Optimizing Preventive Maintenance Models

Michael Bartholomew-Biggs
School of Physics Astronomy and Mathematics, University of Hertfordshire
Bruce Christianson
School of Computer Science, University of Hertfordshire
Ming Zuo
Department of Mechanical Engineering, University of Alberta

Part of this work was carried out while the first author was a Visiting Professor in the Department of Mechanical Engineering at the University of Alberta in December 2003.

Abstract
We deal with the problem of scheduling preventive maintenance (PM) so that, over the operating life of the system, we minimize a performance function which reflects repair and replacement costs as well as the costs of the PM itself. It is assumed that a *hazard rate* model is known which predicts the frequency of system failure as a function of its age; and it is further assumed that each PM produces a step reduction in the *effective age* of the system.

We consider some variations and extensions of a PM scheduling approach proposed by Lin et al [1]. In particular we consider numerical algorithms which may be more appropriate for hazard rate models which are less simple than those used in [1] and we introduce some constraints into the problem in order to avoid the possibility of spurious solutions. We also discuss the use of automatic differentiation (AD) as a convenient tool for computing the gradients and Hessians that are needed by numerical optimization methods.

The main contribution of the paper is a new problem formulation which allows the optimal *number* of occurrences of PM to be determined along with their optimal timings. This formulation involves the global minimization of a non-smooth performance function. In our numerical tests this is done via the algorithm DIRECT proposed by Jones et al [4]; and we show results for a number of examples, involving different hazard rate models, to give an indication of how PM schedules can vary in response to changes in relative costs of maintenance, repair and replacement.

The final part of the paper outlines some future lines of investigation by which the non-smooth global optimization problem could be replaced by a smooth one. This would have the advantage that rapidly convergent local minimization methods could be used to refine the coarser estimates of global solutions provided by DIRECT.

1 Introduction

In this paper we consider the scheduling of preventive maintenance (PM). We begin with the idea that the frequency of breakdown in any system is a function of its *age*. More precisely, we assume that the number of failures occurring between times $t = a$ and $t = b$ is given by

$$\int_a^b h(t)dt$$

where $h(t)$ is the *hazard rate function* . If $h(t)$ is a monotonically increasing function of time then failures occur more often as the system ages. PM can be viewed as a way of causing the system to have an *effective age* which is less than its calendar age. Suppose that, for a system entering service at time $t = 0$, the first PM occurs at time $t_1 = x_1$. Just before this maintenance is carried out the system's effective age y_1 is the same as its calendar age x_1. We suppose that PM reduces the effective age to b_1x_1, where b_1 is some constant lying between 0 and 1. Then, during the period until the next maintenance at time t_2, the effective age of the system is

$$y = b_1x_1 + x, \text{ where } 0 < x < t_2 - t_1.$$

Even though its effective age has been reduced, the failure rate after PM may not be precisely the same as for a genuinely younger system. Therefore we shall assume that the number of failures occurring between $t = 0$ and $t = t_2$ is given by

$$\int_0^{x_1} h(x)dx + \int_0^{x_2} a_1h(b_1x_1 + x)dx.$$

Here $x_2 = t_2 - t_1$ is the interval between the first and second PM and a_1 is some system-dependent constant, greater than or equal to 1.

The effective age of the system just before the second PM at time t_2 is

$$y_2 = b_1x_1 + x_2$$

and immediately after maintenance this is reduced to b_2y_2 for some b_2 such that $b_1 \le b_2 \le 1$. Thus, in the interval between the second and third PM (at time t_3) the effective age is

$$y = b_2y_2 + x = b_2b_1x_1 + b_2x_2 + x, \text{ where } 0 < x < x_3 = t_3 - t_2.$$

Moreover, the predicted number of failures between $t = t_2$ and $t = t_3$ is

$$\int_0^{x_3} a_2a_1h(b_2y_2 + x)dx$$

for some $a_2 \ge 1$.

Now, for $k = 1,..,n$, we let y_k denote the effective age of the system just before the k-th PM at time t_k. We also let x_k denote the time interval $t_k - t_{k-1}$. This implies the relationships

$$t_k = \sum_{i=1}^{k} x_i \tag{1.1}$$

$$y_k = b_{k-1} y_{k-1} + x_k = (\sum_{j=1}^{k-1} B_j x_j) + x_k \quad \text{where} \quad B_j = \Pi_{i=j}^{k-1} b_i. \tag{1.2}$$

The intervals between PM are given in terms of the y_k by

$$x_k = y_k - b_{k-1} y_{k-1}. \tag{1.3}$$

The ideas outlined above are explained more fully by Lin et. al. [1]. To be consistent with their notation we let $H_k(t)$ denote the indefinite integral

$$H_k(t) = \int A_k h(t) dt \quad \text{where} \quad A_k = \Pi_{i=1}^{k-1} a_i.$$

Then the number of failures occurring between t_{k-1} and t_k can be written as

$$H_k(y_k) - H_k(b_{k-1} y_{k-1}).$$

$H_k(t)$ is the *cumulative hazard rate* between the $(k-1)$-th and k-th PM – i.e., for time t in the range

$$x_1 + x_2 + .. + x_{k-1} \leq t \leq x_1 + x_2 + .. + x_k.$$

We can use the ideas outlined above to formulate the problem of determining an optimal schedule for PM. Specifically we consider the the minimization of a performance function of the form

$$C(y) = \frac{R_c}{T} = \frac{\gamma_r + (n-1) + \gamma_m \sum_{k=1}^{n} [H_k(y_k) - H_k(b_{k-1} y_{k-1})]}{y_n + \sum_{k=1}^{n-1} (1 - b_k) y_k} \tag{1.4}$$

where

$$\gamma_r = \frac{\text{Cost of system replacement}}{\text{Cost of PM}} \quad \text{and} \quad \gamma_m = \frac{\text{Cost of minimal system repair}}{\text{Cost of PM}}.$$

The function (1.4) is given in [1] and assumes that PM takes place $n-1$ times with the n-th PM actually being a system replacement. The numerator R_c in (1.4) represents the lifetime cost of the system, expressed as a multiple of the cost of one PM. R_c includes the replacement cost plus the cost of $n-1$ PMs plus the cost of repairs for each breakdown predicted by the hazard-rate function. The denominator, T, is simply the total life of the system. This follows because

$$y_n + \sum_{k=1}^{n-1} (1 - b_k) y_k = y_1 + \sum_{k=1}^{n} y_{k+1} - b_k y_k. \tag{1.5}$$

Using (1.3) and $x_1 = y_1$, the right hand side of (1.5) is equivalent to

$$\sum_{k=1}^{n} x_k = t_n.$$

Hence (1.4) represents the *mean cost* of operating the system.

For the case when the hazard rates are Weibull functions of the form

$$h(t) = \beta t^{\alpha-1} \quad \text{with } \beta > 0 \text{ and } \alpha > 1 \tag{1.6}$$

Lin et. al. in [1] propose a semi-analytic method for finding values of the y_k to minimize (1.4). Their approach allows the number of PM intervals n to be treated as an optimization variable. In the first part of this note we shall use standard nonlinear optimization methods to minimize mean cost for different values of n in order to find the optimum number of PM intervals by explicit enumeration.

Before proceeding any further, however, we must point out that the formulation above has not proved completely satisfactory in practice. When applying numerical optimization methods to (1.4), failures can occur when an iteration takes an exploratory step which causes one or more of the y_k to become negative. Such values for effective age of the system obviously have no physical meaning but there is nothing in the mathematical formulation of our problem to prevent them from occurring. The function $C(y)$ is unbounded below if negative values of y_k are allowed and hence an optimization step which yields a negative $C(y)$ may be accepted by the linesearch. Once a solution estimate has been accepted which has one or more $y_k < 0$, the optimization process will never be able to recover to produce a practical solution with all the y_k positive.

It is also worth pointing out that if we use non-integer values for the shape parameter α in (1.6) then the iterations will fail if any y_k becomes negative because then $C(y)$ is not computable.

One way to prevent such failures is to introduce the transformation $y_k = u_k^2$ and then carry out the optimization in terms of the new variables u_k, which are free to take positive or negative values. Thus, in what follows, we shall consider the problem

$$\text{Minimize} \quad \tilde{C}(u) = \frac{\gamma_r + (n-1) + \gamma_m \sum_{k=1}^{n} [H_k(u_k^2) - H_k(b_{k-1} u_{k-1}^2)]}{u_n^2 + \sum_{k=1}^{n} (1 - b_k) u_k^2}. \tag{1.7}$$

The determination of hazard-rate functions in practical situations is a non-trivial problem which will not be addresed here. For our demonstration examples we shall consider hazard rates of the form

$$h(t) = \beta_1 t^{\alpha-1} + \beta_2; \quad \text{with } \beta_1,\ \beta_2 > 0 \text{ and } \alpha > 1, \tag{1.8}$$

for various choices of α, β_1 and β_2. In all cases we shall use the same constants a_k and b_k as are given in [1]. Thus, for $k = 0, 1, 2, .., n-1$,

$$a_k = \frac{6k+1}{5k+1}, \quad b_k = \frac{k}{2k+1}. \tag{1.9}$$

We will use the cost ratios

$$\gamma_m = 10 \text{ and } \gamma_r = 1000 \tag{1.10}$$

corresponding to a system which is very much more expensive to replace than to repair or maintain.

We shall consider four examples which are defined by (1.8) - (1.10) together with particular values of α, β_1 and β_2.
Problem **SPM0** has $\alpha = 2, \ \beta_1 = 2, \ \beta_2 = 1$
Problem **SPM1** has $\alpha = 2, \ \beta_1 = 1, \ \beta_2 = 2$
Problem **SPM2** has $\alpha = 2.5, \ \beta_1 = 1, \ \beta_2 = 2$
Problem **SPM3** has $\alpha = 1.5, \ \beta_1 = 1, \ \beta_2 = 2$
For **SPM0** and **SPM1** the hazard rates are linear while for **SPM2** and **SPM3** they are, respectively, convex and concave functions of time.

2 Minimizing (1.7) for fixed values of n

For each of the cases **SPM0** – **SPM3** we minimize (1.7) for various values of n. These minimizations are done using Newton's method and taking the arbitrary initial guess

$$u_1 = u_2 ... = u_n = 1. \tag{2.1}$$

Gradients and Hessians of $\tilde{C}(u)$ are obtained using the fortran90 module `oprad` [2] [3] which implements a reverse accumulation approach for automatic differentiation. While analytical derivatives of (1.7) would not be particularly hard to obtain, the use of automatic differentiation makes it very convenient to use the quadratically convergent Newton technique. Moreover, the interface with `oprad` makes it virtually painless to do the coding modifications which result from changes to the PM model and objective function that will be described in later sections.

As a typical illustration we consider **SPM1** when $n = 7$. Figure 1 shows the solution as a plot of effective age against time with an instantaneous decrease taking place every time PM occurs. We can see that the system becmes *effectively younger* at each successive PM. This is to compensate for the fact that the hazard rate function is multiplied by a factor $a_k \geq 1$ after each PM. We also observe – as would probably be expected – that the intervals between PM become shorter over the lifetime of the system.

For **SPM1** with $n = 7$, the Newton method minimizes (1.7) from the initial guess (2.1) in eight iterations. On the first four of these iterations the Hessian matrix is

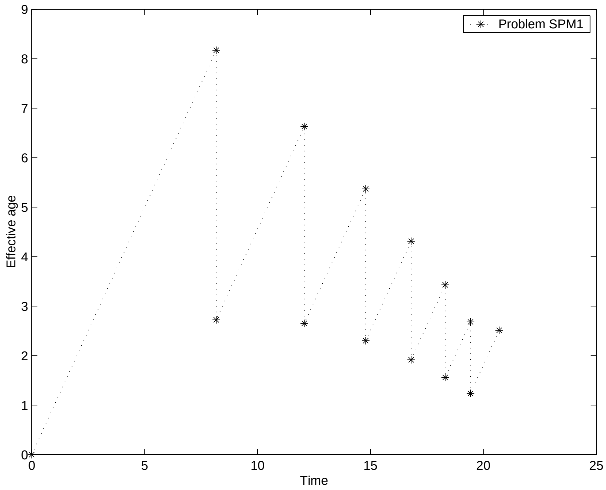

Figure 1: Optimal solution to **SPM1** for $n = 7$

found to be non-positive-definite. This shows that the function (1.7) is non-convex and may therefore have a number of local minima. In fact it certainly *does* have several local minima since, for any optimal point u^*, other equivalent optima can be found simply by changing the sign of one or more of the elements of u^*. However all such local optima give the same value of $\tilde{C}$ and for practical purposes do not represent alternative solutions to the scheduling problem. We need to consider whether there are other local minima of (1.7) with different values of $\tilde{C}$.

To investigate whether the maintenance scheduling problem has multiple solutions, we have applied the global optimization method DIRECT – as proposed by Jones et al [4] – to the function (1.7). DIRECT is a derivative-free algorithm which searches for the global minimum within a hyperbox defined by upper and lower bounds on the variables. It works by systematic subdivision of the initial box into smaller and smaller regions, but is quite efficient because it concentrates its explorations on regions which are judged *potentially optimal*. This technique has proved quite effective on practical problems (see, for instance, [5], [6]). The original paper [4] does not give a precise stopping rule and the usual practice is either simply to run the algorithm for a fixed number of iterations or to terminate when no improvement in the best function value is obtained over a specified number of subdivisions. It has been observed [5] [6] that it can be beneficial to *restart* the algorithm periodically. This involves setting up a new hyperbox which is centred on the best point found so far and beginning the subdivision process all over again. Quite often, the estimate of the global minimum obtained after several such restarts is better than the one reached by doing the same total number of DIRECT iterations from the original starting point.

We have applied DIRECT to (1.7) in the following way. After obtaining a solution

$u_1^*,..,u_n^*$ (e.g. by Newton's method) we have used DIRECT to explore the region

$$0 \le u_i \le 2\bar{u} \quad \text{where} \quad \bar{u} = \frac{1}{n}\sum_{i=1}^{n} u_i^*.$$

To date we have not found any better local minimum of $\tilde{C}$ in such a region and we therefore feel justified in assuming that Newton's method is indeed finding the global minimum of mean cost for each n. Therefore we now go on to consider the effect of changing n, the number of applications of PM.

3 Minimizing (1.7) for varying n

The figures below show plots of minimum values of (1.7) against n. Figure 2 shows that **SPM0** has a unique minimum at $n = 13$. This agrees with behaviour reported in [1] for hazard rate functions of the form (1.6).

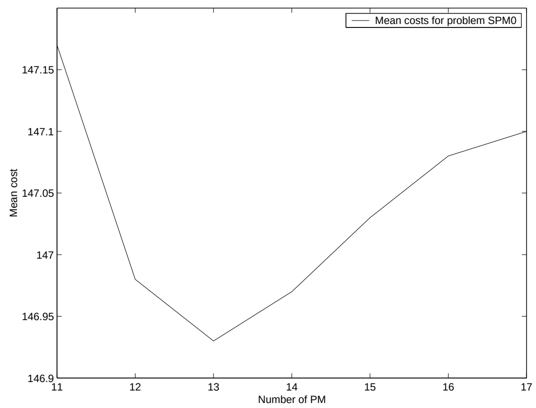

Figure 2: Solutions of **SPM0** for various n

Figures 3 - 5 show that, when there is a constant term in the definition of h, as in (1.8), there may be more than one local minimum of optimal mean cost for varying n. In problem **SPM1** (Figure 3) there are two almost equal minima. Figures 4 and 5 show that for **SPM2** the better result occurs at the larger value of n while for **SPM3** this situation is reversed.

This ambiguity about the optimal PM schedules is, however, more apparent than real. When the y_k values at the optimum for **SPM1** with $n = 12$ are substituted in (1.3) we get some PM intervals, x_k, which are negative. Thus it is only the solution with $n = 9$ that provides a practical PM strategy. Similar remarks apply to the **SPM2** result with $n = 14$ and the **SPM3** result with $n = 8$. Hence we deduce that constraints should be added to the problem of minimizing (1.7) in order to avoid spurious solutions with negative PM intervals.

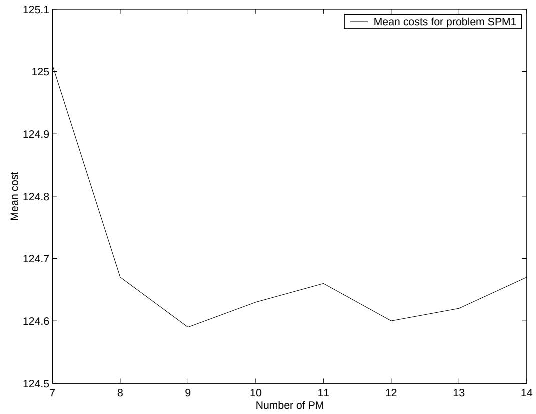

Figure 3: Solutions of **SPM1** for various n

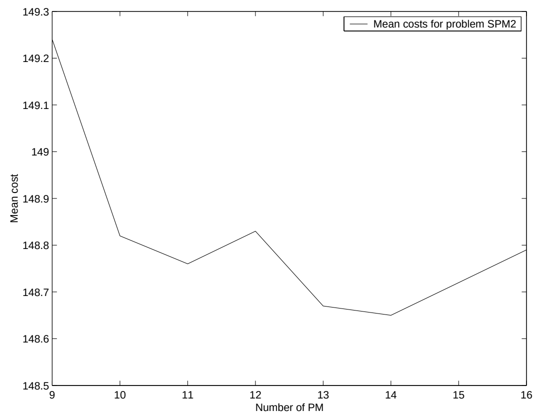

Figure 4: Solutions of **SPM2** for various n

4 Adding constraints to the minimization of (1.7)

The time intervals between each PM are given by (1.3) and to keep these positive we should consider the problem

$$\text{Minimize} \quad \tilde{C}(u) \quad \text{s.t.} \quad u_k^2 - b_{k-1}u_{k-1}^2 \geq 0, \; k = 2,..,n. \tag{4.1}$$

We can solve (4.1) by applying a sequential quadratic programming algorithm (for instance the quasi-Newton method described in [7]). The relationship between solutions of (4.1) and n for **SPM1** is shown in Figure 6. Solutions are unconstrained when $n < 11$, but when $n = 10 + j$ the last j of the inequalities in (4.1) are binding and the optimal values of $\tilde{C}$ are then such that the solution at $n = 9$ is clearly the best. Similar behaviour can be seen at solutions of (4.1) for **SPM2** and **SPM3**.

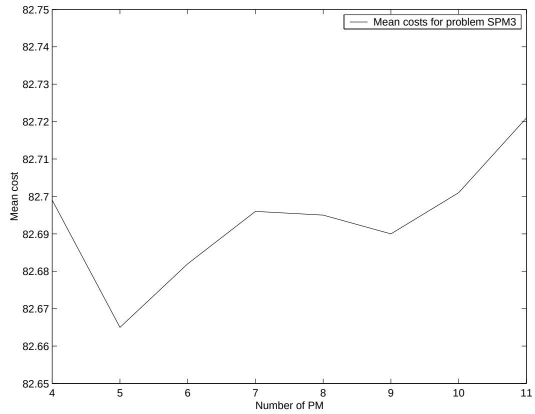

Figure 5: Solutions of **SPM3** for various n

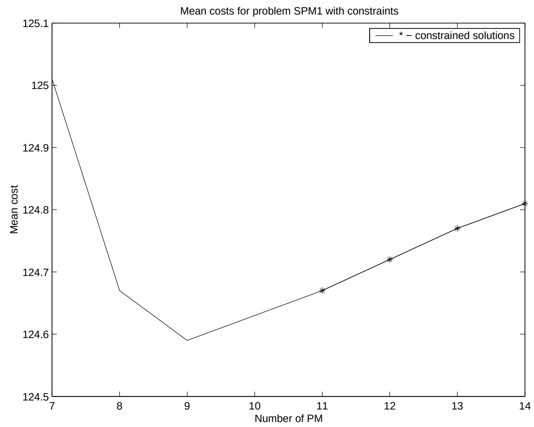

Figure 6: Solutions of **SPM1** (with constraints) for various n

In practical terms, the constraint in (4.1) may be considered rather weak since we would not want intervals between maintenance to become arbitrarily small. Hence we could replace (4.1) by the more general problem

$$\text{Minimize} \quad \tilde{C}(u) \;\; \text{s.t.} \;\; u_k^2 - b_{k-1}u_{k-1}^2 \geq \alpha u_1^2, \; k = 2, .., n. \tag{4.2}$$

for some $\alpha < 1$. This prevents PM intervals from becoming shorter than some specified fraction of the time between the system enetering service and its first scheduled maintenance.

5 Formulating the problem in terms of PM intervals

Another – and probably better – way of avoiding spurious solutions with some $x_k < 0$ is to treat the intervals between PM as the optimization variables rather than the effective ages. More specifically, if we let $v_1, .., v_n$ be optimization variables we can define a mean cost function $\bar{C}(v)$ as follows. First we set

$$y_1 = x_1 = v_1^2 \tag{5.1}$$

and then, for $k = 2, ..., n$

$$x_k = v_k^2; \quad y_k = b_{k-1} y_{k-1} + x_k. \tag{5.2}$$

This will ensure that the x's and y's are all non-negative. Now we can evaluate $C(y)$ by (1.4) and set

$$\bar{C}(v) = C(y). \tag{5.3}$$

We can solve the PM scheduling problem by finding the unconstrained minimum of $\bar{C}(v)$ using the Newton method as we did for the problem involving $\tilde{C}$ given by (1.7). As mentioned earlier, the use of the automatic differentiation tool `oprad` [2], [3] simplifies the process of obtaining $\nabla\bar{C}$ and $\nabla^2\bar{C}$. As might be expected, minimizing (5.3) gives the same PM schedules as those obtained by solving the constrained formulation (4.1).

We could use a similar approach when the intervals between PM are bounded below. We would simply have to change the relationship between the x_k and the fictitious optimization variables v_k by writing

$$x_k = \alpha u_1^2 + v_k^2 \tag{5.4}$$

instead of the first expression in (5.2). With this change, the minimum of (5.3) would give the same PM schedule as the solution to (4.2).

6 Minimizing mean cost w.r.t. *n*

We now consider how to find the optimum number of PM intervals without resorting to explicit enumeration. To do this we introduce an extra continuous variable ν which represents the number of PMs. In order to compute mean cost we need to consider how to deal with non-integer values of ν. In fact this can be done quite straightforwardly.

We let n denote the integer part of ν and we set $\theta = \nu - n$. Obviously $\theta < 1$ (but θ may become arbitrarily close to 1). We now suppose that $n-1$ *complete* PMs and one *partial* PM are performed before the replacement of the system. The partial maintenance is regarded as the n-th PM and it can be supposed to reduce the effective age of the system to

$$y_n - \theta(y_n - b_n y_n) = (1-\theta+\theta b_n) y_n = \tilde{b}_n y_n$$

instead of $b_n y_n$. The $(n+1)$-th PM is a system replacement and it takes place when the effective age is y_{n+1}. The (relative) cost of repairs between t_{n-1} and t_{n+1} is

$$\gamma_m [H_n(y_n) - H_n(b_{n-1} y_{n-1}) + H_{n+1}(y_{n+1}) - H_{n+1}(\tilde{b}_n y_n)].$$

The time elapsed between the last full PM and the replacement of the system is

$$y_n - b_{n-1} y_{n-1} + y_{n+1} - \tilde{b}_n y_n.$$

Using these ideas, the mean cost is determined as follows. We must choose a value N which represents the maximum number of PMs that can take place. We then need optimization variables $y_1, \ldots, y_N$ and ν. We then perform the following calculations.

$$n = \lfloor \nu \rfloor; \quad \theta = \nu - n; \;\; \tilde{b}_n = 1 - \theta + \theta b_n \tag{6.1}$$

$$R_c = \gamma_r + (\nu - 1) + \gamma_m \sum_{k=1}^{n} [H_k(y_k) - H_k(b_{k-1} y_{k-1})] + \gamma_m [H_{n+1}(y_{n+1}) - H_{n+1}(\tilde{b}_n y_n)] \tag{6.2}$$

$$T = y_n + \sum_{k=1}^{n-1} (1 - b_k) y_k + y_{n+1} - \tilde{b}_n y_n. \tag{6.3}$$

$$C(y, \nu) = \frac{R_c}{T}. \tag{6.4}$$

We want to minimize $C(y, \nu)$ subject to the constraint that PM intervals are non-negative. Therefore we require

$$y_k - b_{b-1} y_{k-1} \geq 0 \;\text{ for }\; k = 1, .., n-1 \quad \text{and} \;\; y_{n+1} - \tilde{b}_n y_n \geq 0. \tag{6.5}$$

Moreover we also want ν to be an integer value and so we require

$$\theta(1 - \theta) = 0. \tag{6.6}$$

Now $C(y, \nu)$ is a continuous *but non-differentiable* function of $y_1, .., y_N$ and ν. Specifically, there are jumps in derivatives because

$$\frac{\partial C}{\partial y_k} = 0 \;\text{ for }\; \nu < k - 1; \qquad \frac{\partial C}{\partial y_k} \neq 0 \;\text{ when }\; \nu \geq k - 1. \tag{6.7}$$

Moreover the number of constraints (6.5) depends on ν. Hence the problem of minimizing (6.4) subject to (6.5), (6.6) cannot be solved by a standard SQP method. A possible alternative is to minimize a non-differentiable exact penalty function of the form

$$C(y, \nu) + \rho_1 \sum_{k=2}^{n} |(y_k - b_{k-1} y_{k-1})_-| + \rho_1 |(y_{n+1} - \tilde{b}_n y_n)_-| + \rho_2 |\theta(1-\theta)|. \tag{6.8}$$

where $(z)_-$ denotes $\mathrm{Min}(0,z)$.

To ensure the positivity of the y values we could employ the $y = u^2$ transformation and minimize

$$\tilde{C}(u,\nu) + \rho_1 \sum_{k=2}^{n} |(u_k^2 - b_{k-1}u_{k-1}^2)_-| + \rho_1 |(u_{n+1}^2 - \tilde{b}_n u_n^2)_-| + \rho_2 |\theta(1-\theta)|. \quad (6.9)$$

However, a more elegant approach which avoids the constraints (6.5) would be to consider an extension of the cost function $\bar{C}(v)$ defined in (5.1) – (5.3) so as to include the extra variable ν. We would calculate $\bar{C}$ by first setting

$$x_1 = v_1^2; \;\; y_1 = x_1;$$

and then, for $k = 2,..,n$,

$$x_k = v_k^2; \quad y_k = b_{k-1}y_{k-1} + x_k.$$

We then use (6.1) – (6.3) and finally set

$$\bar{C}(v,\nu) = \frac{R_c}{T}. \quad (6.10)$$

Penalty terms to enforce non-negativity of the PM intervals are now not needed and so the scheduling problem is to minimize (6.10) subject only to the equality constraint (6.6). This can be solved by minimizing

$$\bar{C}(v,\nu) + \rho_2 |\theta(1-\theta)|. \quad (6.11)$$

We can handle the (global) minimization of the non-smooth functions (6.8), (6.9) or (6.11) by using the non-gradient algorithm DIRECT [4]. A global optimization method may be necessary because it is possible that functions including a penalty term $\rho_2|\theta(1-\theta)|$ may have local minima when $\theta \approx 0$ or $\theta \approx 1$ for different values of the variable ν.

6.1 Minimizing (6.11)

We now apply DIRECT to the minimization of (6.11). We propose the following, semi-heuristic approach, based on restarts, which has proved quite effective in practice.

Algorithm A

Choose a range $n_{min} \le n \le N$

Choose starting values $\hat{v}_k$, $k = 1,...,N$.

Set starting value

$$\hat{\nu} = \frac{n_{min} + N}{2}.$$

Choose initial box-size $\pm\Delta\nu_k,\ \pm\Delta\nu$, for DIRECT as

$$\Delta\nu_k = 0.99\hat{\nu}_k,\ \ k = 1,..,N;\quad \Delta\nu = \frac{N - n_{min}}{2}.$$

After M iterations of DIRECT perform a *restart* by re-centering the search on (ν_k^*, ν^*) – the best point found so far. The box-size is then reset to

$$\Delta\nu_k = Max(1, 0.99\nu_k^*),\ \ k = 1,..,N;\ \ \Delta\nu = Min(\nu^* - n_{min}, N - \nu^*)$$

Re-starts continue until a cycle of M iterations of DIRECT produces a change less than 0.01% in the value of $\bar{C}$.

Algorithm A was applied to **SPM1** – **SPM3** with $n_{min} = 1,\ N = 20$ and $M = 100$. The starting guessed values for the $\hat{\nu}_k$ were

$$\hat{\nu}_1 = 5,\ \ \hat{\nu}_k = Max(0.9\hat{\nu}_{k-1}, 1),\ \ k = 2,..,N$$

and the penalty parameter in (6.11) was $\rho_2 = 0.1$. Results are shown in Table 1.

	$\bar{C}$	Number of PM	DIRECT iterations	Restarts
SPM1	124.59	9	400	3
SPM2	148.76	11	500	4
SPM3	82.665	5	300	2

Table 1: Scheduling solutions with Algorithm A

In each case the optimum agrees with what was obtained by minimizing $\tilde{C}$ by Newton's method for successive fixed values of n.

By means of Algorithm A we have been able to follow the spirit of the work described in [1] and treat the number of PM applications as an optimization variable (rather than using explicit enumeration as in the previous section of this paper). This numerical, rather than analytical, approach should still be applicable when the hazard rate function has a less-simple form than those considered in [1] and in this paper.

Further trials with Algorithm A show that it is relatively insensitive to the choice of n_{min} and N. Broadly speaking, if the range for n is increased (e.g. $n_{min} = 1,\ N = 30$) the number of restarts needed for cases **SPM1** – **SPM3** is one more than in Table 1. Similarly, the number of restarts can be reduced by one when n lies in a smaller range such as $4 \le n \le 12$.

In our examples, the mean- cost function (6.11) is rather flat in the vicinity of the optimum. Thus, for **SPM2**, the minimum value of $\bar{C}$ when $n = 10$ is 148.83 which is only slightly worse than the global minimum 148.76 when $n = 11$. Because of this, the arbitrary choices made in Algorithm A (such as $M = 100$) may sometimes cause it to terminate with a value of ν corresponding to a number of PMs which differs slightly from the true global solution. In practical terms this is unlikely to represent a significant increase in mean-cost.

Some further experience with Algorithm A is given in the next subsection.

6.2 Sensitivity of PM to relative repair and replacement cost

We now consider what happens to the optimal number of PMs as γ_m, the relative cost of minimal repair, increases and decreases. In particular we shall take $\gamma_m = 100$ and $\gamma_m = 1$ to compare with the choice $\gamma_m = 10$ used in the examples already solved. Using Algorithm A we get the results in Table 2.

	$\gamma_m = 1$	$\gamma_m = 10$	$\gamma_m = 100$
SPM1	32.426 (13)	124.59 (9)	572.03 (5)
SPM2	46.826 (17)	148.76 (11)	592.2 (6)
SPM3	15.123 (7)	82.665 (5)	500.64 (3)

Table 2: Optimum mean cost and number of PM intervals for varying γ_m

The optimal number of PM applications increases as the repair cost comes closer to the cost of preventive maintenance. Conversely, the number of PMs decreases as the relative cost of repair increases.

Table 3 shows how changes to the relative replacement cost influence the optimal number of PMs. As before we use $\gamma_m = 10, \gamma_r = 1000$ as a reference. Clearly n increases and decreases with γ_r.

	$\gamma_r = 500$	$\gamma_r = 1000$	$\gamma_r = 2000$
SPM1	96.65 (7)	124.59 (9)	163.35 (11)
SPM2	109.93 (9)	148.76 (11)	205.59 (13)
SPM3	70.27 (4)	82.665 (5)	98.16 (6)

Table 3: Optimum mean cost and number of PM intervals for varying γ_r

7 A differentiable alternative to (6.11)

We have already noted that the heuristic elements in Algorithm A mean that there is no guarantee of convergence. As a consequence, it has sometimes happened during our computational tests that the algorithm terminates with the number of PM intervals differing by 1 from the true optimum. Hence we now consider ways in which we can check – and possibly improve – a solution produced by Algorithm A.

One simple and pragmatic approach would be to *assume* that the number of PM intervals given by Algorithm A is within ± 1 of being optimal. We could then use the solution as a starting guess for applying Newton's method to $\bar{C}(v)$ with the three candidate values of n and pick the minimum with the lowest mean cost.

It seems more elegant, however, to use the same objective function both in Algorithm A and in any subsequent refinement process. Several authors ([8], [9], [10]) have suggested using an approximate global minimum located by DIRECT as a starting point for a rapidly convergent local minimization method. As a way of obtaining the optimum precisely, this is likely to be more efficient than persisting with many iterations of DIRECT which is not designed for fast ultimate convergence to high accuracy solutions.

The idea of refining a DIRECT solution by a fast local minimization method does not work well for function (6.11), however. The non-differentiability of (6.4) has already been noted in the paragraph containing (6.7). Therefore we cannot apply the rapidly-convergent Newton method – or even a quasi-Newton method – to (6.11). We must consider an alternative formulation of the problem that may help us to get around this difficulty.

We can retain the idea of a *partial maintenance* from section 6. If N is the largest number of PM to be permitted, then we can consider the optimization variables as being the effective ages $y_k,\ k = 1,..,N$ together with the extra quantities $\theta_k,\ k = 1,..,N$. The values of θ_k may lie between 0 and 1 and are used in the same way as θ in section 6 to allow each PM to be either complete or partial. We calculate the relative cost of maintenance between the k-th and $(k+1)$-th PM as follows. The k-th PM reduces the effective age from y_k to $\tilde{b}_k$ where

$$\tilde{b}_k = 1 - \theta_k + \theta_k b_k. \tag{7.1}$$

Hence the cost of repairs performed between time t_k and t_{k+1} is

$$\gamma_m[H_{k+1}(y_{k+1}) - H_{k+1}(\tilde{b}_k y_k)]. \tag{7.2}$$

The total cost of all the PMs can be written as

$$\sum_{k=1}^{N-1} \theta_k \tag{7.3}$$

and so the lifetime cost of the system is

$$R_c = \gamma_r + \sum_{k=1}^{N-1} \theta_k + \gamma_m[\sum_{k=1}^{N-1} H_{k+1}(y_{k+1}) - H_{k+1}(\tilde{b}_k y_k)]. \tag{7.4}$$

The life of the system is

$$T = y_N + \sum_{k=1}^{N-1} (1 - \tilde{b}_k) y_k. \tag{7.5}$$

R_c and T are now defined in terms of $y_1, ..., y_N$ and $\theta_1, ..., \theta_N$ and are continuously differentiable with respect to these variables. Expressions for the derivatives of

$$\tilde{C}(y, \theta) = \frac{R_c}{T} \tag{7.6}$$

with respect to $y_1, .., y_N$, $\theta_1, ..., \theta_N$ can be obtained fairly easily. Alternatively, these derivatives can be evaluated by `oprad`.

Differentiability of $\tilde{C}(y, \theta)$ is only part of the story, however. We need to minimize this function subject to the constraints

$$\theta_k (1 - \theta_k) = 0, \;\; k = 1, .., N \tag{7.7}$$

so that partial PMs do not appear in an optimum schedule. Minimizing (7.6) subject to restrictions (7.7) can be expected to produce a solution where for some $n \leq N$

$$\theta_k = 1, \;\; k = 1, .., n; \quad \theta_k = 0, \;\; y_k = y_{k-1}, \;\; k = n+1, .., N.$$

Dealing with (7.6), (7.7) by minimizing the exact penalty function

$$C(y, \theta) + \rho \sum_{k=1}^{N-1} |\theta_k (1 - \theta_k)|$$

is counter-productive for our purposes because this is still a non-smooth problem. Therefore we look at Fletcher's differentiable exact or *ideal* penalty function [11], [12]. This is designed to enable us to solve the general problem

$$\text{Minimize} \quad F(x) \quad \text{s.t.} \;\; c_i(x) = 0, \; i = 1, .., m$$

by unconstrained minimization of

$$E(x) = F - c^T (AA^T)^{-1} Ag + \rho c^T c \tag{7.8}$$

where $g = \nabla F(x)$ and A is the Jacobian matrix whose rows are the constraint normals $\nabla c_k(x)^T$ for $k = 1, .., m$.

For the particular problem of minimizing (7.6) subject to (7.7) the calculation of the second term in (7.8) is fairly easy. For clarity of explanation we suppose that $\nabla C(y, \theta)$ and $\nabla(\theta_k (1 - \theta_k)$ are partitioned do that the first N elements are partial derivatives w.r.t. the y_k and the next N are derivatives w.r.t. the θ_k. Then, clearly,

$\nabla\theta_k(1-\theta_k$ is a vector of all zeros except for $(1-2\theta_k)$ in position $N+k$. Hence Ag is a vector of length N whose k-th element is

$$(1-2\theta_k)\frac{\partial C}{\partial\theta_k}.$$

AA^T is an $N\times N$ diagonal matrix whose (k,k)-th element is $(1-2\theta_k)^2$. Hence the second term in (7.8) is

$$c^T(AA^T)^{-1}Ag=\sum_{k=1}^{N}\frac{\theta_k(1-\theta_k)}{1-2\theta_k}\frac{\partial C}{\partial\theta_k}.$$

In full, therefore, the function (7.8) for the PM scheduling problem is

$$E(y,\theta)=C(y,\theta)-\sum_{k=1}^{N}\frac{\theta_k(1-\theta_k)}{1-2\theta_k}\frac{\partial C}{\partial\theta_k}+\rho\sum_{k=1}^{N}\theta_k^2(1-\theta_k)^2. \tag{7.9}$$

(7.9) is a differentiable function whose global minimum determines an optimal PM schedule. In principle, this global minimum can be estimated using DIRECT and then refined by a fast local gradient method. In practice, unfortunately, matters are still not quite so simple, since (7.9) involves gradients of the original objective function (7.6). Therefore second and higher derivatives of $C(y,\theta)$ are involved in computing ∇E or $\nabla^2 E$ for use by quasi-Newton or Newton techniques. This procedural difficulty is not insurmountable, however, and ∇E can be obtained using automatic differentiation techniques as described in [13], [14]. Practical implementation of this approach remains a topic for further research.

8 Conclusions

In this paper we have considered maintenance scheduling problems similar to those discussed in [1] but we have taken a numerical rather than analytical approach to the optimization calculations. The potential benefit of this is to explore the computational issues that might be involved when the hazard rate functions are not such simple analytical expressions as the Weibull functions (1.6). In our numerical experiments we have used automatic differentiation tools to obtain gradients and Hessians of the performance functions. This has made it a very straightforward matter to implement changes in problem formulation.

The main contribution of this paper is the formulation of the PM scheduling problem given in section 6 which enables the number of PMs to be treated as a continuous optimization variable. A method of dealing with this problem formulation is given as Algorithm A which applies a *global* minimization technique to the non-smooth function (6.11). This has been used with some success on some demonstration examples.

Two lines for further investigation have been opened in this paper. The first, which is comparatively straightforward, is to consider the effect on PM scheduling of lower bounds on maintenance intervals, using the problem formulation (4.2) (or the equivalent unconstrained problem at the end of section 5). The second and more substantial area of work concerns the solution of the scheduling problem by minimizing (7.9) by means of a variant of Algorithm A which includes final refinement by means of a fast gradient method. There are some non-trivial implementation issues here in relation to the use of automatic differentiation to evaluate the penalty function E and its derivatives.

References

[1] Lin, Daming, Ming Zuo and Richard Yam, Sequential Imperfect Preventive Maintenance Models with Two Categories of Failure Modes, *Naval Research Logistics*, 48, 134-183, 2001

[2] Christianson, Bruce, Automatic Hessians by Reverse Accumulation, IMA Journal of Numerical Analysis, **12**, pp 135-150, 1992

[3] Brown, Stephen, PhD Thesis, University of Hertfordshire, 1995.

[4] Jones, D.R., C.D. Perttunen and B.E. Stuckman, Lipschitzian Optimization without the Lipschitz Constant, Journ. Opt. Theory & Applics, **79**, pp 157-181, 1993.

[5] Wilson, Simon, PhD Thesis University of Hertfordshire, 2003

[6] Bartholomew-Biggs, Michael, Parkhurst, Stephen and Wilson, Simon, Using DIRECT to Solve an Aircraft Routing Problem, Computational Optimization and Applications, **21**, pp 311-323, 2002.

[7] Bartholomew-Biggs, Michael, Recursive Quadratic Programming Methods Based on the Augmented Lagrangian, Mathematical Programming Study **31**, pp 21-41, 1987.

[8] Bartholomew-Biggs, Michael and Kane, Stephen, A Global Optimization Problem in Portfolio Selection, Presented at APMOD 2004, Brunel University, June 2004.

[9] Huyer, W. and Neumaier, A. Global Optimization by Multilevel Coordinate Search, J. Global Optimization **14**, pp 331-355, 1999.

[10] Jones, D.T. Private Communication

[11] Fletcher, R., A Class of methods for Nonlinear Programming with Termination and Convergence properties, J.Abadie(ed) *Integer and Nonlinear Programming*, pp 157-175, North-Holland, Amsterdam, 1970.

[12] Fletcher, R. and Lill, S.A., A Class of Methods for Nonlinear programming II: Computational Experience. J.B. Rosen, O.L. Mangasarian and K. Ritter, eds *Nonlinear programming*, Academic Press, 1972.

[13] Christianson, Bruce, A Geometric Approach to Fletcher's ideal Penalty Function, J. Optim. Theory Appl. **84**, pp 433-441, 1995.

[14] Christianson, Bruce, Reverse Accumulation of Functions Containing Gradients, Technical Report 278, Numerical Optimisation Centre, University of Hertfordshire, 1993