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

Abstract

We deal with the problem of scheduling preventive maintenance (PM) for a system so that, over its operating life, 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 age. It is also 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 [6]. In particular we consider numerical algorithms which may be more appropriate for hazard rate models which are less simple than those used in [6] 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 [19]. 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.

1 Introduction

In this paper we consider the scheduling of preventive maintenance (PM). This problem has been widely discussed in the literature over the last decade – see for example Dekker [1], Pham & Wang [2], Miyamoto et al [3], Tsai et al [5], Lin et al [6], Wang [7], Nakamura et al [8] and Qian et al [4], [9] and other references contained therein. These papers consider models and solution algorithms that can be used to determine PM schedules that optimize a measure of system performance – e.g., minimizing mean cost over a lifetime or maximizing lifetime per unit-cost.

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.

There are two sorts of intervention that can occur during the operating lifetime of a system. *Minimal repair* (e.g. replacement of broken components) can take place whenever a failure happens and we shall suppose that this is always done in such a way that the system is restored to the satisfactory state it was in just before failure took place. *Preventive maintenance* (e.g. cleaning, oiling, re-calibration) happens according to a pre-determined schedule and is not triggered by specific events. Such maintenance can be viewed as causing the system to have an *effective age* which is less than its calendar age. If PM could be so thoroughly performed as to restore the system to the state at the beginning of its lifetime then the effective age would be reduced to zero. This is referred to as *perfect* PM. On the other hand, if PM were simply a matter of replacing broken components (as in minimal repair) then there would be no reduction in effective age. In practice, PM can be thought of as being *imperfect* in that it does decrease the effective age of the system without making it as good as new.

These notions of effective age and imperfect maintenance have been discussed by Nakagawa [10], [11], [12] and by Kijima et al [13]. 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 $b_1 x_1$, 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

$$y = b_1 x_1 + x, \quad \text{where} \ \ 0 < x < t_2 - t_1.$$

Perfect PM would imply $b_1 = 0$ in the above equation while minimal repair corresponds to $b_1 = 1$. Hence imperfect PM is modelled by taking $0 < b_1 < 1$.

In what follows, we shall follow the notation and models chosen by Lin et al [6]. They take the view that, 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

$$\int_0^{x_1} h(x)dx + \int_0^{x_2} a_1 h(b_1 x_1 + x)dx.$$

Here $x_2 = t_2 - t_1$ is the interval between the first and second PM and $a_1 (\geq 1)$ is some system-dependent constant.

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

$$y_2 = b_1 x_1 + x_2$$

and immediately after maintenance this is reduced to $b_2 y_2$ for some b_2 such that $b_1 \leq b_2 \leq 1$. Thus, in the interval between the second and third PM (at time t_3) the effective age is

$$y = b_2 y_2 + x = b_2 b_1 x_1 + b_2 x_2 + x, \quad \text{where} \quad 0 < x < x_3 = t_3 - t_2.$$

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

$$\int_0^{x_3} a_2 a_1 h(b_2 y_2 + x)dx$$

for some $a_2 \geq 1$.

We can now generalise the above ideas and, 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

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

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

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

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

For consistency with notation used in [6] 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

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

3

$H_k(t)$ 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 [6] 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 particular 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. [6] use a semi-analytic method for finding values of the y_k to minimize (1.4). By setting to zero the first partial derivatives $\partial C/\partial y_k (k = 1, .., n)$, they derive a system of equations which can be solved explicitly to give $y_1, .., y_{n-1}$ in terms of y_n. Thus $C(y)$ is effectively reduced to a one-variable function which can be minimized using a standard direct-search method (like bisection) or by a rapidly convergent gradient-based approach like Newton's method.

Approaches like the one proposed in [6] are succesful when the functions $H_k(t)$ and $h_k(t)$ are sufficiently simple to permit the formulation of analytical expressions for $y_1, ..., y_{n-1}$ in terms of y_n. In practice, however, it may not be realistic to restrict

the hazard rate functions to such easily manageable forms as (1.6). Moreover, even when the $h(t)$ are given by relatively simple expressions, a solution algorithm like that in [6] will need to be modified whenever the form of $h(t)$ changes.

Analytic (or semi-analytic) approaches to the optimization of PM schedules are quite commonly used (e.g. [8], [9]). Reliance on techniques which are tailored to particular assumptions about the problem effectively ignores the existence of powerful and efficient general minimization algorithms (see for instance [14], [15], [16]). There are many numerical techniques which can be applied to minimize a function like (1.4) and most of them do not require strong restrictions to be imposed on the form of the $H_k(t)$ functions. In later sections of this paper we shall describe the use of some of these techniques.

In our remarks so far we have considered the minimization of $C(y)$ with respect to $y_1, ..., y_n$ where n, the number of PMs applied, is fixed. In practice we may also wish to determine an optimal value for n. One advantage of the approach in [6] is that – in the special case where hazard rates are given by (1.6) – it allows n to be treated as an independent optimzation variable. This is not so easy to do when applying a numerical optimization algorithm to (1.4) because n is an integer-valued variable, in contrast to the continuously varying $y_1, ..., y_n$. Therefore, in sections 2–5 below we shall use standard optimization techniques to minimize mean cost for several different values of n in order to obtain the optimum number of PM by explicit enumeration. In section 6, however, we shall show how the maintenance scheduling problem can be reformulated so that the optimal value of n can be obtained by a general-purpose numerical optimization algorithm.

Before proceeding any further, we must now 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 the 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 with one or more $y_k < 0$ has been accepted, 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 sections 2 and 3, we shall consider the

problem

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

In section 4 we shall deal with the positivity of the y_k by adding constraints to the minimization of (1.4).

Before proceeding to a discussion of numerical case-studies, it is worth stating clearly that, for practical systems, the determination of a suitable hazard-rate function and values for the constants $a_1, a_2, .., b_1, b_2, ..$ is a non-trivial matter. We do not propose to address it in this paper since our purpose is to explore some issues in the numerical minimization of functions like (1.7). Therefore we shall consider some demonstration examples involving artificial hazard rate functions of the form

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

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 [6]. Thus, for $k = 0, 1, 2, .., n-1$,

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

We will use the cost ratios

$$\gamma_m = 10 \ \text{ and } \ \gamma_r = 1000 \qquad (1.10)$$

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 $\alpha, \ \beta_1$ 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

$$u_1 = u_2 ... = u_n = 1. \qquad (2.1)$$

Gradients and Hessians of $\tilde{C}(u)$ are obtained using the fortran90 module `oprad` [17] [18] which implements a *reverse accumulation* approach for *automatic differentiation* (AD). The ideas and techniques of AD – which have become increasingly widely known over the last decade – provide a means of evaluating derivatives of numerical functions without the labour of forming and coding explicit analytical expressions. It works by repeated use of the chain rule applied to numerical values rather than symbolic expressions. In `oprad` this is achieved by operator overloading (although it can also done by a pre-processing approach). We now give a brief description of AD based on using overloaded operators in the code which evaluates the function which is to be differentiated.

The *forward accumulation* version of AD associates with each program variable v a vector $\dot{v}$ which holds partial derivatives of v w.r.t. the independent variables $x_1,..,x_n$. A pair $V = \{v, \dot{v}\}$ is a new data type, often called a *doublet*. Doublets X_k corresponding to each x_k are initialised so that $\dot{x}_k$ is the k-th column of the identity matrix. Real arithmetic operations are overloaded as in

$$X_i X_j = \{x_j x_j,\ x_i \dot{x}_j + x_j \dot{x}_i\}; \quad \sin(X_j) = \{sin(x_j),\ cos(x_j)\dot{x})j\}.$$

If code for evaluating $f(x_1,..,x_n)$ is re-written with real variables re-declared as doublets then the result will be a doublet $\{f, \dot{f}\}$ where $\dot{f}$ is the value of ∇f.

In the reverse accumulation form of AD, used by `oprad`, floating point operations are overloaded so as to create a trace of the program evaluation for $f(x_1,..,x_n)$. This trace consists of a list of elementary arithmetic steps, in the order that they were performed, together with addresses of the variables involved and values of their partial derivatives at that stage of the computation. For instance, for the operation $w = u/v$ we would record values of u, v and w and also the information

$$\frac{\partial w}{\partial u} = \frac{1}{v}, \quad \frac{\partial w}{\partial v} = -\frac{u}{v^2}.$$

A real-valued *adjoint* variable $\bar{v}$ is then associated with each program variable v. When f has been evaluated, its adjoint $\bar{f}$ is set to 1 and all other adjoints are set to zero. A reverse calculation through the program trace is then performed. For the forward step $w = u/v$, this reverse calculation would be

$$\bar{u} = \bar{u} + \frac{\partial w}{\partial u}\bar{w}, \quad \bar{v} = \bar{v} + \frac{\partial w}{\partial v}\bar{w}$$

which causes $\bar{u}$ to be equal to the partial derivative of f w.r.t. the program variable u. It follows that, at the end of the reverse pass, the adjoints $\bar{x}_k$ associated with the independent variables will contain the partial derivatives $\partial f / \partial x_k$.

To calculate second partial derivatives we need to combine the forward and reverse approaches. This means that the program variables and their adjoints are of type doublet. Christianson [17] shows that at the end of the reverse pass the doublet adjoint $\bar{X}_k = \{\bar{x}_k, \dot{\bar{x}}_k\}$ will contain the value of $\partial f / \partial x_k$ in its first component and

7

the k-th column of $\nabla^2 f$ in its second component. It is also shown in [17] that the arithmetic cost of evaluating the Hessian in this way is less than $6n + 4$ times the cost of evaluating the original function.

While analytical derivatives of (1.7) would not be particularly hard to obtain, the use of `oprad` makes it very convenient to use the quadratically convergent Newton technique. Moreover, it becomes 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.

We now consider the solution to **SPM1** when $n = 7$ in order to show a typical PM schedule. 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 becomes *effectively younger* at each successive PM. This is to compensate for the fact that the hazard rate function is multiplied by a factor $q_k \geq 1$ after each PM. We also observe – as would probably be expected – that the intervals between PM become shorter over the lifetime of the system.
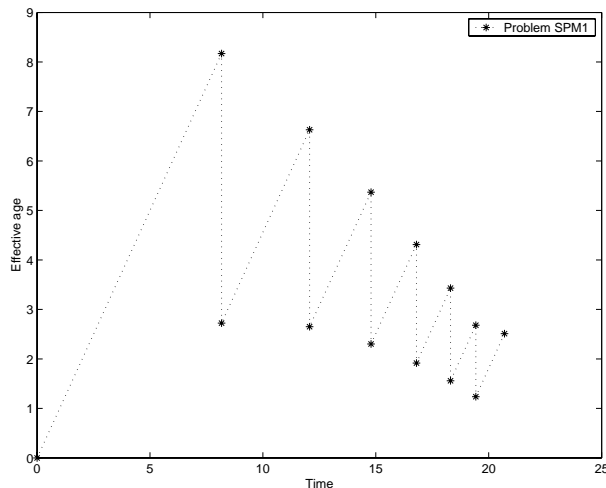

Figure 1: Optimal solution to **SPM1** for $n = 7$

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 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 $\tilde{C}$.

8

To investigate whether the maintenance scheduling problem has multiple solutions, we have applied the global optimization method DIRECT – as proposed by Jones et al [19] – 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. Being a deterministic method, it differs considerably from well-known stochastic techniques such as Simulated Annealing or Genetic Algorithms (which are used by Miyamoto et al [3] to solve a PM scheduling problem). DIRECT 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 effective on practical problems (see, for instance, [20], [21]). The original paper [19] 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 [20] [21] 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

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

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 [6] for hazard rate functions of the form (1.6).

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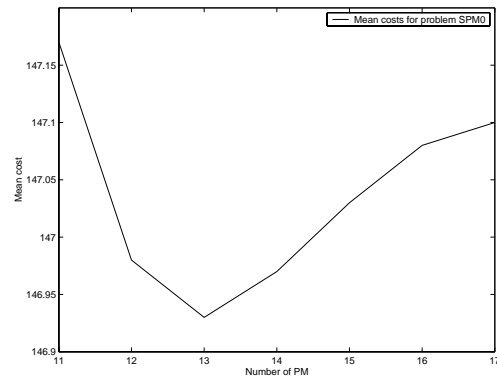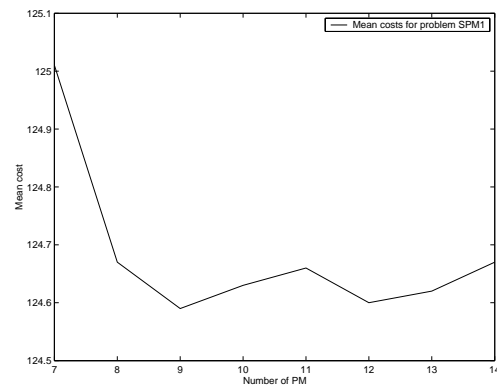
Figure 2: Solutions of **SPM0** for various *n*

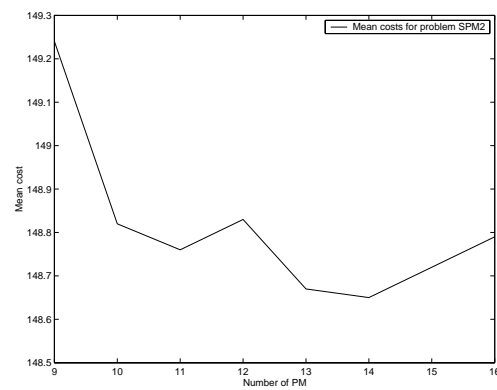Figure 3: Solutions of **SPM1** for various *n*

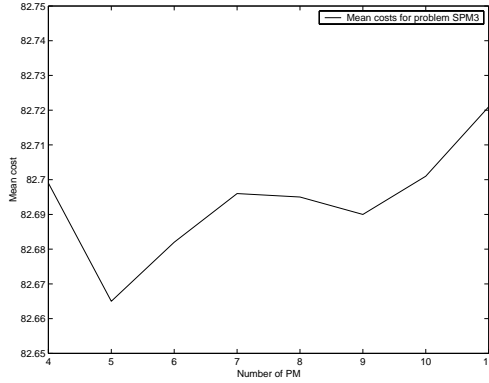Figure 4: Solutions of **SPM2** for various *n*

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

This ambiguity about the optimal PM schedules is, however, more apparent than real. At the optimum for **SPM1** with $n = 12$, substitution of the y_k values in (1.3) gives 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.

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

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

We can solve (4.1) by applying a sequential quadratic programming algorithm (for instance the quasi-Newton method described in [22]). 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 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**.

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

$$\text{Minimize} \quad \tilde{C}(u) \ \text{ s.t. } \ u_k^2 - b_{k-1}u_{k-1}^2 \geq \alpha u_1^2, \ k = 2,..,n. \tag{4.2}$$
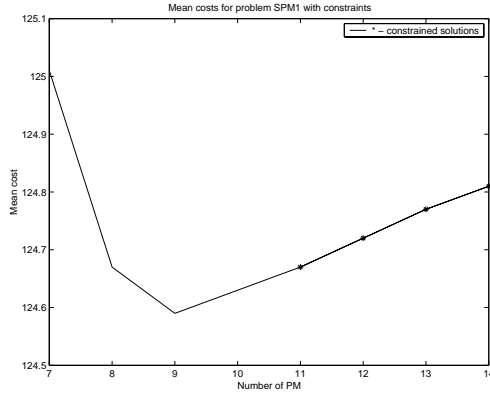
11

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

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` [17], [18] 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).

12

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. Taking the optimization variables as $y_1, ..., y_N$ and ν, we perform the following calculations.

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

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

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

13

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

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

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

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

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

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 sequential quadratic programming 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 $\text{Min}(0, z)$. For suitably large ρ_1 and ρ_2 a minimum of (6.8) coincides with a minimum of (6.4) subject to constraints (6.5), (6.6).

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

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

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 $k = 2, .., n$,

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

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

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

14

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

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

for suitably large ρ_2. 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 [19]. 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 a semi-heuristic approach, based on restarts, which has proved quite effective in practice.

Algorithm A

Choose a range $n_{min} \leq n \leq N$
Choose starting values $\hat{v}_k$, $k = 1, ..., N$.
Set starting value
$$\hat{\nu} = \frac{n_{min} + N}{2}.$$
Choose initial box-size $\pm \Delta v_k$, $\pm \Delta \nu$, for DIRECT as

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

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

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

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 $\hat{v}_k$ were

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

and the penalty parameter in (6.11) was $\rho_2 = 0.1$. Results are shown in Table 1. In each case the optimum agrees with what was obtained by minimizing $\tilde{C}$ by Newton's method for successive fixed values of n.

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. $n_{min} = 1$, $N = 30$)

	$\bar{C}$	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

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 $4 \leq n \leq 12$.

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.

By means of Algorithm A we have been able to follow the spirit of the work described in [6] and treat the number of PM applications as an optimization variable (rather than using explicit enumeration). This numerical, rather than analytical, approach should still be applicable when the hazard rate function has a less-simple form than those considered in [6] and in this paper.

6.2 Sensitivity of PM to relative repair and replacement cost

We now consider what happens to the optimal number of PMs as γ_n, the relative cost of minimal repair, increases and decreases. In particular we shall take $\gamma_n = 100$ 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. 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.

	$\gamma_m = 1$	$\gamma_m = 10$	$\gamma_m = 100$
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_n

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 Conclusions and further work

In this paper we have considered maintenance scheduling problems similar to those discussed in [6] 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.

A 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 successfully on demonstration examples.

An interesting area for future work concerns the replacement of the non-smooth function (6.11) by a differentiable alternative. This would enable us to use a variant of Algorithm A in which an approximate global minimum obtained by DIRECT is refined using a fast gradient method (as suggested in [23], [24], [25], for instance). Some initial work has been done in this area, which is summarised below.

Note first that *both* terms in (6.11) are non-differentiable. The non-differentiability of (6.4) is discussed in the paragraph containing (6.7) and the second (absolute value) term in (6.11) is also clearly non-differentiable when $\theta = 0$ or 1. The cost function (6.4) can be replaced by a differentiable alternative $\tilde{C}(y, \theta)$ involving extra optimization variables $\theta_1, .., \theta_N$ (see [26] for details). Each θ_k is used – like θ in section 6 – to allow the k-th PM to be treated as complete or partial. We then need to solve

$$\text{Minimize}_{y,\theta} \quad \tilde{C}(y, \theta) \quad \text{s.t.} \quad \theta_k(1 - \theta_k) = 0, \ k = 1, .., N \qquad (7.1)$$

in which both the objective function and the constraints are differentiable. In order to avoid a non-smooth constraint penalty term like that in (6.11) we can use Fletcher's differentiable exact or *ideal* penalty function [27], [28]. This is designed to solve the general problem

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

by unconstrained minimization of

$$E(x) = F - c^T (AA^T)^{-1} Ag + \rho c^T c \tag{7.2}$$

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 (7.1) the calculation of the second term in (7.2) is quite straightforward (see [26]) and we get

$$E(y, \theta) = \tilde{C}(y, \theta) - \sum_{k=1}^{N} \frac{\theta_k (1 - \theta_k)}{1 - 2\theta_k} \frac{\partial \tilde{C}}{\partial \theta_k} + \rho \sum_{k=1}^{N} \theta_k^2 (1 - \theta_k)^2. \tag{7.3}$$

as a differentiable function whose global minimum determines an optimal PM schedule. In principle, the global minimum of this function can be estimated using DIRECT and then refined by a fast local gradient method. In practice, however, matters are still not completely straightforward because the penalty function (7.3) involves first derivatives of $\tilde{C}$. Therefore second and higher derivatives of $\tilde{C}$ are involved in computing ∇E or $\nabla^2 E$ for use by quasi-Newton or Newton techniques. Fortunately `oprad` can be used to obtain both $\nabla \tilde{C}$ and $\nabla^2 \tilde{C}$ by reverse AD and therefore we have the means to calculate ∇E and use a quasi-Newton method to minimize $E(y, \theta)$.

Our use of (7.2) in the preceding paragraph has taken advantage of the relatively simple form of the equality constraints in (7.1). It is worth noting that, for more general problems, Christianson [29] has considered a parameter-free form of the Fletcher penalty function and has shown in [30] and [31] how AD can be used to obtain its derivatives.

References

[1] Dekker, Rommert, Applications of maintenance optimization models: a review and analysis, Reliability Engineering and System safety, **51**93) pp 229-240, 1996.

[2] Pham, H & H. Wang, Imperfect maintenance, European Journal of Operational Research **94**(3), pp 425-438, 1996.

[3] Miyamoto A., K. Kawamura & H. Nakamura, Bridge Management System and Maintenance Optimization for Existing Bridges, ComputerAided Civil and Infrastructure Engineering, 15, pp 45-55, 2000

[4] Qian, C., S. Nakamura & T. Nakagawa, Replacement policies for cumulative damage model with maintenance costs, Scientiae Mathematicae, 3, pp 117-126, 2000

[5] Tsai, Y.-T., K.-S.Wang & H.-Y. Teng, Optimizing preventive maintenance for mechanical componenets using genetic algorithms, Reliability Engineering and System Safety, **74** pp 89-97, 2001.

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

[7] Wang, H., A survey of maintenance policies of deteriorating systems, European Journal of Operational Research, **139**(3), pp 249-489, 2002.

[8] Nakamura S., C. Qian, I. Hayashi & T. Nakagawa, An optimal maintenance time of automatic monitoring system of ATM with two kinds of breakdowns, Computers and Mathematics with Applications, 46, pp 1095-1101, 2003

[9] Qian C., S. Nakamura S & T.Nakagawa, Replacement and minimal repair policies for a cumulative damage model with maintenance, Computers and Mathematics with Applications, 46, pp 1111-1118, 2003

[10] Nakagawa, T., Optimum policies when preventive maintenance is imperfect, IEEE Trans. Reliability, R-28, pp 331-331, 1979.

[11] Nakagawa, T., Sequential imperfect preventive maintenance policies, IEEE Trans. Reliability R-37, pp 295-298, 1988.

[12] Nakagawa, T., Imperfect preventive maintenance models, S. Osaki (ed) *Stochastic Models in Reliability and Maintenance*, pp 125-153, Springer, Berlin, 2002.

[13] Kijima, M., H. Marimura & Y. Suzuki, Periodical replacement problem without assuming minimal repair, European Journal of Operational Research, **37**(2), pp 194-203, 1988.

[14] Fletcher, R., Practical Methods of Optimization, 2nd ed. Wiley, 1987.

[15] Gill, P.E., W. Murray, and M. Wright. Practical Optimization. Academic Press, 1981.

[16] Bartholomew-Biggs, M.C. Nonlinear Optimization with Financial Applications, Kluwer Academic Publishers, 2004.

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

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

[19] 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.

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

[21] Bartholomew-Biggs, M.C., S.C. Parkhurst and S. Wilson, Using DIRECT to Solve an Aircraft Routing Problem, Computational Optimization and Applications, **21**, pp 311-323, 2002.

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

[23] Bartholomew-Biggs, M.C. and S. Kane, A Global Optimization Problem in Portfolio Selection, Presented at APMOD 2004, Brunel University, June 2004.

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

[25] Jones, D.T. Private Communication

[26] Bartholomew-Biggs, M.C., B. Christianson & M. Zuo, Optimizing preventive maintenance models, Numerical Optimization Centre Technical Report, University of Hertfordshire, 2005.

[27] 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.

[28] Fletcher, R. and S.A. Lill, 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.

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

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

[31] Bartholomew-Biggs, M.C., S. Brown, B. Christianson, and L.C.W. Dixon, Automatic Differentiation of Algorithms, J. Comp. and Applied Maths, **124** pp 171-190, 2000.