a single component i; i=1,2,3. L_i is the assigned life limit for the ith component; i=1,2,3. T stands for the time horizon.
Ten problems were solved by integer linear programming using the formulation suggested by Dickman, Epstein, and Wilamowky (DEW) and using **our fsirmgl2XThe** software used was HYPER LINDO .
Table 2 gives for ten sample problems

i. the total time
ii. the number of pivots required to solve the problems by each of the methods
iii. the optimal objective function values
iv. number of pivots that were completed when the integer solution that was obtained is actually optimal but not certified to be so
v. the objective function value (LP lower bound) obtained by solving the linear programming relaxation
vi. the number of pivots required to solve the LP relaxation
vii. the objective function value of the best integer solution (IP upper bound) found while solving the LP problem

viii. the number of pivots completed when the best IP solution was found.

A study of the tables shows that the three component problem can be solved efficiently by dynamic programming. For this study, the values of L_1, L_2, and L_3 are taken as 3, 4, and 5 respectively. The time periods are taken as 22, 27, and 32. Problems are solved for various scenarios assigning various values to the cost parameters C, C_1, C_2, and C_3 .

The time taken to arrive at an optimal solution is consistently lower for dynamic programming formulation compared to integer programming formulation. For example, problem number 1 considered T equal to 22, the D.P. solution is obtained in 25 seconds whereas the D.E.W formulation has taken 34 minutes and R.R. formulation has taken 2 minutes. For prblem 10, the D.P. solution is obtained **in 27** secondshe D.E.W formulation has taken more than 150 minutes and R.R. formulation has taken 7 minutes.

Our formulation of the integer programming problem is more efficient than the DEW formulation.

Table 1

DYNAMIC PROGRAMMING : COMPUTATIONAL SUMMARY

LIVES OF COMPONENTS: $L_1 = 3$, $L_2 = 4$, $L_3 = 5$.

P.No.	C_1	C_2	C_3	C	T	OBJ. Value	TIME (SEC)
1	1	2	3	4	22	64	25
2	1	2	3	2.5	22	52	25
3	1	2	3	0.5	22	34.5	25
4	3	1	2	4	22	68	25
5	3	1	2	2.5	22	57.5	25
6	3	1	2	0.5	22	39.5	25
7	2	3	1	4	22	69	25
8	2	3	1	2.5	22	58	25
9	2	3	1	0.5	22	38.5	30
10	1	2	3	4	27	81	30
11	1	2	3	2.5	27	66	30
12	1	2	3	0.5	27	42.5	30
13	1	2	3	4	32	96	35
14	1	2	3	2.5	32	78	35
15	1	2	3	0.5	32	52	35
16	1	2	3	4	50	155	40
17	1	2	3	2.5	50	126.5	40
18	1	2	3	0.5	50	83.5	40
19	3	1	2	4	50	160	40
20	3	1	2	2.5	50	136	40
21	3	1	2	0.5	50	93.5	40
22	2	3	1	4	50	160	40
23	2	3	1	2.5	50	136	40
24	2	3	1	0.5	50	91.5	40
25	1	2	3	4	100	315	80
26	1	2	3	2.5	100	256.5	80
27	1	2	3	0.5	100	169.5	80

contd..

Table 1 (Contd)

DYNAMIC PROGRAMMING : COMPUTATIONAL SUMMARY
LIVES OF COMPONENTS: $L_1 = 3$, $L_2 = 4$, $L_3 = 5$.

P.No.	C_1	C_2	C_3	C	T	OBJ. Value	TIME (SEC)
28	3	1	2	4	100	328	80
29	3	1	2	2.5	100	278.5	80
30	3	1	2	0.5	100	191	80
31	2	3	1	4	100	329	80
32	2	3	1	2.5	100	279	80
33	2	3	1	0.5	100	188	80
34	1	2	3	4	50	95	105
35	1	2	3	2.5	50	79.5	105
36	1	2	3	0.5	50	54.5	105
37	3	1	2	4	50	97	105
38	3	1	2	2.5	50	82	105
39	3	1	2	0.5	50	59.5	105
40	2	3	1	4	50	96	105
41	2	3	1	2.5	50	81	105
42	2	3	1	0.5	50	58.5	105

Table 2

INTEGER PROGRAMMING : COMPUTATIONAL SUMMARY

LIVES OF COMPONENTS: $L_1 = 3$, $L_2 = 4$, $L_3 = 5$.

P.No.	T	D.E.W TOTAL TIME(MIN) (PIVOTS)	R.R TOTAL TIME(MIN) (PIVOTS)	D.E.W OBJ. VALUE (PIVOTS)	R.R OBJ. VALUE (PIVOTS)	D.E.W LP LOWER BOUND (PIVOTS)	R.R LP LOWER BOUND (PIVOTS)	D.E.W IP UPPER BOUND (PIVOTS)	R.R IP UPPER BOUND (PIVOTS)
1	22	34 (25843)	2 (577)	64 (15889)	64 (247)	50.33 (79)	59.33 (135)	None	71 (79)
2	22	33 (29878)	3 (1135)	52 (10263)	52 (674)	42.33 (79)	48.33 (159)	None	56.5 (78)
3	22	12 (9931)	1 (410)	34.5 (674)	34.5 (235)	31.66 (78)	33.66 (146)	None	36.5 (97)
4	22	14 (12440)	2 (739)	68 (1239)	68 (106)	55.33 (72)	64.5 (142)	None	68 (106)
10	27	> 150 (>15,587)	7 (3288)	81 (15,587)	81 (886)	62.66 (106)	73.33 (188)	None	88 (100)
11	27	---	10 (4223)	---	66 (468)	52.66 (102)	59.83 (177)	None	68 (112)
12	27	---	1 (349)	---	42.5 (257)	39.33 (114)	41.83 (166)	None	46.5 (123)
13	32	---	18 (5913)	---	96 (603)	76 (112)	87.25 (232)	None	98 (153)
14	32	---	20 (6599)	---	78 (603)	64 (121)	71.33 (242)	None	86.5 (115)
15	32	---	14 (4524)	---	52 (2587)	48 (1120)	50 (227)	None	53.5 (163)

REFERENCES

1. Bernard Dickman, S. Epstein and Y. Wilamowsky, A Mixed Integer Linear Programming Formulation for Multi-Component Deterministic Opportunistic Replacement, OPSEARCH, Vol. 28, No. 3 (1991).

2. S. Epstein and Y. Wilamowsky, A Statistical Model for Jet Engine Maintenance, Presented at the Annual Meeting of the American Statistical Association, San Diego, CA(1978).

3. S. Epstein and Y. Wilamowsky, A Heuristic - Dynamic Programming Replacement Model, Presented at the Annual Meeting of TIMS, Honolulu, HA(1979).

4. S. Epstein and Y. Wilamowsky. A Disk Replacement Policy for Jet Engines, Ann.Soc.Logist.Engnrs. 5,35-36 (1980).

5. S. Epstein and Y. Wilamowsky, A Replacement Schedule for Multicomponent Life - Limited parts, Naval Re. Logist, Q.29, 685-692 (1982).

6. Epstein, S. and Y. Wilamowsky, Opportunistic Replacement in a Deterministic Environment, Computers and Operation Research, Vol 12, No. 3 ((1985).

7. L.L. George and J.A. Day, Opportunistic Replacement of Fusion Power System Parts, Presented at Reliability and Maintainability Symposium, Los Angeles, CA (1982).

8. Jorgenson D.W. and R. Radner, Optimal Replacement and Inspection of Stochastically Failing Equipment, Rand, Paper P-2074 (1960).

9. Rao A.K and M.R. Rao, A Note on Multi-Component Deterministic Opportunistic Replacemnent Problem, Working Paper No. 80, Indian Institute of Management Bangalore, Bangalore, India, May 1996.