# The blast furnaces problem

D. STENGOS and L.C. THOMAS

*Department of Decision Theory, University of Manchester, Manchester M13 9PL, U.K.*

Received November 1978
Revised January 1979

We consider a maintenance and overhaul problem, where there are several identical pieces of equipment, and the only cost is an 'out-of-service' one, which increases more than linearly with the number of items out of service. Using the techniques of Markov decision processes, we find the optimal policy for the case of two units. The problem has its origin in the maintenance of blast furnaces at steel works.

## 1. Introduction

The problem of replacement or overhaul of equipment which deteriorates with usage is one of the standard applications of Markov decision processes. This class of problems was introduced by Derman [2], and further results obtained by Ross [5]. Subsequently, the papers on the subject have been legion, as the reference section of the review article by Pierskalla and Voelker [4] will show. Almost all of these problems have a cost structure consisting of a running cost depending on the state of the equipment and a replacement or overhaul cost. When a replacement is required, it is done instantaneously.

In the problem considered in this paper, the emphasis is somewhat different. We have $n$ identical pieces of equipment which, from time to time, will need overhauling, either because they have failed during operation or to prevent such a failure. The state of the equipment and its probability of failure is described by the number of periods since the last overhaul. An overhaul takes a measurable period of time to perform – in our case, $L$ periods. The cost of running the system, no matter what its state, is considered insignificant when compared with the 'loss of revenue' cost incurred while the equipment is being overhauled. Moreover, this cost increases more than linearly with the number of pieces of equipment being overhauled at that time. Thus, if only one piece of equipment is being overhauled, the cost per period is $c$; if two, it is $d$ ($>2c$); if three, $e > \frac{3}{2} d$, and so on. There is no cost if all the items are working. The object is to find which policy of preventive overhauls minimises the average cost per period. Obviously the decision whether or not to overhaul a piece of equipment will depend not only on the number of periods since overhaul of that equipment, but also the state of the other pieces. One wants to avoid too many pieces of equipment failing in close succession, and thus being overhauled at the same time, but on the other hand too frequent preventive replacements also increase the cost.

An example of the above problem is to examine the problem of overhauling blast furnaces in a steel works. A number of furnaces are used to produce 'pig-iron" from the raw ore, which is in turn converted to various forms of steel in the rest of the steel works. The furnaces have to be overhauled to prevent, or because of, failure, and this means that other parts of the steel works are affected. The disturbance is substantially more, the more furnaces that are out of action, until if all furnaces are being overhauled the whole steel works has to stop. Thus the cost of overhauling the furnaces increases more than linearly with the number of furnaces out of action.

In this paper, we deal with the case where there are only two pieces of equipment, and where the probability, $p_i$, that an item will not fail in the next period, given that it is $i$ periods since the last overhaul, is nonincreasing with $i$. There is no upper bound on the number of periods since overhaul, so the state of a piece of equipment can be any integer $i$, $i \geqslant 0$. In Section 2 we describe how the problem can be set up as a Markov decision process. In Section 3 we concentrate on the relative value successive approximations algorithm, which was introduced by White [8] for the finite-state case and used in the infinite state case by Kushner [3]. We show that the problem satisfies the conditions which guarantee the convergence of the algorithm, which in turn implies the existence of an optimal overhauling policy. We use these results in Section 4 to describe the form of the optimal policy. In Section 5, we give two simple examples of the problem.


European Journal of Operational Research 4 (1980) 330–336.

## 2. Model formulation

The system consists of two identical units $A$, $B$, whose characteristics are totally described by the number of periods since the last overhaul finished. Thus a typical state of the system is $(i, j)$, where unit $A$ is operating $i$ periods since the last overhaul and $B$ is $j$ periods since its last overhaul. At the start of each period, we have to decide whether we want to overhaul one, or both the units, or to let them continue working for the next period. If a unit, $i$ periods since last overhaul, is allowed to continue operating, then with probability $p_i$ it will operate successfully, and with probability $1 - p_i$ it will fail and have to be overhauled. It is assumed throughout that $p_i$ is a monotonically non-increasing function of $i$, and that $1 > p_0 > 0$. When a unit is overhauled, either through choice or failure, it takes $L$ periods to do so, which we represent by the states $\Delta_1, \Delta_2, ..., \Delta_L$. Thus if a unit fails during operation, at the start of the next period it is in state $\Delta_1$; if it is decided to overhaul it preventively, then at the start of the next period it will be in state $\Delta_2$, as it has been overhauled for one complete period. The only cost involved is that due to 'lost operating capacity', when a unit is being overhauled. Thus the cost when in state $(i, j)$ is 0, in $(i, \Delta_j)$ or $(\Delta_j, i)$ is $c$, and $(\Delta_i, \Delta_j)$ is $d$, where $d \geqslant 2c$, no matter which action is taken. The aim is to choose the overhauling policy that minimises the cost per period.

The system is a Markov decision process with state space $\tilde{S} = S \times S$, where $S = \{\Delta_i | 1 \leqslant i \leqslant L\} \cup \{i | i \geqslant 0, i \text{ an integer}\}$. The only action available in state $(\Delta_i, \Delta_j)$ is to continue overhauling, while in $(i, \Delta_j)$, $[(\Delta_j, i)]$, one can either overhaul $A$ $[B]$ or let it operate. For states $(i, j)$ one can either overhaul both items, or let $A$ or $B$ be overhauled, or let both operate.

Since the two items are identical, any equation is invariant under reversal of the order of the two components of the state. From state $(\Delta_i, \Delta_j)$, $1 \leqslant i, j < L$, the system will be in $(\Delta_{i+1}, \Delta_{j+1})$ next period; $(\Delta_i, \Delta_L)$ goes to $(\Delta_{i+1}, 0)$, $1 \leqslant i < L$ and $(\Delta_L, \Delta_L)$ to $(0, 0)$. For the state $(\Delta_i, j)$ if we decide to overhaul we move to $(\Delta_{i+1}, \Delta_2)$ with certainty; if we allow $B$ to continue, then with probability $p_j$ we go to $(\Delta_{i+1}, j + 1)$ and with probability $1 - p_j$ to $(\Delta_{i+1}, \Delta_1)$. Lastly, in state $(i, j)$ if we allow both units to operate, then with probability $p_i p_j$ the next state is $(i + 1, j + 1)$; with probability $p_i (1 - p_j)$ it will be $(i + 1, \Delta_1)$; with probability $p_j (1 - p_i)$, it will be $(\Delta_1, j + 1)$; and with probability $(1 - p_i)(1 - p_j)$ it is $(\Delta_1, \Delta_1)$. If we overhaul $A$ $[B]$ then with probability $p_j$ $[p_i]$ we start the next period in state $(\Delta_2, j + 1)$ $[(i + 1, \Delta_2)]$ and with probability $1 - p_j$ $[1 - p_i]$ we go to $(\Delta_2, \Delta_1)$ $[(\Delta_1, \Delta_2)]$. If we overhaul both, we go to $(\Delta_2, \Delta_2)$ with certainty.

We call a policy reasonable if it does not require a unit to be overhauled when it is only 0 or 1 period since the last overhaul, and the other unit is already being or has just finished being overhauled.

**Lemma 1.** *A reasonable policy corresponds to a Markov chain with only one ergodic class of states (though there may be transient states).*

**Proof.** Let $R = \{\Delta_1, ..., \Delta_L, 0\}$ and $F = R \times R$. It is obvious that from any state of $\tilde{S}$, under any policy, there is a positive probability of going to a state in $F$. Thus if we show the states in $F$ communicate (using the terminology of Chung [1]), it follows that there is only one closed set of states. If the system is in state $(\Delta_1, \Delta_{1+k})$, it will pass through all the states $\{(\Delta_i, \Delta_{i+k}), 1 \leqslant i \leqslant L - k\}$ automatically. Thus if we can show that there is a positive probability of reaching $(\Delta_1, \Delta_{k+2})$ from $(\Delta_1, \Delta_{k+1})$, repeated application of this result implies that we can get from any state in F to any other state with positive probability.

From $(\Delta_1, \Delta_{k+1})$ we automatically go to the state $(\Delta_{L-k+1}, 0)$. Under a reasonable policy with probability $1 - p_0$ the system will next be in $(\Delta_{L-k+2}, \Delta_1)$ and hence automatically in $(0, \Delta_k)$. With probability $p_0$ the next state is $(1, \Delta_{k+1})$ and with probability $1 - p_1$ thence to $(\Delta_1, \Delta_{k+2})$. Hence the states of F are communicating.

They are also aperiodic because we can go from $(\Delta_1, \Delta_k)$ via $(\Delta_{L-k+2}, 0)$, $(\Delta_{L-k+3}, \Delta_1)$, ..., $(0, \Delta_{k-1})$ to $(\Delta_1, \Delta_k)$ in $L + 1$ steps or via $(\Delta_{L-k+2}, 0)$, $(\Delta_{L-k+3}, 1)$, $(\Delta_{L-k+4}, \Delta_1)$, ..., $(0, \Delta_{k-2})$, $(1, \Delta_{k-1})$ to $(\Delta_1, \Delta_k)$ in $L + 2$ steps.

## 3. Relative value successive approximation algorithm

In order both to prove that there is an optimal stationary overhaul policy and to find some of its properties, we will need to use the relative value successive approximations algorithm, and so in this section we describe the algorithm and show that it can be applied to the above Markov decision process. In state $i$ of a general state space $T$, we can take any action from the set $K_i$, which gives an immediate cost $r_i^k$ and transition probabilities $p_{ij}^k$, then the minimal average cost per period, $g$, satisfies the optimality equations for all $i$;

$$g + v(i) = \min_{k \in K_i} \{r_i^k + \sum_{j \in T} p_{ij}^k v(j)\}, \qquad \forall\, i \in T \tag{3.1}$$

and the minimising $k$'s give the optimal policy. White [8] looked at the finite state problem, with a state, 0, such that for some $n$, and $\alpha > 0$, and any choice of actions $k_1, k_2, ..., k_n$, $p_{i0}^{k_1 \cdot k_2 \cdots k_n} \geqslant \alpha$, $\forall\, i \in T$. He showed that under this condition the algorithm defined by

$$v_{n+1}(i) + g_{n+1} = \min_{k \in K_i} \{r_i^k + \sum_{j \in T} p_{ij}^k v_n(j)\}, \qquad i \neq 0,$$

$$g_{n+1} = \min_{k \in K_0} \{r_0^k + \sum_{j \in T} p_{0j}^k v_n(j)\} \tag{3.2}$$

has the property that $g_n$ and $v_n(i)$ converge to a solution of (3.1). Schweitzer [6] generalised the algorithm and Kushner [3] extended it to the countable state case (with $T = (0, 1, 2, ...)$) in the following way.

**Theorem 1** [3, Theorem 6.5, p. 156]. *Suppose*

(C1) *Each of the policies gives rise to a Markov chain with only one ergodic class.*

(C2) $|r_i^k| \leqslant M$, $\forall\, i \in T$, $k \in K_i$.

(C3) *For each* $\epsilon > 0$ *and* $i \in T$, *there is an* $n(i, \epsilon)$ *such that* $\Sigma_{j=n(i,\epsilon)}^{\infty} p_{ij}^k < \epsilon$, $\forall\, k \in K_i$.

(C4) *There is a* $\alpha > 0$ *and* $n$ *such that for some state* 0 *and any actions* $k_1, ..., k_n$, $p_{i0}^{k_1, ..., k_n} \geqslant \alpha > 0$, $\forall\, i \in T$, *then there is an optimal stationary policy and* $g_n$ *and* $v_n(\cdot)$ *defined by* (3.2) *converge to the optimal* $g$ *and* $v(i)$ *which satisfy* (3.1).

*If we show that conditions* (C1) – (C4) *of Theorem* 1 *apply in the above problem, then we know that there exists an optimal overhaul policy which is stationary, and that the relative value successive approximations algorithm converges to it.*

**Lemma 2.** *For the set of all reasonable policies, conditions* (C1)–(C4) *of Theorem 1 hold.*

**Proof.** (C1) follows from Lemma 1, while (C2) is trivial, since the only costs are 0, $c$ and $d$. Also, (C3) follows immediately from the fact that the action space corresponding to each state is finite.

For (C4), we want to show that after $N$ steps we have a positive probability of being in state (0, 0). Under any policy, and starting in any state, there is a probability of at least $(1 - p_0)^2$ of being in $F$ at the end of the first period, either because of failure or a decision to overhaul one or both of them. Thus it is enough to show that starting from any state in $F$ after $N$ further moves, we have a positive probability of being in (0, 0). For any reasonable policy, an item can go $\Delta_L, 0, \Delta_1$ with probability $1 - p_0$, and $\Delta_L, 0, 1, \Delta_1$, with probability $p_0(1 - p_1)$. Thus it has cycles of length $L + 1$ or $L + 2$ with probabilities $(1 - p_0)$, $p_0(1 - p_1)$ respectively. Given an item now in $\Delta_i$ (think of 0 as $\Delta_{L+1}$), $1 \leqslant i \leqslant L + 1$, we want an $N$, so that in $N$ steps it has a positive probability of being in $\Delta_{L+1}$, having completed a whole number of cycles of length either $L + 1$ or $L + 2$, i.e., we want an $N$ so that

$$N + i = c_i(L + 1) + d_i(L + 2) \tag{3.3}$$

holds for all $i$, $1 \leqslant i \leqslant L + 1$, where $c_i$ and $d_i$ must be integers. Trivially, if we let $N$ be $(L + 1)^2$, and $d_i = i$, $c_i = L + 1 - i$, (3.3) holds. Thus in $(L + 1)^2$ more periods, starting in any state of $F$, the system has probability at least $(1 - p_0)^2 \min\{(1 - p_0)^{L+1}, (p_0(1 - p_1)^{L+1})\}$ of being in (0, 0). Thus (C4) holds and the successive approximation algorithm converges.

For this problem, the successive approximations algorithm has the following form, recalling that equations are invariant under reversal of the ordering of the co-ordinates of the states

$$v_0(\cdot, \cdot) = 0 \qquad \text{for all states of } S, \tag{3.4}$$

$$\begin{aligned} g_{n+1} = \min\{&p_0^2 v_n(1, 1) + p_0(1 - p_0)v_n(1, \Delta_1) \\ &+ p_0(1 - p_0)v_n(\Delta_1, 1) \\ &+ (1 - p_0)^2 v_n(\Delta_1, \Delta_1); c + p_0 v_n(1, \Delta_2) + \\ &+ (1 - p_0)v_n(\Delta_1, \Delta_2); \\ &c + p_0 v_n(\Delta_2, 1) + (1 - p_0)v_n(\Delta_2, \Delta_1); \\ &d + v_n(\Delta_2, \Delta_2)\}, \end{aligned} \tag{3.5}$$

$$\begin{aligned} g_{n+1} + v_{n+1}(i, j) = \min\{&p_i p_j v_n(i + 1, j + 1) \\ &+ p_i(1 - p_j)v_n(i + 1, \Delta_1) + p_j(1 - p_i)v_n(\Delta_1, j + 1) \\ &+ (1 - p_i)(1 - p_j)v_n(\Delta_1, \Delta_1); \\ &c + p_i v_n(i + 1, \Delta_2) + (1 - p_i)v_n(\Delta_1, \Delta_2); \\ &c + p_j v_n(\Delta_2, j + 1) + (1 - p_j)v_n(\Delta_2, \Delta_1); \\ &d + v_n(\Delta_2, \Delta_2)\}, \qquad i, j \geqslant 0, \end{aligned} \tag{3.6}$$

$$\begin{aligned} g_{n+1} + v_{n+1}(i, \Delta_k) = \min\{&c + p_i v_n(i + 1, \Delta_{k+1}) + \\ &(1 - p_i)v_n(\Delta_1, \Delta_{k+1}); \\ &d + v_n(\Delta_2, \Delta_{k+1})\}, \qquad i \geqslant 0, 1 \leqslant k \leqslant L, \end{aligned} \tag{3.7}$$

$$g_{n+1} + v_{n+1}(\Delta_i, \Delta_k) = d + v_n(\Delta_{i+1}, \Delta_{k+1}), \quad 1 \leqslant i, k \leqslant L \tag{3.8}$$

where $\Delta_{L+1}$ is identified with the state 0 in (3.7) and (3.8).

## 4. Form of the optimal policy

In this section, we describe the form of the optimal policy among the set of all reasonable policies. Our main result is

**Theorem 2.** *If one blast furnace is being overhauled, then the optimal action is never to overhaul the other one. If blast furnace A (B) has been i periods since overhaul and B (A) is j periods since overhaul with $i \leqslant j$, then the optimal action is to overhaul B (A) only, if $j \geqslant j_i$, and to repair neither if $j < j_i$, where $j_i$, which may be infinity, is a critical number depending only on i.*

The proof will consist of a series of lemmas concerning the properties of the value function $v_n$, in the relative value successive approximations algorithm. Note that substituting $v_0(\cdot,\cdot) = 0$ in (3.5)–(3.8) gives

$$g_1 = 0,\ \ v_1(i,j) = 0,\ \ v_1(i,\Delta_k) = c \ \text{ and } \ v_1(\Delta_l,\Delta_k) = d, \quad i,j \geqslant 0, \quad 1 \leqslant l,k \leqslant L \tag{4.1}$$

**Lemma 3.** (1) *Let $h(i)$ be non-decreasing with $h(i) \leqslant h(\Delta_1)$, $\forall\, i$, $p_i$ non-increasing, and*

$$g(i) = p_i h(i+1) + (1-p_i)h(\Delta_1),\quad \forall\, i \geqslant 0, \tag{4.2}$$

*then $g(i)$ is non-decreasing in i with $g(i) \leqslant h(\Delta_1)$.*

(b) *Let $h(i,j)$ be non-decreasing in i and j separately, when the other component is fixed, and $h(i,\Delta_1)$, $h(\Delta_1,i)$ be non-decreasing in i, $p_i$ non-increasing with $h(i,j) \leqslant h(i,\Delta_1)$, $h(\Delta_1,j) \leqslant h(\Delta_1,\Delta_1)$, $\forall\, i,j \geqslant 0$. Let*

$$\begin{aligned} g(i,j) = {} & p_i p_j h(i+1,j+1) + (1-p_i)p_j h(\Delta_1,j+1) + \\ & p_i(1-p_j)h(i+1,\Delta_1) \\ & + (1-p_i)(1-p_j)h(\Delta_1,\Delta_1),\quad i,j \geqslant 0, \end{aligned} \tag{4.3}$$

$$g(i,\Delta_1) = p_i h(i+1,\Delta_1) + (1-p_i)h(\Delta_1,\Delta_1), \tag{4.4}$$

$$g(\Delta_1,j) = p_j h(\Delta_1,j+1) + (1-p_j)h(\Delta_1,\Delta_1), \tag{4.5}$$

*then $g(i,j)$ is non-decreasing in i and j separately when the other component is fixed, and $g(\Delta_1,i)$, $g(i,\Delta_1)$ are non-decreasing in i, and*

$$g(i,j) \leqslant g(i,\Delta_1), \qquad g(\Delta_1,i) \leqslant h(\Delta_1,\Delta_1).$$

**Proof.** (a) The proof follows the usual stochastic dominance argument. For all $i \geqslant 1$

$$\begin{aligned} g(i-1) &= h(i) + (1-p_{i-1})(h(\Delta_1) - h(i)) \leqslant h(i) + \\ &\quad + (1-p_i)(h(\Delta_1) - h(i)) \\ &= p_i h(i) + (1-p_i)h(\Delta_1) \leqslant p_i h(i+1) + \\ &\quad + (1-p_i)h(\Delta_1) = g(i). \end{aligned} \tag{4.4}$$

(b) For all $i, j \geqslant 0$, using part (a) and the assumptions of part (b)

$$\begin{aligned} g(i,j) &= p_j[p_i h(i+1,j+1) + (1-p_i)h(\Delta_1,j+1)] + \\ &\quad + (1-p_j)[p_i h(i+1,\Delta_1) + (1-p_i)k(\Delta_1,\Delta_1)] \\ &\leqslant p_j[p_{i+1}h(i+2,j+1) + (1-p_{i+1})h(\Delta_1,j+1)] \\ &\quad + (1-p_j)][p_{i+1}h(i+2,\Delta_1) \\ &\quad + (1-p_{i+1})h(\Delta_1,\Delta_1)] = g(i+1,j) \\ &\leqslant p_j h(\Delta_1,j+1) + (1-p_j)h(\Delta_1,\Delta_1) = g(\Delta_1,j). \end{aligned} \tag{4.7}$$

Similarly

$$g(i,j) \leqslant g(i,j+1) \leqslant g(i,\Delta_1) \tag{4.8}$$

and by part (a) $g(\Delta_1,i)$, $g(i,\Delta_1)$ are monotonically non-decreasing in $i$ and bounded above by $h(\Delta_1,\Delta_1)$.

We now prove some properties of the $v_n(\cdot,\cdot)$ defined in (3.5)–(3.8).

**Lemma 4.** For $n \geqslant 1$

(i) $v_n(\cdot,\cdot)$ is invariant under reversal of the order of the co-ordinates;

(ii) $v_n(i,\Delta_s) \leqslant v_n(i+1,\Delta_s) \leqslant v_n(\Delta_1,\Delta_s)$, $\forall i \geqslant 0$, $1 \leqslant s \leqslant L$;

(iii) $v_n(i,j) \leqslant v_n(i+1,j) \leqslant v_n(\Delta_1,j) \leqslant v_n(\Delta_1,\Delta_1)$, $\forall i,j \geqslant 0$;

(iv) $c + v_n(\Delta_s,\cdot) \leqslant d + v_n(\Delta_{s+1},\cdot)$ $1 \leqslant s \leqslant L-1$, $c + v_n(\Delta_L,\cdot) \leqslant d + v_n(0,\cdot)$.

**Proof.** Clearly from (4.1) these properties hold for $n = 1$. We proceed by induction. (i) is trivial because the equations (3.5)–(3.8) defining $v_{n+1}(i,j)$, $v_{n+1}(j,i)$ are invariant under co-ordinate reversal. Since this is also true for the values of the $v_n(\cdot,\cdot)$ on the right-hand side of the equations, it must be true for $v_{n+1}(\cdot,\cdot)$.

In (3.7) defining $v_{n+1}(i,\Delta_s)$, $1 \leqslant s \leqslant L$, $i \geqslant 0$, (ii) of the induction hypothesis, together with Lemma 3, imply that $v_{n+1}(i,\Delta_s)$ is non-decreasing with $i$. Further, (ii) and (iv) of the induction hypothesis imply, for $i \geqslant 0$, $1 \leqslant s \leqslant L$, where $\Delta_{L+1}$ is identified with 0,

$$g_{n+1} + v_{n+1}(i, \Delta_s) \leqslant c + p_i v_n(i+1, \Delta_{s+1}) + (1-p_i)v_n(\Delta_1, \Delta_{s+1}) \leqslant c + v_n(\Delta_1, \Delta_{s+1}) \leqslant d + v_n(\Delta_2, \Delta_{s+1}) = g_{n+1} + v_{n+1}(\Delta_1, \Delta_s). \tag{4.9}$$

So $v_{n+1}(i, \Delta_s) \leqslant v_{n+1}(\Delta_1, \Delta_s)$.

For (iii), looking at (3.6) which defines $v_{n+1}(i, j)$, $i, j \geqslant 0$, and using (iii) of the induction hypothesis and Lemma 3 implies that the first two expressions on the right-hand side of (3.6) are monotonically non-decreasing in $i$, while the second two are independent of $i$. Thus $v_{n+1}(i, j) \leqslant v_{n+1}(i+1, j)$, $\forall i, j \geqslant 0$. Moreover, using (iv) of induction hypothesis,

$$g_{n+1} + v_{n+1}(i, j) \leqslant c + p_j v_n(\Delta_2, j+1) + (1-p_j)v_n(\Delta_2, \Delta_1) = g_{n+1} + v_{n+1}(\Delta_1, j) \leqslant c + v_n(\Delta_2, \Delta_1) \leqslant d + v_n(\Delta_2, \Delta_2) = g_{n+1} + v_{n+1}(\Delta_1, \Delta_1). \tag{4.10}$$

So $v_{n+1}(i, j) \leqslant v_{n+1}(\Delta_1, j) \leqslant v_{n+1}(\Delta_1, \Delta_1)$.

Finally, to prove the (iv) part of the induction hypothesis,

$$c + v_{n+1}(\Delta_s, \Delta_k) = c + d + v_n(\Delta_{s+1}, \Delta_{k+1}) - g_{n+1} \leqslant 2d + v_n(\Delta_{s+2}, \Delta_{k+1}) - g_{n-1} = d + v_{n+1}(\Delta_{s+1}, \Delta_k), \quad 1 \leqslant s \leqslant L-2,\ 1 \leqslant k \leqslant L-1. \tag{4.11}$$

Similarly it follows that

$$c + v_{n+1}(\Delta_{L-1}, \Delta_k) \leqslant d + v_{n+1}(\Delta_L, \Delta_k), \quad 1 \leqslant k \leqslant L-1 \tag{4.12}$$

and

$$c + v_{n+1}(\Delta_s, \Delta_L) \leqslant d + v_{n+1}(\Delta_{s+1}, \Delta_L), \quad 1 \leqslant s \leqslant L-1. \tag{4.13}$$

Now,

$$d + v_{n+1}(0, \Delta_k) = d + c + p_0 v_n(1, \Delta_{k+1}) + (1-p_0)v_n(\Delta_1, \Delta_{k+1}) - g_{n+1} \geqslant d + c + v_n(0, \Delta_{k+1}) - g_{n+1} \quad \text{(by (ii) of hypothesis)}$$
$$= c + v_{n+1}(\Delta_L, \Delta_k), \quad 1 \leqslant k \leqslant L-1 \tag{4.14}$$

and a similar argument gives $d + v_{n+1}(0, \Delta_L) \geqslant c + v_{n+1}(\Delta_L, \Delta_L)$.

For $v_{n+1}(\Delta_s, i)$, $i \geqslant 0$, $1 \leqslant s \leqslant L-2$,

$$c + v_{n+1}(\Delta_s, i) \leqslant 2c + p_i v_n(\Delta_{s+1}, i+1) + (1-p_i)v_n(\Delta_{s+1}, \Delta_1) - g_{n+1} = c + p_i(c + v_n(\Delta_{s+1}, i+1)) + (1-p_i)(c + v_n(\Delta_{s+1}, \Delta_1)) - g_{n+1}$$
$$\leqslant c + d + p_i v_n(\Delta_{s+2}, i+1) + (1-p_i)v_n(\Delta_{s+2}, \Delta_1) - g_{n+1} = d + v_{n+1}(\Delta_{s+1}, i) \tag{4.15}$$

A similar argument gives $c + v_{n+1}(\Delta_{L-1}, i) \leqslant d + v_{n+1}(\Delta_L, i)$, $\forall i \geqslant 0$.

Lastly, look at $c + v_{n+1}(\Delta_L, i) \leqslant d + v_{n+1}(0, i)$. By (3.6) $v_{n+1}(0, i)$ is the minimum of four terms. Lemma 3 and the induction hypothesis imply that the minimum is one of the first two terms of (3.6), i.e. either overhaul neither or the $i$ period old one. Thus there are only two cases to consider, namely

$$g_{n+1} + v_{n+1}(0, i) = p_0 p_i v_n(1, i+1) + p_0(1-p_i)v_n(1, \Delta_1) + (1-p_0)p_i v_n(\Delta_1, i+1) + (1-p_0)(1-p_i)v_n(\Delta_1, \Delta_1), \tag{4.16}$$

$$g_{n+1} + v_{n+1}(0, i) = c + p_0 v_n(1, \Delta_2) + (1-p_0)v_n(\Delta_1, \Delta_2). \tag{4.17}$$

In (4.16) using (iii) and (iv) of the induction hypothesis gives

$$d + v_{n+1}(0, i) \geqslant d - g_{n+1} + p_i v_n(0, i+1) + (1-p_0)v_n(0, \Delta_1) \geqslant 2c - g_{n+1} + p_i v_n(0, i+1) + (1-p_i)v_n(0, \Delta_1) = c + v_n(\Delta_L, i). \tag{4.18}$$

For (4.17), the induction hypothesis gives

$$d + v_{n+1}(0, i) \geqslant c - g_{n+1} + p_0(c + v_n(1, \Delta_1)) + (1-p_0)(c + v_n(\Delta_1, \Delta_1)) \geqslant c - g_{n+1} + c + v_n(1, \Delta_1) \geqslant c - g_{n+1} + c + v_n(0, \Delta_1) \geqslant c + v_{n+1}(\Delta_L, i). \tag{4.19}$$

The last inequality follows, since

$$v_{n+1}(\Delta_L, i) = -g_{n+1} + c + p_i v_n(0, i+1) + (1-p_i)v_n(0, \Delta_1) \leqslant -g_{n+1} + c + v_n(0, \Delta_1). \tag{4.20}$$

The main result now follows from Lemma 4.

**Proof of Theorem 2.** Since we proved in Lemma 2 that the conditions for the convergence of the relative value algorithm are satisfied, $\lim_{n\to\infty} v_n(\cdot, \cdot) = v(\cdot, \cdot)$ exists and satisfies the optimality equation. Moreover, the properties proved in Lemma 4 for $v_n(\cdot, \cdot)$ will still hold in the limit so $v(\cdot, \cdot)$ is invariant under reversal of co-ordinates and properties (ii), (iii), and

(iv) hold for $v(\cdot, \cdot)$. Thus when we look at the optimality equation, again identifying $\Delta_{L+1}$ with 0,

$$g + v(i, \Delta_s) = \min\{c + p_i v(i+1, \Delta_{s+1}) + (1-p_i)v(\Delta_1, \Delta_{s+1}); \quad d + v(\Delta_2, \Delta_{s+1})\}, \qquad 1 \leqslant s \leqslant L. \tag{4.21}$$

Lemma 4(iii) gives $v(i+1, \Delta_{s+1}) \leqslant v(\Delta_1, \Delta_{s+1})$ and Lemma 4(iv) tells us that $c + v(\Delta_1, \Delta_{s+1}) \leqslant d + v(\Delta_2, \Delta_{s+1})$. Thus the first term on the right-hand side is the minimum, and so if $B$ ($A$) is out of action, the optimal decision is not to overhaul $A$ ($B$).

Similarly for $i \leqslant j$,

$$\begin{aligned} g + v(i, j) = \min\{&p_i p_j v(i+1, j+1) + p_i(1-p_j)v(i+1, \Delta_1) \\ &+ p_j(1-p_i)v(\Delta_1, j+1) + (1-p_i)(1-p_j)v(\Delta_1, \Delta_1); \\ &c + p_i v(i+1, \Delta_2) + (1-p_i)v(\Delta_1, \Delta_2); \\ &c + p_j v(\Delta_2, j+1) + (1-p_j)v(\Delta_2, \Delta_1); d + v(\Delta_2, \Delta_2)\}. \end{aligned} \tag{4.22}$$

The fact that $v(i, \Delta_2)$ is non-decreasing in $i$ and bounded above by $v(\Delta_1, \Delta_2)$, and that

$$c + v(\Delta_1, \Delta_2) \leqslant d + v(\Delta_2, \Delta_2)$$

together with Lemma 3 imply that the first two terms are always less than or equal to the second two terms. Thus we do not overhaul either or overhaul, $B$, the one which is longer since being overhauled. $B$ will be overhauled if

$$\begin{aligned} &c + p_i v(i+1, \Delta_2) + (1-p_i)v(\Delta_1, \Delta_2) \\ &\leqslant p_i p_j v(i+1, j+1) + p_i(1-p_j)v(i+1, \Delta_1) \\ &\quad + p_j(1-p_i)v(\Delta_1, j+1) + (1-p_i)(1-p_j)v(\Delta_1, \Delta_1). \end{aligned} \tag{4.23}$$

Since by Lemma 4 the right-hand side is monotonically non-decreasing in $j$, and the left-hand side is independent of $j$, if $j_i$ is the smallest $j$ which satisfies this, then for all $j > j_i$, the inequality holds and we overhaul $B$. Notice that $j_i$ depends only on $i$, since only $i$ and $j$ appear in the inequality.

## 5. Examples

We close with two very simple examples. It is obvious that the relative value successive approximations algorithm, described in Section 3, will prove useful computationally for finite state problems (i.e. where $p_i = 0$ for some $i$). Even for the infinite state problem Stengos [7] has shown that a modified form of this algorithm can still be applied in this case. For very small problems policy iteration also converges very quickly, since Theorem 2 gives us an idea of the optimal policy.

Our first example verifies the obvious result, that if $p_i = p$, for all $i \geqslant 0$, then $j_i = \infty$ for all $i$ – that is, if the failure rate is constant, then one never performs a preventive overhaul. In this case, the expressions are such that one can obtain an analytic solution to the optimal equation.

Let $L = 2, p_i = p$, then, using Theorem 2, the optimality equations become

$$\begin{aligned} g + v(i, j) = \min\{&p^2 v(i+1, j+1) + p(1-p)v(i+1, \Delta_1) \\ &+ p(1-p)v(j+1, \Delta_1) + (1-p)^2 v(\Delta_1, \Delta_1); \\ &c + pv(i+1, \Delta_2) + (1-p)v(\Delta_1, \Delta_2)\}, \quad i \leqslant j; \end{aligned} \tag{5.1}$$

$$g + v(i, \Delta_1) = c + pv(i+1, \Delta_2) + (1-p)v(\Delta_1; \Delta_2); \tag{5.2}$$

$$g + v(i, \Delta_2) = c + pv(i+1, 0) + (1-p)v(\Delta_1, 0); \tag{5.3}$$

$$g + v(\Delta_1, \Delta_1) = d + v(\Delta_2, \Delta_2); \quad g + v(\Delta_1, \Delta_2) = d + v(\Delta_2, 0);$$

$$g + v(\Delta_2, \Delta_2) = d + v(0, 0) \quad \text{and} \quad v(0, 0) = 0.$$

These equations used the results of Theorem 2. It is an easy exercise to show that a solution is given by

$$g = \frac{4(1-p)(c+d-dp)}{(3-2p)^2} \quad v(i, j) = 0, \quad \forall i, j > 0;$$

$$v(i, \Delta_1) = (2(c-g) + (1-p)(d-g))/p, \quad i \geqslant 0;$$

$$v(i, \Delta_2) = ((2-p)(c-g) + (1-p)^2(d-g))/p, \quad i \geqslant 0;$$

$$v(\Delta_2, \Delta_2) = d - g; \qquad v(\Delta_1, \Delta_1) = 2(d-g);$$

$$v(\Delta_1, \Delta_2) = ((2-p)(c-g) + (1-p+p^2)(d-g))/p.$$

In this case the first term on the right-hand side of (5.1) is always less than the second, so one never overhauls preventively.

The second example is one where $p_i = p^i$ for $0 \leqslant i \leqslant 3$ and $p_4 = 0$, so that one can think of it as a finite state problem, $L = 2, c = 2, d = 6, p_0 = 0.9$, $p_1 = (0.9)^2, p_3 = (0.9)^3, p_4 = 0$. Using policy improvement and starting with the policy $j_i = \infty$ for all $i$, after two iterations we obtained the optimal policy $j_1 = \infty$, $j_2 = 4, j_3 = 3, j_4 = 4$.

The optimal value of $g$ was 1.448 and the $v(\cdot, \cdot)$ are given in Table 1.

Table 1

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| State | (0, 0) | (0, 1) | (0, 2) | (0, 3) | (0, 4) | $(0, \Delta_1)$ | $(0, \Delta_2)$ | (1, 1) | (1, 2) | (1, 3) |
| $\nu(\cdot, \cdot)$ | 0.000 | 0.518 | 0.859 | 1.058 | 1.290 | 2.085 | 1.069 | 1.448 | 1.880 | 2.206 |
| Action | N.O. | N.O. | N.O. | N.O. | N.O. | N.O. | N.O. | N.O. | N.O. | N.O. |
| State | (1, 4) | $(1, \Delta_1)$ | $(1, \Delta_2)$ | (2, 2) | (2, 3) | (2, 4) | $(2, \Delta_1)$ | $(2, \Delta_2)$ | (3, 3) | (3, 4) |
| $\nu(\cdot, \cdot)$ | 2.419 | 2.738 | 1.533 | 2.733 | 3.113 | 3.286 | 3.286 | 1.804 | 3.997 | 3.997 |
| Action | N.O. | N.O. | N.O. | N.O. | N.O. | O.B. | N.O. | N.O. | O.E. | O.B. |
| State | $(3, \Delta_1)$ | $(3, \Delta_2)$ | (4, 4) | $(4, \Delta_1)$ | $(4, \Delta_2)$ | $(\Delta_1, \Delta_1)$ | $(\Delta_1, \Delta_2)$ | $(\Delta_2, \Delta_2)$ | | |
| $\nu(\cdot, \cdot)$ | 3.997 | 2.057 | 6.173 | 6.173 | 2.637 | 9.104 | 5.621 | 4.552 | | |
| Action | N.O. | N.O. | O.E. | N.O. | N.O. | N.O. | N.O. | N.O. | | |

N.O. means No Overhaul, O.E. means Overhaul Either and O.B. means Overhaul B.

## References


[1] K.L. Chung, Markov Chains with Stationary Transition Probabilities (Springer, Berlin, 1960).

[2] C. Derman, Finite State Markov Decision Processes (Academic Press, New York, 1970).

[3] H. Kushner, Introduction to Stochastic Control (Holt, Rinehart and Winston, New York, 1971).

[4] W.P. Pierskalla and J.A. Voelker, A survey of maintenance models: the control and surveillance of deteriorating systems, Naval Res. Logist. Quart. 23 (1976) 353–388.

[5] S. Ross, Applied Probability Models with Optimisation Applications (Holden–Day, San Francisco, 1970).

[6] P.J. Schweitzer, Iterative solution of the functional equations of undiscounted Markov renewal programming, J. Math. Anal. Appl. 34 (1971) 495–501.

[7] D. Stengos and L.C. Thomas, Finite state approximations for denumerable state Markov decision processes – the Average Cost Case, Notes in Decision Theory No. 60, Dept. of Decision Theory, Manchester.

[8] D.J. White, Dynamic programming, Markov chains, and the method of successive approximations, J. Math. Anal. Appl. 6 (1963) 373–376.