# A Note on the Multi-Component Deterministic Opportunistic Replacement

**By**

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

**May 1996**

Please address all correspondence to:

A.K. Rao /M R. Rao
Professor
Indian Institute of Management
Bannerghatta Road
Bangalore 560 076
India

Fax (080) 6644050

# A Note on the Multi-Component Deterministic Opportunistic Replacement

by

A.K. Rao
M.R. Rao
Indian Institute of Management Bangalore

## 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, an alternative approach to solving the two-component problem is given. A Dyanamic 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 compututaional advantages are dicussed.

# A Note on the Multi-Component Deterministic Opportunistic Replacement

by

A.K. Rao
M.R. Rao
Indian Institute of Management Bangalore

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

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

## 2. PROBLEM FORMULATION FOR A 2 COMPONENT SITUATION

The formulation is as per [ 6 ] with slight 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

Assume that $L_A < L_B$ .

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. aAlso,

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 = $N_A + N_B - 1$.

Let r be a positve integer such that

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

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 ( $1 \leq x \leq N_B$)

$y$ : No. of B's used from the start of the cycle( $1 \leq y \leq N_A$)

$\Delta T_x$ : Time differential between the x'th A and the next B, ( $1 \leq \Delta T_x \leq L_B$ )

$\Delta T_y$ : Time differential between the y'th B and the next A, ( $1 \leq \Delta T_y \leq L_A$ )

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

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

$C(x)$ : cost per unit time for a double removal at x'th A and equal to

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

$C(y)$ : cost per unit time for a double removal at y'th B and equal to

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

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. The objective is to find the x or y that yields the minimum of all possible C(x) and C(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 seems simpler, is given below.

## 3. ALTERNATIVE METHOD OF SOLUTION

We first find the condition for local optimum for C(x) and then find the condition for global optimum. Similar procedure will be followed for finding the condition for global optimum for $C(y)$. The minimum of these two global optimums will give the optimum for the problem.

The cost function $C(x)$ is

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

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

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

For any integer x, let us define an integer $K(x)$ such that the following inequality is satisfied:

$$(\ K(x) - 1\ )\ L_B < x\ L_A < K(x)\ L_B$$

It is obvious that $K(x) = f_1(x)$. Thus,

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

For any local optimal x,

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

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

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

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

$$\geq 0 \quad \text{if} \quad x\ K(x+1) - (x+1)\ K(x) \geq 0$$

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

Similarly, it follows that

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

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

$$\geq 0 \quad \text{if} \quad x\ K(x-1) - (x-1)\ K(x) \geq 1$$

Thus for local optimum , x should satisfy

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

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

**Note:** Obviously, $K(x+1) \geq K(x)$ and

$\{ K(x+1) - K(x) \}\, L_B \leq L_B$.

Hence, $0 \leq K(x+1) - K(x) \leq 1$.

Substituting $x-1$ for $x$, we get

$0 \leq K(x) - K(x-1) \leq 1$.

**Result 1:**

If x is a local optimum, then $K(x+1) = K(x) + 1$.

Proof:

Suppose for a local optimum x , $K(x+1) = K(x)$. Then, condition (1) becomes

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

i.e. $-K(x) \geq 0$ which cannot happen as $L_A < L_B$. Hence,

$$K(x+1) = K(x) + 1. \qquad (3)$$

**Result 2:**

If x is a local optimal, then $K(x-1) = K(x)$.

Proof:

Suppose for a local optimum x, $K(x-1) = K(x)-1$. Then, condition (2) becomes

$$x\,(K(x)-1) - (x-1)\,K(x) \geq 1$$

i.e. $-x + K(x) \geq 1$.

But, $( K(x) - 1 )\, L_B < x\,L_A < x\,L_B$

Hence, $-x + K(x) < 1$, which is a contradiction. Thus,

$$K(x-1) = K(x). \qquad (4)$$

These conditions for optimal x are diagrammatically represented below:

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

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

**Condition for global optimum of C(x):**

Let $x_1$ and $x_2$ be two local optimal for C(x) with $x_1 < x_2$. Then if $C(x_1) - C(x_2) \geq 0$, then obviously we can drop point $x_1$ from consideration. This condition after simplification reduces to

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

Similarly, if

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

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

$$(K(x)-1)L_B < (x-1) L_A < x L_A < K(x) L_B < (x+1) L_A$$

we need to compare the successive points x, the quantity, $(x_2 K(x_1) - x_1 K(x_2))/(x_2 - x_1)$ with $C/(C+C_B)$ and then select one of the points. This will lead us to the minimum of C(x).

Similar analysis is done for the cost function $C(y)$ as shown below:

The cost function $C(y)$ is

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

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

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

For any integer y, let us define an integer $L(y)$ such that the following inequality is satisfied:

$$(\ L(y) - 1\ )\ L_A < y\ L_B < L(y)\ L_A$$

It is obvious that $L(y) = f_2(y)$. Thus,

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

For any local optimum x,

$$Q(y+1) - Q(y) \geq 0 \text{ and}$$

$$Q(y-1) - Q(y) \geq 0.$$

$$Q(y+1) - Q(y) = \frac{(C+C_A)}{L_B}\left[\frac{L(y+1)}{y+1} - \frac{L(y)}{y}\right] - \frac{C}{L_B}\left[\frac{1}{y+1} - \frac{1}{y}\right]$$

$$= \frac{C+C_A}{L_B y\,(y+1)}\left[y\ L(y+1) - (y+1)\ L(y) + \frac{C}{C+C_A}\right]$$

$$\geq 0 \text{ if } y\,L(y+1) - (y+1)\ L(y) \geq 0$$

( Since y and $L(y)$ are integers and $C/(C+C_A)$ is positive and less than 1 ).

Similarly, it can be shown that

$$Q(y-1) - Q(y) = \frac{(C+C_A)}{L_B}\left[\frac{L(y-1)}{y-1} - \frac{L(y)}{y}\right] - \frac{C}{L_B}\left[\frac{1}{y-1} - \frac{1}{y}\right]$$

$$= \frac{C+C_A}{L_B y\,(y-1)}\left[y\ L(y-1) - (y-1)\ L(y) - \frac{C}{C+C_A}\right]$$

$$\geq 0 \text{ if } y\,L(y-1) - (y-1)\ L(y) \geq 1$$

Thus for local optimum , y should satisfy

$$y\,L(y+1) - (y+1)\,L(y) \geq 0 \tag{5}$$

$$y\,L(y-1) - (y-1)\,L(y) \geq 1 \tag{6}$$

**Note:** For any y, we have

$\{ L(y+1) - 1 \}\, L_A - L(y)\, L_A < L_B$ and

$L(y+1)\, L_A - \{ L(y) - 1\}\, L_A > L_B$, we have

$(r-1) \leq L(y+1) - L(y) \leq r+1.$

Hence, L(y+1) can take one of the values L(y)+ r or L(y)+ r+ 1.

Substtuting y-1 for y we conclude that L(y-1) can take one of the values L(y) - r or L(y) - r - 1.

**Result 3:**

If y is a local optimum, then $L(y+1) = L(y) + r + 1$.

Proof:

Suppose $L(y+1) = L(y) + r$.

Then condition (5) for local optimality is

$$y\,\{ L(y) + r \} - (y+1)\,L(y) \geq 0$$

i.e. $y\,r - L(y) \geq 0$.

But $r \leq L_B/ L_A$ . Hence

$$y\,r \leq y\,L_B/ L_A < L(y).$$

i.e. $y\,r - L(y) < 0$ and hence y cannot be a local optimum.

Hence, $L(y+1) = L(y) + r + 1$.

**Result 4:**

If y is a local optimum, then $L(y-1) = L(y) - r$.

Proof:

Suppose $L(y-1) = L(y) - r - 1$. Then condition (6) for local optimum is

$$y\{L(y) - r - 1\} - (y-1)\,L(y) \geq 1$$

i.e. $L(y) - (r+1)\,y \geq 1$.

But $r+1 > L_B/L_A$ and hence

$y(r+1) > y\,L_B/L_A > \{L(y) - 1\}\,L_A / L_A = L(y) - 1$.

i.e. $L(y) - (r+1)\,y < 1$ which is a contradiction.

Hence, $L(y-1) = L(y) - r$.

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

**Condition for global optimum of C(y):**

Let $y_1$ and $y_2$ be two local optimal for C(y) with $y_1 < y_2$. Then if $C(y_1) - C(y_2) \geq 0$, we can drop point $y_1$ from consideration. This condition after simplification reduces to

$$\frac{y_2\,L(y_1) - y_1 L(y_2)}{y_2 - y_1} \geq \frac{C}{C + C_A}$$

Similarly, if

$$\frac{y_2\,L(y_1) - y_1 L(y_2)}{y_2 - y_1} < \frac{C}{C + C_A}$$

we can drop $y_2$ from consideration of global optimal.

Thus, for any sequence of points y which satisfy the condition

$$(L(y)-r-1)L_A < (y-1)\, L_B < (L(y)-r)\, L_A \leq$$
$$(L(y)-1)\, L_A < y\, L_B < L(y)\, L_A < (L(y)+r)\, L_B <$$
$$(y+1)\, L_B < (L(y)+r+1)\, L_A$$

we need to compare the successive points y, the quantity, $(y_2\, L(y_1) - y_1\, L(y_2))/(y_2 - y_1)$ with $C/(C+C_A)$ and then select one of the points. This will lead us to the global minimum of C(y).

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

## 4. 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.

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

We will formulate a two 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 elasped 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; $j = 0, 1, 2, ...., T$.

The initial conditions are

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

The recursive equation is

$$
\begin{aligned}
f_{j+1}(n_1, n_2) &= f_j(n_{1+1}, n_{2+1}) \quad \text{if } n_1 < L_A \text{ and } n_2 < L_B \\
&= \text{Min}\{ C + C_A + f_j(1, n_2 + 1),\ C + C_A + C_B \\
&\qquad + f_j(1,1) \} \quad \text{if } n_1 = L_A \text{ and } n_2 < L_B \\
&= \text{Min}\{ C + C_B + f_j(n_1+1, 1),\ C + + C_A + C_B \\
&\qquad + f_j(1,1) \} \quad \text{if } n_1 < L_A \text{ and } n_2 = L_B \\
&= C + C_A + C_B + f_j(1,1) \\
&\qquad \text{if } n_1 = L_A \text{ and } n_2 = L_B
\end{aligned}
$$

This formulation can be extended to a K-component situation.

## 5. 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 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

$$X_{ij} = \begin{cases} 1 & \text{if component } j \text{ is replaced in period } i \\ 0 & \text{otherwise} \end{cases}$$

$$Y_i = \begin{cases} 1 & \text{if there is any replacement in period } i \\ = 0 & \text{otherwise} \end{cases}$$

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

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

subject to

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

$$\sum_{j=1}^{K} X_{ij} - n\, Y_i \leq 0 \; ; \; i = 1,2,\ldots, T \qquad (7)$$

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

$$X_{ij} \geq 0 \quad \text{for all } (i,j).$$

Some simplifications to this formulation are suggested in [ 1 ]. For instance, there will be no replacement in 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; $X_{31} = 1$; $Y_3 = 1$. If T = 100, there will be , not including the above fixed variables, 293 continuous variables, 97 integer 0-1 variables. In this case, there will be 385 constraints, not counting the redundant contraints. Costraint set (7), together with the objective function coeficient 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 (7) by

$$X_{ij} - Y_i \leq 0 \ ; \quad i = 1,2,\ldots,T \qquad (8)$$
$$j = 1,2,\ldots,K$$

This increases the number of constraints by (n-1)T. But, these constraints are strong inequalities and the linear programming bound obtained by using constraints (8) is typically much better than the linear programming bound obtained by using (7).

## 6. COMPUTIONAL 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 formulation (RR). The software used was HYPER LINDO

Table 2(a) gives the approximate total time and the number of pivots required to solve the problems by each of the methods. Table 2(b) gives the optimal objective function values. In addition, the same table gives the number of pivots that were completed when the integer solution that was obtained is actually optimal but not certified to be so. As mentioned earlier, the total number of pivots required to solve the entire problem is given in Table 2(a).

Table 2(c) gives the objective function value ( LP lower bound) obtained by solving the linear programming relaxation. In addition, the number of pivots required to solve the LP relaxation is also given in the same Table.

Table 2(d) gives the objective function value of the best integer solution ( IP upper bound ) found while solving the LP problem. The same Table gives 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. Our formulation of mixed integer programming 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 |

**Table 1 (Contd)**

**DYNAMIC PROGRAMMING : COMPUTATIONAL SUMMARY ( CONTD )**
**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) |
|---|---|---|---|---|---|---|---|
| 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 |
| 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 |

**Table 1 (Contd)**

**DYNAMIC PROGRAMMING : COMPUTATIONAL SUMMARY ( CONTD )**
**LIVES OF COMPONENTS: $L_1$ = 5, $L_2$ = 6, $L_3$ = 7.**

| P.No. | $C_1$ | $C_2$ | $C_3$ | C | T | OBJ. Value | TIME (SEC) |
|---|---|---|---|---|---|---|---|
| 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(a)

**INTEGER 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 | D.E.W TOTAL TIME(MIN) (PIVOTS) | R.R TOTAL TIME(MIN) (PIVOTS) |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 | 22 | 34 (25843) | 2 (577) |
| 2 | 1 | 2 | 3 | 2.5 | 22 | 33 (29878) | 3 (1135) |
| 3 | 1 | 2 | 3 | 0.5 | 22 | 12 (9931) | 1 (410) |
| 4 | 3 | 1 | 2 | 4 | 22 | 14 (12440) | 2 (739) |
| 10 | 1 | 2 | 3 | 4 | 27 | > 150 | 7 (3288) |
| 11 | 1 | 2 | 3 | 2.5 | 27 | --- | 10 (4223) |
| 12 | 1 | 2 | 3 | 0.5 | 27 | --- | 1 (349) |
| 13 | 1 | 2 | 3 | 4 | 32 | --- | 18 (5913) |
| 14 | 1 | 2 | 3 | 2.5 | 32 | --- | 20 (6599) |
| 15 | 1 | 2 | 3 | 0.5 | 32 | --- | 14 (4524) |

**Table 2(b)**

**INTEGER 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 | D.E.W OBJ. VALUE (PIVOTS) | R.R OBJ. VALUE (PIVOTS) |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 | 22 | 64 (15889) | 64 (247) |
| 2 | 1 | 2 | 3 | 2.5 | 22 | 52 (10263) | 52 (674) |
| 3 | 1 | 2 | 3 | 0.5 | 22 | 34.5 (674) | 34.5 (235) |
| 4 | 3 | 1 | 2 | 4 | 22 | 68 (1239) | 68 (106) |
| 10 | 1 | 2 | 3 | 4 | 27 | 81 (15,587) | 81 (886) |
| 11 | 1 | 2 | 3 | 2.5 | 27 | --- | 66 (468) |
| 12 | 1 | 2 | 3 | 0.5 | 27 | --- | 42.5 (257) |
| 13 | 1 | 2 | 3 | 4 | 32 | --- | 78 (603) |
| 14 | 1 | 2 | 3 | 2.5 | 32 | --- | 78 (603) |
| 15 | 1 | 2 | 3 | 0.5 | 32 | --- | 52 (2587) |

**Table 2(c)**

**INTEGER 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 | D.E.W LP LOWER BOUND (PIVOTS) | R.R LP LOWER BOUND (PIVOTS) |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 | 22 | 50.33 (79) | 59.33 (135) |
| 2 | 1 | 2 | 3 | 2.5 | 22 | 42.33 (79) | 48.33 (159) |
| 3 | 1 | 2 | 3 | 0.5 | 22 | 31.66 (78) | 33.66 (146) |
| 4 | 3 | 1 | 2 | 4 | 22 | 55.33 (72) | 64.5 (142) |
| 10 | 1 | 2 | 3 | 4 | 27 | 62.66 (106) | 73.33 (188) |
| 11 | 1 | 2 | 3 | 2.5 | 27 | 52.66 (102) | 59.83 (177) |
| 12 | 1 | 2 | 3 | 0.5 | 27 | 39.33 (114) | 41.83 (166) |
| 13 | 1 | 2 | 3 | 4 | 32 | 76 (112) | 87.25 (232) |
| 14 | 1 | 2 | 3 | 2.5 | 32 | 64 (121) | 71.33 (242) |
| 15 | 1 | 2 | 3 | 0.5 | 32 | 48 (1120 | 50 (227) |

**Table 2(d)**

**INTEGER PROGRAMMING : COMPUTATIONAL SUMMARY (CONTD)**
**LIVES OF COMPONENTS: $L_1 = 3$, $L_2 = 4$, $L_3 = 5$.**

| P.No. | $C_1$ | $C_2$ | $C_3$ | C | T | D.E.W 1P UPPER BOUND (PIVOTS) | R.R IP UPPER BOUND (PIVOTS) |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 | 22 | NONE | 71 (79) |
| 2 | 1 | 2 | 3 | 2.5 | 22 | NONE | 56.5 (78) |
| 3 | 1 | 2 | 3 | 0.5 | 22 | NONE | 36.5 (97) |
| 4 | 3 | 1 | 2 | 4 | 22 | NONE | 68 (106) |
| 10 | 1 | 2 | 3 | 4 | 27 | NONE | 88 (100) |
| 11 | 1 | 2 | 3 | 2.5 | 27 | NONE | 68 (112) |
| 12 | 1 | 2 | 3 | 0.5 | 27 | NONE | 46.5 (123) |
| 13 | 1 | 2 | 3 | 4 | 32 | NONE | 98 (153) |
| 14 | 1 | 2 | 3 | 2.5 | 32 | NONE | 86.5 (115) |
| 15 | 1 | 2 | 3 | 0.5 | 32 | 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).