A General Approach for the Coordination of Maintenance Frequencies

R. Dekker, R. van Egmond,
J.B.G. Frenk and R.E. Wildeman
Erasmus University Rotterdam

Abstract

A maintenance activity carried out on a technical system often involves a system-dependent set-up cost that is the same for all maintenance activities carried out on that system. Grouping activities thus saves costs since execution of a group of activities requires only one set-up. By now, there are several multi-component maintenance models available in the literature, but most of them suffer from intractability when the number of components grows, unless a special structure is assumed. An approach that can handle many components was introduced in the literature by Goyal et al. However, this approach requires a specific deterioration structure for components. Moreover, the authors present an algorithm that is not optimal and there is no information of how good the obtained solutions are. In this paper we present an approach that solves the model of Goyal et al. to optimality. Furthermore, we extend the approach to deal with more general maintenance models like minimal repair and inspection that can be solved to optimality as well. Even block replacement can be incorporated, in which case our approach is a good heuristic.

Keywords: Maintenance, multiple components, global optimisation, Lipschitz optimisation

1 Introduction

The cost of maintaining a component of a technical system (a transport fleet, a machine, a road, a building, etc.) often consists of a cost that depends on the specific component involved and of a fixed cost that only depends on the system. In that case the system-dependent cost, the so-called set-up cost, is the same for all maintenance activities carried out on the system. For example, the set-up cost can consist of the down-time cost due to production loss if the system cannot be used during maintenance, or of the preparation cost associated with erecting a scaffolding or opening a machine. Set-up costs can be saved when maintenance activities are executed simultaneously, since execution of a group of activities requires only one set-up.

For a literature overview of the field of maintenance of multi-component systems, we refer to the review article by Cho and Parlar [3]. By now there are several methods that can handle multiple components. However, most of them suffer from intractability when the number of components grows, unless a special structure is assumed. For instance, the maintenance of a deteriorating system is frequently described using Markov decision theory (see, for example Howard [14], who was the first to use such a problem formulation). Since the state space in such problems grows exponentially with the number of components, the Markov decision modelling of multi-component systems is not tractable for more than three non-identical components (see, for example Bäckert and Rippin [1]). For problems with many components heuristic methods can be applied. For instance, Dekker and Roelvink [6] present a heuristic replacement criterion in case always a fixed group of components is replaced. Van der Duyn Schouten and Vanneste [17] study structured strategies, viz. (n, N)-strategies, but provide an algorithm for only two identical components.

An approach that can handle many components was introduced by Goyal and Kusy [10] and Goyal and Gunasekaran [9]. In this approach a basis interval for maintenance is taken and it is assumed that components can only be maintained at integer multiples of this interval, thereby saving set-up costs. The authors present an algorithm that iteratively determines the basis interval and the integer multiples. The algorithm has two disadvantages. The first is that only components with a very specific deterioration structure can be handled, which makes it impossible to apply it to well-known maintenance models. The second disadvantage is that the algorithm often gives solutions that are not optimal and that there is no information of how good the solutions are (see Van Egmond, Dekker and Wildeman [19]).

The idea of using a basic cycle time and individual integer multiples was first applied in the definition of the joint replenishment problem in inventory theory, see Goyal [7]. The joint replenishment problem can be considered as a very simple case of the maintenance problem of Goyal and Kusy [10]. A method to solve the joint replenishment problem to optimality was presented by Goyal [8]. However, this method is based on enumeration and is computationally prohibitive. Moreover, it is not clear how this method can be extended to deal with the more general cost functions in case of maintenance optimisation. Many heuristics have appeared in the joint replenishment literature (see Goyal and Satir [11]). But again, it is not clear how these heuristics will perform in case of the more general maintenance cost functions.

In this paper we present a general approach for the coordination of maintenance frequencies, thereby pursuing the idea of Goyal et al. [9, 10]. With this approach we can easily solve the model of Goyal et al. to optimality, but we can also incorporate many other maintenance models like minimal repair, inspection and block replacement. We can also efficiently solve the joint replenishment problem to optimality (see Dekker, Frenk and Wildeman [5]).

Our solution approach is based on global optimisation of the problem. We first apply a relaxation and find a corresponding feasible solution. This relaxation yields a lower bound on an optimal solution so that we can decide whether the feasible solution is good enough. If not so, we apply a global-optimisation procedure on an interval that is obtained by the relaxation and that contains an optimal solution. For the special cases of Goyal et al., the minimal-repair model and the inspection model it is then possible to apply Lipschitz optimisation to find a solution with an arbitrarily small deviation from an optimal solution in little time. For the block-replacement model we will apply a good heuristic.

This paper is structured as follows. In the next section we give the problem formulation. In Section 3 we rewrite the problem to be able to use solution techniques that will be discussed in Section 4. In Section 5 we present numerical results and in Section 6 we draw conclusions.

2 Problem Definition

Consider a multi-component system with components i, $i \in \{1, \ldots, n\}$. Creating an occasion for preventive maintenance on one or more of these components involves a set-up cost S, independent of how many components are maintained. The set-up cost can be due to, for example system down-time. Because of this set-up cost S there is an economic dependence between the individual components.

In this paper we consider preventive maintenance activities of the block type, that is, of which the next execution time depends only on the time passed since the latest execution. Otherwise, for example in case of age replacement, execution of maintenance can no longer be coordinated and one has to use opportunity policies or modified block replacement.

On an occasion for maintenance, component i can be preventively maintained at an extra cost of c_i^p. Let $M_i(x)$ be the expected total deterioration costs of component i (due to failures, repairs, etc.), x time units after its latest preventive maintenance. We assume that $M_i(\cdot)$ is continuous. We assume furthermore that after preventive maintenance a component can be considered as good as new. Consequently, the average costs $\Phi_i(x)$ of component i, when component i is preventively maintained on an occasion each x time units, amount to

$$\Phi_i(x) = \frac{c_i^p + M_i(x)}{x}, \; x > 0. \tag{1}$$

Since the function $M_i(\cdot)$ is continuous, the function $\Phi_i(\cdot)$ is also continuous.

To reduce costs by exploiting the economic dependence between components, maintenance on individual components can be combined. We assume that preventive maintenance is carried out at a basis interval of T time units (that is, each T time units an occasion for preventive maintenance is created) and that preventive maintenance on a component can only be carried out at integer multiples of this basis interval T: component i is preventively maintained each $k_i T$ time units, $k_i \in \mathbb{N}$. The objective is the minimisation of the total average costs per time unit. The total average costs are the sum of the average set-up cost and the individual average costs of each component. Consequently, we have the following problem:

$$\inf \left\{ \frac{S}{T} + \sum_{i=1}^{n} \Phi_i(k_i T) \ : \ k_i \in \mathbb{N},\ T > 0 \right\}. \tag{2}$$

In the formulation of this problem we assume that components are preventively maintained at fixed intervals that are integer multiples of a basis interval. This strategy is called *indirect grouping*, since the groups are formed indirectly and are not fixed over time. Another grouping strategy is *direct grouping*, where the n components are partitioned into disjoint groups that are fixed over time, that is, a component is then always maintained in the same group.

The disadvantage of direct-grouping models is that they can be solved to optimality for only a small number of components, unless a special structure is assumed (for example, when the frequency of a group of components equals the largest *individual* frequency among the components in that group, see Van Dijkhuizen and Van Harten [18]). Van Eijs, Heuts and Kleijnen [20] show that in an inventory environment (where the same grouping strategies apply for items that are jointly replenished), the indirect-grouping strategy always outperforms the direct-grouping approach. This is why we will consider indirect grouping in this paper.

The idea of modelling maintenance at fixed intervals that are integer multiples of a basis interval originates from inventory theory, see Goyal [7]. It was introduced in maintenance by Goyal and Kusy [10], and Goyal and Gunasekaran [9]. However, these authors allow only a very specific deterioration-cost function $M_i(\cdot)$ for component i, $i \in \{1, \ldots, n\}$. By allowing more general deterioration-cost functions, this modelling approach can also be applied to well-known preventive-maintenance strategies such as minimal repair, inspection and block replacement.

By choosing the appropriate function $M_i(\cdot)$, the following models can be incorporated (see also Dekker [4], who provides an extensive list of these models; here we only mention some important ones).

Special Case of Goyal and Kusy. Goyal and Kusy [10] use the following deterioration-cost function: $M_i(x) = \int_0^x (f_i + v_i t^e)dt$, where f_i and v_i are non-negative constants for component i and $e \geq 0$ is the same for all components. Notice that $e = 1$ represents the joint replenishment problem as commonly encountered in inventory literature, see also Dekker, Frenk and Wildeman [5].

Special Case of Goyal and Gunasekaran. The deterioration-cost function that is used by Goyal and Gunasekaran [9] is slightly different from that of Goyal and Kusy [10]. They take $M_i(x) = \int_0^{Y_i(x-X_i)} (a_i + b_i t)dt$, where x must of course be larger than X_i, and a_i, b_i, X_i and Y_i are non-negative constants for component i. In this expression, Y_i denotes the average utilisation factor of component i and X_i is the time required for maintenance of component i. Consequently, they take $e = 1$ in the deterioration-cost function of Goyal and Kusy, and they take individual down-time and utilisation factors into account.

Minimal-Repair Model. According to a standard minimal-repair model (see, for example Dekker [4]), component i is preventively replaced at fixed intervals of length x, with failure repair occurring whenever necessary. A failure repair restores the component into a state as good as before. Consequently, $M_i(x) = c_i^r \int_0^x r_i(t)dt$, where $r_i(\cdot)$ denotes the rate of occurrence of failures, and c_i^r the failure-repair cost. Here $M_i(x)$ expresses the expected repair costs incurred in the interval $[0, x]$ due to failures. Notice that this model incorporates the special case of Goyal and Kusy if we take $c_i^r = 1$ and $r_i(t) = f_i + v_i t^e$.

Inspection Model. In a standard inspection model (see Dekker [4]), component i is inspected at fixed intervals of length x, with a subsequent replacement when at inspection the component turns out to have failed. If a component fails before it is inspected, it stays inoperative until it is inspected. After inspection, a component can be considered as good as new. Here we have $M_i(x) =$

$c_i^u \int_0^x F_i(t)dt$, where c_i^u is the failure cost per unit time and $F_i(\cdot)$ is the cdf of the failure distribution of component i.

Block-Replacement Model. According to a standard block-replacement model (see Dekker [4]), component i is replaced upon failure and preventively after a fixed interval of length x. Consequently, $M_i(x) = c_i^f N_i(x)$, where $N_i(x)$ denotes the renewal function (expressing the expected number of failures in $[0, x]$), and c_i^f the failure-replacement cost.

In the following section we present a general approach to solve (2). This approach allows a different $M_i(\cdot)$ for each component. Thus it is also possible to mix the different models above; it is, for instance, possible to combine the maintenance of a component according to the minimal-repair model with the maintenance of a component according to an inspection model.

3 Analysis of the Problem

Goyal and Kusy [10] and Goyal and Gunasekaran [9] apply an iterative algorithm to solve the problem in the previous section for their specific deterioration-cost functions. The authors initialise each $k_i = 1$ and then find the corresponding optimal T by setting the derivative of the cost function of (2) as a function of T to zero. Subsequently, the authors find for each i a value of k_i, in two different ways. Goyal and Kusy [10] find for each i the optimal integer k_i belonging to T by looking in a table that is made in advance for each component and that gives the optimal k_i for disjoint ranges of T. Goyal and Gunasekaran [9] find for each i the optimal *real-valued* k_i by setting the derivative of the cost function of (2) as a function of k_i to zero and by rounding this real-valued k_i to the nearest integer. Once a value for k_i is found, it is compared to the k_i in the previous iteration (in this case the initialisation). When for each i the k_i in the two iterations are equal, the algorithm terminates. Otherwise a new optimal T is found for the current values of k_i, and subsequently new values of k_i are determined, and so on, until for all i the k_i in two consecutive iterations are equal.

The advantage of this algorithm is that it is fast. This is primarily due to the special deterioration structure of the components in the cases of Goyal et al., which makes it possible to find an analytical expression for the optimal T given values of k_i, and also to find a value for the k_i in little time.

The specific deterioration structure of the components is at the same time a great disadvantage, since there is little room for differently modelled components. It is possible to extend the algorithm to deal with the more general maintenance models given in the previous section, but in that case a value for an optimal T given values of k_i has to be computed numerically, and the same holds for the corresponding values of k_i. As a result, the algorithm will become much slower.

The greatest disadvantage of the algorithm of Goyal et al. is, however, that it is often stuck in a local optimal solution (see Van Egmond, Dekker and Wildeman [19]). There is no indication whatsoever of how good the solutions are when this occurs. This implies that even if we extend the algorithm to deal with more general maintenance models (which we will do anyway to study its performance in Section 5), we do not have any guarantee concerning the quality and optimality of the obtained solutions.

In inventory theory literature many heuristics have appeared for the special cost functions in the joint replenishment problem (see Goyal and Satir [11]). Though some heuristics can be modified to deal with the cost functions of maintenance optimisation, the performance of these heuristics cannot be guaranteed. That is why we will focus in this paper on an alternative solution approach that is based on the global optimisation of (2). We first apply a relaxation and find a corresponding feasible solution to the problem. This relaxation yields a lower bound on an optimal solution so that we can decide whether the feasible solution is good enough. If not so, we apply a global-optimisation procedure in an interval that is obtained by the relaxation and that contains an optimal T. For the special cases of Goyal et al., the minimal-repair model and the inspection model it is then possible to find a solution with an arbitrarily small deviation from an optimal solution. For the block-replacement model this is not possible, but application of a fast golden-section search heuristic yields a good solution as well. In all cases our approach outperforms that of Goyal et al. (in original form when the special functions are used, and in extended form otherwise). Our approach can also be applied to find an optimal solution to the joint replenishment problem, see Dekker, Frenk and Wildeman [5]. In that case the procedure can be made even more efficient, since the cost functions in the joint replenishment problem have a very simple form.

To make the optimisation problem (2) mathematically more tractable, we substitute T by $1/T$. Using this transformation, the relaxation that will be discussed in the next section becomes an easily solvable convex-programming problem if each of the individual cost functions $\Phi_i(\cdot)$ is given by one of the special cases of Goyal et al., the minimal-repair model or the inspection model. As will be shown later, one cannot generally derive that the relaxation is a convex-programming problem without this transformation.

Clearly, using the above transformation the optimisation problem (2) is then equivalent with

$$\inf\left\{ST + \sum_{i=1}^{n} \Phi_i(k_i/T) \ : \ k_i \in \mathbb{N},\ T > 0\right\} =$$

$$(P) \qquad \inf_{T>0}\left\{ST + \sum_{i=1}^{n} \inf\{\Phi_i(k_i/T) \ : \ k_i \in \mathbb{N}\}\right\}.$$

Denote now by $v(P)$ the optimal objective value of (P) and by $T(P)$ an optimal T (if it exists). Notice that if $T(P)$ and certain values of $k_1, k_2, \ldots, k_n$ are optimal for (P), then $T = 1/T(P)$ and the same values of $k_1, k_2, \ldots, k_n$ are optimal for the optimisation problem (2).

To simplify the optimisation problem (P), we introduce the following definition and assumption.

Definition 1 *A function $f(x)$, $x \in (0, \infty)$, is called unimodal with respect to $b \geq 0$ if $f(x)$ is decreasing for $x \leq b$ and increasing for $x \geq b$. That is, $f(y) \geq f(x)$ for every $y \leq x \leq b$, and $f(y) \geq f(x)$ for every $y \geq x \geq b$.*

Observe that by this definition it is immediately clear that any increasing function $f(x)$, $x \in (0, \infty)$, is unimodal with respect to $b = 0$.

Assumption 1 *For each $i = 1, \ldots, n$ the optimisation problem (P_i) given by $\inf\{\Phi_i(x) \ : \ x > 0\}$ has a finite optimal solution $x_i^* > 0$. Furthermore, for each i the function $\Phi_i(\cdot)$ is unimodal with respect to x_i^*.*

By Assumption 1 the optimisation problem (P) can be simplified considerably. To this end we introduce the interval $I_i^{(k)} := [k/x_i^*, (k+1)/x_i^*]$, $k = 0, 1, \ldots$, and we observe that for $t \in I_i^{(k)}$ it holds that $k/t \leq x_i^* \leq (k+1)/t$, so that

$$x_i^* \leq (k+1)/t \leq (k+2)/t \leq (k+3)/t \leq \cdots$$

and

$$x_i^* \geq k/t \geq (k-1)/t \geq \cdots \geq 1/t.$$

Therefore, as by Assumption 1 the function $\Phi_i(\cdot)$ is unimodal with respect to x_i^*, we have that

$$\Phi_i((k+1)/t) \leq \Phi_i((k+2)/t) \leq \Phi_i((k+3)/t) \leq \cdots$$

and

$$\Phi_i(k/t) \leq \Phi_i((k-1)/t) \leq \cdots \leq \Phi_i(1/t).$$

This implies that for a given t it is easy to determine an optimal integer k_i, since now we have that

$$\inf\{\Phi_i(k_i/t) \ : \ k_i \in \mathbb{N}\} = \begin{cases} \Phi_i(1/t) & \text{if } t \in I_i^{(0)} \\ \min\{\Phi_i(k/t), \Phi_i((k+1)/t)\} & \text{if } t \in I_i^{(k)},\ k = 1, 2, \ldots \end{cases}$$

Consequently, if we define

$$g_i(t) := \inf\{\Phi_i(k_i/t) \ : \ k_i \in \mathbb{N}\},$$

we have

$$g_i(t) = \begin{cases} \Phi_i(1/t) & \text{if } t \in I_i^{(0)} \\ \min\{\Phi_i(k/t), \Phi_i((k+1)/t)\} & \text{if } t \in I_i^{(k)},\ k = 1, 2, \ldots \end{cases} \tag{3}$$

It is not difficult to verify that by Assumption 1 and the fact that $\Phi_i(\cdot)$ is continuous, the function $g_i(\cdot)$ is also continuous. In Figure 1 an example of the function $g_i(\cdot)$ is given.

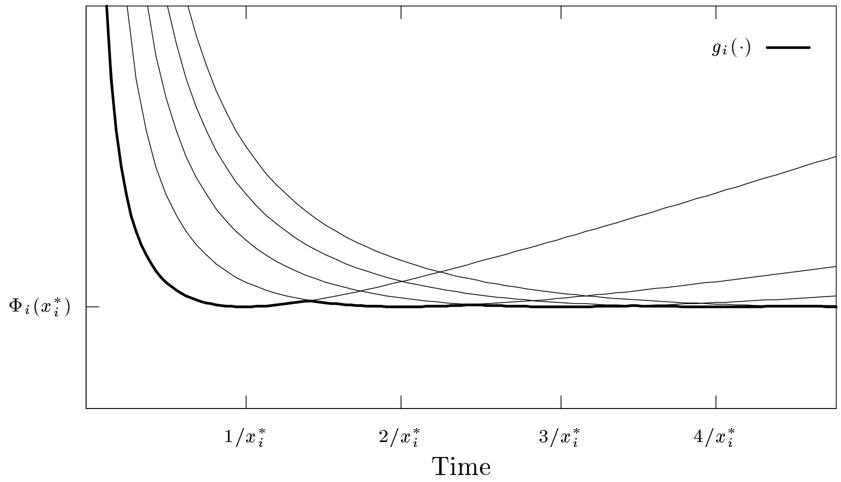

Figure 1. An example of the function $g_i(\cdot)$. The thin lines are the graphs of the functions $\Phi_i(1/t), \Phi_i(2/t), \ldots, \Phi_i(5/t)$. The (bold) graph of $g_i(\cdot)$ is the lower envelope of these functions.

Notice that for a given t, the function $g_i(t)$ and the corresponding k can easily be calculated once an optimal solution x_i^* of (P_i) is known. A given t is within the interval $I_i^{(k)}$ for which $k = \lfloor t x_i^* \rfloor$, with $\lfloor \cdot \rfloor$ denoting the lower-entier function. Consequently, if $k = 0$, one function evaluation (viz. of $\Phi_i(1/t)$) is necessary to compute $g_i(t)$. Otherwise, if $k \geq 1$, two function evaluations are necessary. Under Assumption 1 the optimisation problem (P) now has the following simplified representation:

$$(Q) \qquad \inf_{T>0} \left\{ ST + \sum_{i=1}^{n} g_i(T) \right\},$$

with $g_i(\cdot)$ given by (3). Denote by $v(Q)$ the optimal objective value of (Q) and by $T(Q)$ an optimal T (if it exists). Under Assumption 1 we have that the problems (P) and (Q) are equivalent, which implies that if $T(Q)$ and certain values of $k_1, k_2, \ldots, k_n$ are optimal for (Q), then $T(P) = T(Q)$ and the same values of $k_1, k_2, \ldots, k_n$ are optimal for problem (P) and, as a result, $T = 1/T(Q)$ and $k_1, k_2, \ldots, k_n$ are also optimal for the original optimisation problem (2). In the remainder we assume that Assumption 1 is satisfied and, consequently, we will consider from now on the optimisation of problem (Q).

Below we show that in general the special cases of Goyal et al., the minimal-repair model and the inspection model satisfy Assumption 1 when the optimisation problem (P_i) has a finite optimal solution $x_i^* > 0$. To verify this we need the following lemma.

Lemma 1 *If $M_i(\cdot)$ is concave on $(0, b_i)$ and convex on (b_i, ∞), $b_i \geq 0$, and the optimisation problem (P_i) given by $\inf\{\Phi_i(x) \ : \ x > 0\}$ has a finite optimal solution $x_i^* > 0$, then the function $\Phi_i(\cdot)$ is unimodal with respect to x_i^*.*

PROOF: If $b_i > 0$ and $M_i(\cdot)$ is concave on $(0, b_i)$ then $c_i^p + M_i(\cdot)$ is also concave on $(0, b_i)$. This implies that for every $0 < z_1 < z_2 < b_i$ we have that $c_i^p + M_i(z_1) = c_i^p + M_i((z_1/z_2)z_2) > (z_1/z_2)(c_i^p + M_i(z_2))$. Hence, by equation (1), $\Phi_i(z_1) > \Phi_i(z_2)$ and, consequently, $\Phi_i(\cdot)$ is strictly decreasing on $(0, b_i)$. Since x_i^* is a minimum for $\Phi_i(\cdot)$ this implies that $x_i^* \geq b_i$. If $b_i = 0$ then we also have that $x_i^* \geq b_i$, since $x_i^* > 0$.

Since $M_i(\cdot)$ is convex on (b_i, ∞), $c_i^p + M_i(\cdot)$ is also convex on (b_i, ∞), $b_i \geq 0$. By Theorems 3.51 and 3.21 of Martos [15] this implies that $\Phi_i(t) = (c_i^p + M_i(t))/t$ is unimodal on (b_i, ∞) with respect to x_i^*, and so, if $b_i = 0$, the desired result follows immediately. For the case $b_i > 0$ we showed above that $\Phi_i(\cdot)$ is strictly decreasing on $(0, b_i)$. Together with the result that the continuous function $\Phi_i(\cdot)$ is unimodal on (b_i, ∞) with respect to x_i^*, we then obtain that $\Phi_i(\cdot)$ is unimodal on the interval $(0, \infty)$ with respect to x_i^*, which completes the proof. □

Theorem 1 *If each (P_i), $i = 1, \ldots, n$, has a finite solution $x_i^* > 0$ and is formulated according to one of the special cases of Goyal et al., the minimal-repair model with a unimodal rate of occurrence of failures or the inspection model, then Assumption 1 is satisfied.*

PROOF: We will prove that if for a certain $i \in \{1, \ldots, n\}$ the optimisation problem (P_i) has a finite solution $x_i^* > 0$ for one of the models mentioned, then the function $\Phi_i(\cdot)$ is unimodal with respect to x_i^*. Consider therefore an arbitrary $i \in \{1, \ldots, n\}$ and distinguish between the different models.

1. Special Case of Goyal and Kusy. It is easy to show (by setting the derivative of $\Phi_i(\cdot)$ to zero) that the optimisation problem (P_i) has an optimal solution $x_i^* = \{(c_i^p(e+1))/(v_i e)\}^{1/(e+1)}$. This solution is finite and positive if and only if v_i, e and c_i^p are strictly larger than zero, and by the assumption that $x_i^* > 0$ we can assume that this is the case.

   We have that $M_i(x) = \int_0^x (f_i + v_i t^e)dt = f_i x + (v_i/(e+1))x^{e+1}$, so that $M_i''(x) = e v_i x^{e-1} > 0$ and, as a result, $M_i(\cdot)$ is (strictly) convex on $(0, \infty)$. By Lemma 1 we then have that $\Phi_i(\cdot)$ is unimodal with respect to x_i^*.

2. Special Case of Goyal and Gunasekaran. It is easy to show (by setting the derivative of $\Phi_i(\cdot)$ to zero) that the optimisation problem (P_i) has an optimal solution $x_i^* = \{2(c_i^p - a_i X_i Y_i)/(b_i Y_i^2) + X_i^2\}^{1/2}$. This solution is finite and positive if and only if b_i and Y_i are strictly larger than zero and $c_i^p > X_i Y_i (a_i - b_i X_i Y_i/2)$, and by the assumption that $x_i^* > 0$ we can assume that this is the case.

   We have that $M_i(x) = \int_0^{Y_i(x-X_i)} (a_i + b_i t)dt = a_i Y_i (x - X_i) + b_i Y_i^2 (x - X_i)^2/2$, so that $M_i''(x) = b_i Y_i^2 > 0$ and, as a result, $M_i(\cdot)$ is (strictly) convex on $(0, \infty)$. By Lemma 1 we then have that $\Phi_i(\cdot)$ is unimodal with respect to x_i^*.

3. Minimal-Repair Model. If the rate of occurrence of failures $r_i(\cdot)$ is unimodal with respect to a value $b_i \geq 0$, then as $M_i(x) = c_i^r \int_0^x r_i(t)dt$ it follows that $M_i'(\cdot)$ is decreasing on $(0, b_i)$ and increasing on (b_i, ∞). Hence $M_i(\cdot)$ is concave on $(0, b_i)$ and convex on (b_i, ∞). Since the optimisation problem (P_i) has a finite solution $x_i^* > 0$ we then have by Lemma 1 that $\Phi_i(\cdot)$ is unimodal with respect to x_i^*.

   Notice that if $b_i = 0$, then $r_i(\cdot)$ is increasing on $(0, \infty)$ and $M_i(\cdot)$ is convex on $(0, \infty)$ If $r_i(\cdot)$ is unimodal with respect to a b_i strictly larger than zero, then $\Phi_i(\cdot)$ follows a bathtub pattern. In Lemma 1 we showed that in that case $x_i^* \geq b_i$. As the function $M_i(\cdot)$ is convex on (b_i, ∞), it is a forteriori convex on (x_i^*, ∞), a result that will be used later to prove that the relaxation of (Q) is a convex-programming problem (see Lemma 2).

4. Inspection Model. Since $M_i(x) = c_i^u \int_0^x F_i(t)dt$, we have that $M_i'(x)$ is increasing on $(0, \infty)$, and hence that $M_i(x)$ is convex on $(0, \infty)$. Since the optimisation problem (P_i) has a finite solution $x_i^* > 0$ we then have by Lemma 1 that $\Phi_i(\cdot)$ is unimodal with respect to x_i^*.

Consequently, if for each $i = 1, \ldots, n$ one of the above models is used (possibly different models for different i), then $\Phi_i(\cdot)$ is unimodal with respect to x_i^* and so we have verified that Assumption 1 is satisfied. □

In Dekker [4] it is shown that a finite optimal solution x_i^* of the optimisation problem (P_i) for the minimal-repair model exists if $\lim_{t\to\infty} r_i(t) = \infty$, or if $\lim_{t\to\infty} r_i(t) = c$ and $\lim_{t\to\infty}\{ct - \int_0^t r(s)ds\} > c_i^p$ for some $c > 0$. A finite optimal solution for the inspection model model exists if $c_i^p < c_i^u EX_i$, with EX_i the expected lifetime of component i.

In Figure 2 an example of the objective function of problem (Q) is given. In general (Q) has several local minima, even for the simple models described above. This is due to the shape of the functions $g_i(\cdot)$ and it is inherent to the fact that the k_i have to be integer. In the following section we show that when these integrality constraints are relaxed, often a much easier problem is obtained; for the special cases of Goyal et al., the minimal-repair model and the inspection model this relaxed problem turns out to be a single-variable convex-programming problem that is easy to solve. The relaxation can be used to obtain an approximation of the optimal objective value $v(Q)$ and to determine an interval that contains an optimal value $T(Q)$. When the approximation is not good enough, we can apply global-optimisation techniques within this interval to find an optimal solution.

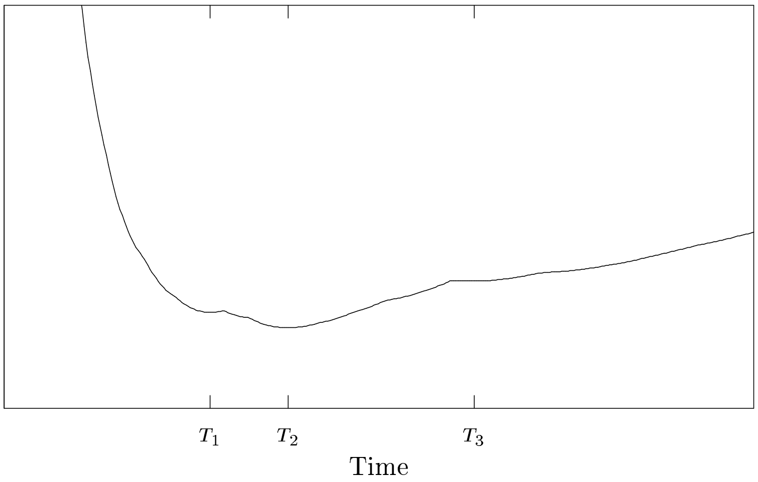

Figure 2. An example of the objective function of problem (Q). There are three local minima: at T_1, T_2 and T_3.

4 Solving Problem (Q)

In this section we develop a solution procedure for problem (Q). To this end we first consider a relaxation.

4.1 A Relaxation for (Q)

If we replace in (Q) the constraints $k_i \in \mathbb{N}$ by $k_i \geq 1$ (in the definition of $g_i(\cdot)$) then clearly $v(Q) \geq v(Q_{rel})$ with $v(Q_{rel})$ the optimal objective value of the optimisation problem given by

$$(Q_{rel}) \qquad \inf_{T>0}\left\{ ST + \sum_{i=1}^{n} \inf\{\Phi_i(k_i/T) \ : \ k_i \geq 1\}\right\}.$$

Let $T(Q_{rel})$ be a corresponding optimal value of T (if it exists).

To simplify this problem as we did in the previous section we need a much weaker assumption than Assumption 1.

Assumption 2 *For each $i = 1, \ldots, n$ the optimisation problem (P_i) given by $\inf\{\Phi_i(x) \ : \ x > 0\}$ has a finite optimal solution $x_i^* > 0$. Furthermore, for each $i = 1, \ldots, n$ it holds that $\Phi_i(\cdot)$ is increasing on $[x_i^*, \infty)$.*

Theorem 1 showed that for the special cases of Goyal et al., the minimal-repair model with a unimodal rate of occurrence of failures and the inspection model, Assumption 1 is satisfied when (P_i) has a finite solution $x_i^* > 0$. As a result, also Assumption 2 is satisfied for these models.

By Assumption 2 the optimisation problem (Q_{rel}) can be simplified. Analogously to equation (3) we have

$$g_i^{(R)}(t) := \inf\{\Phi_i(k_i/t) \ : \ k_i \geq 1\} = \begin{cases} \Phi_i(1/t) & \text{if } t < 1/x_i^* \\ \Phi_i(x_i^*) & \text{if } t \geq 1/x_i^*. \end{cases} \tag{4}$$

In Figure 3 an example of the function $g_i^{(R)}(\cdot)$ is given.

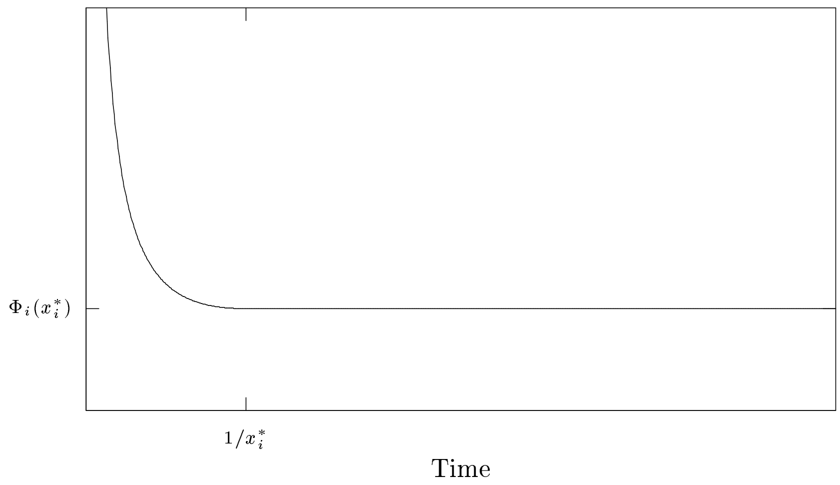

Figure 3. An example of the function $g_i^{(R)}(\cdot)$. Notice the similarity with the graph of $g_i(\cdot)$ in Figure 1.

Now (Q_{rel}) has the following simplified representation:

$$(R) \qquad \inf_{T>0} \left\{ ST + \sum_{i=1}^{n} g_i^{(R)}(T) \right\}.$$

Denote by $v(R)$ the optimal objective value of (R) and by $T(R)$ an optimal T (if it exists). In the remainder we will assume the (R) always has an optimal solution. Notice that by Assumption 2 it follows that $v(R) = v(Q_{rel})$ and $T(R) = T(Q_{rel})$, since (R) and (Q_{rel}) are equivalent. In Figure 4 an example of the objective function of problem (R) is given.

Observe that if each $g_i^{(R)}(\cdot)$ is convex, then (R) is a convex-programming problem. We can prove that $g_i^{(R)}(\cdot)$ is convex on $(0, \infty)$ if the function $M_i(\cdot)$ is convex on (x_i^*, ∞), so that (R) is a convex-programming problem if each function $M_i(\cdot)$ is convex on (x_i^*, ∞). This is established by the following lemma.

Lemma 2 *If the function $M_i(\cdot)$ is convex on (x_i^*, ∞), then the function $g_i^{(R)}(\cdot)$ is convex on $(0, \infty)$.*

PROOF: To show that the function $g_i^{(R)}(t)$ is convex it is sufficient to show that $\Phi_i(1/t)$ is convex on $(0, 1/x_i^*)$. If $\Phi_i(1/t)$ is convex on $(0, 1/x_i^*)$ then it follows from the fact that $\Phi_i(1/(1/x_i^*)) = \Phi_i(x_i^*)$ is

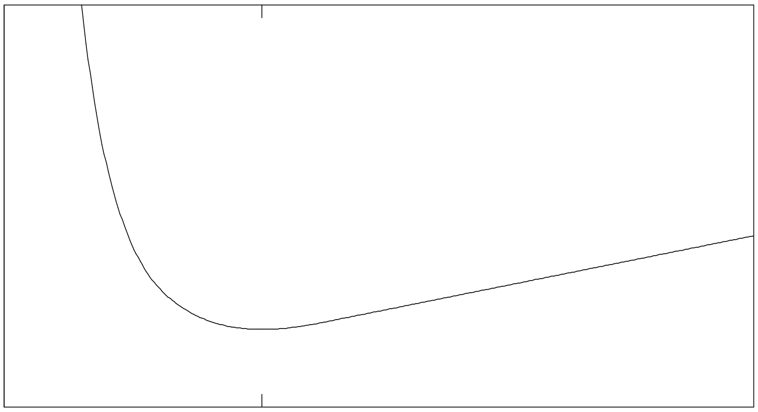

Figure 4. An example of the objective function of problem (R).

the minimal value of $\Phi_i(\cdot)$ on $(0,\infty)$, that $\Phi_i(1/t)$ is also decreasing on $(0, 1/x_i^*)$. Then it follows from the definition of $g_i^{(R)}(t)$ (see (4)) that $g_i^{(R)}(t)$ is convex on $(0,\infty)$.

So we have to prove that $\Phi_i(1/t)$ is convex on $(0, 1/x_i^*)$. We will prove that $\Phi_i(1/t)$ is convex on $(0, 1/b_i)$ if $M_i(t)$ is convex on (b_i,∞). So let $M_i(t)$ be convex on (b_i,∞), then $c_i^p + M_i(t) = t\Phi_i(t)$ is also convex on (b_i,∞). Define for a function $f(\cdot)$

$$s_f(t, t_0) = \frac{f(t) - f(t_0)}{t - t_0},$$

and let $f(t) := t\Phi_i(t)$ and $g(t) := \Phi_i(1/t)$. Then it is easy to verify that

$$s_f(t, t_0) = \Phi_i(t_0) - (1/t_0)s_g(1/t, 1/t_0). \tag{5}$$

Application of the well-known criterion of increasing slopes valid for convex functions (see, e.g., Proposition 1.1.4 in Chapter I of Hiriart-Urruty and Lemaréchal [12]), we have for the convex function $f(t) = t\Phi_i(t)$ on (b_i,∞) that $s_f(t, t_0)$ is increasing in $t > b_i$ for every $t_0 > b_i$. Using (5) this implies that $\Phi_i(t_0) - (1/t_0)s_g(1/t, 1/t_0)$ is increasing in $t > b_i$ for every $t_0 > b_i$. Since $\Phi_i(t_0)$ and $1/t_0$ are constants, the function $-s_g(1/t, 1/t_0)$ is then increasing in $t > b_i$ for every $t_0 > b_i$. Hence, $s_g(1/t, 1/t_0)$ is increasing as a function of $1/t < 1/b_i$ for every $1/t_0 < 1/b_i$, which is equivalent with $s_g(x, x_0)$ is increasing in $x < 1/b_i$ for every $x_0 < 1/b_i$. Using again the criterion of increasing slopes for convex functions it follows that $g(t) = \Phi_i(1/t)$ is convex on $(0, 1/b_i)$.

If $M_i(t)$ is convex on (x_i^*,∞), that is, if $b_i = x_i^*$, then we have that $\Phi_i(1/t)$ is convex on $(0, 1/x_i^*)$, which completes the proof. (Notice that if $M_i(t)$ is convex on $(0,\infty)$, that is, if $b_i = 0$, then we have that $\Phi_i(1/t)$ is also convex on $(0,\infty)$.) □

Theorem 2 *If each (P_i), $i = 1,\ldots,n$, has a finite solution $x_i^* > 0$ and is formulated according to one of the special cases of Goyal et al., the minimal-repair model with a unimodal rate of occurrence of failures or the inspection model, then problem (R) is a convex-programming problem.*

PROOF: In the proof of Theorem 1 we showed that for the minimal-repair model with a unimodal rate of occurrence of failures the function $M_i(\cdot)$ is convex on (x_i^*,∞). In case of an increasing rate of occurrence of failures, $M_i(\cdot)$ is even convex on $(0,\infty)$, and thus a forteriori convex on (x_i^*,∞). We also showed that for the special cases of Goyal et al. and the inspection model the function $M_i(\cdot)$ is convex on $(0,\infty)$, so that $M_i(\cdot)$ is a forteriori convex on (x_i^*,∞). Consequently, if for each $i = 1,\ldots,n$ one of the above models is used (possibly different models for different i), then by Lemma 2 the corresponding $g_i^{(R)}(\cdot)$ are convex so that problem (R) is a convex-programming problem. □

We can now explain why we applied in the previous section the transformation of T into $1/T$ in the original optimisation problem (2). We saw that (R) is a convex-programming problem if each function $g_i^{(R)}(t)$ is convex on $(0,\infty)$. In the proof of Lemma 2 we showed that this is the case if each function $\Phi_i(1/t)$ is convex on $(0,1/x_i^*)$. We showed furthermore that the function $\Phi_i(1/t)$ is convex on $(0,1/x_i^*)$ if $M_i(t)$ is convex on (x_i^*,∞) (which is generally the case for the models described before). If we would not apply the transformation of T into $1/T$ as we did in the previous section, we would have that the corresponding relaxation is a convex-programming problem only if each function $\Phi_i(t)$ is convex on (x_i^*,∞), which is a much more restrictive condition and is in general not the case (not even for the models mentioned before). Summarising, the transformation of T into $1/T$ causes the relaxation to be a convex-programming problem for the models described before, a result that otherwise does not generally hold.

If (R) is a convex-programming problem, it can easily be solved to optimality. When the functions $g_i^{(R)}(\cdot)$ are differentiable (which is the case if the functions $\Phi_i(\cdot)$ are differentiable), we can set the derivative of the cost function in (R) equal to zero and subsequently find an optimal solution with the bisection method. When the functions $g_i^{(R)}(\cdot)$ are not differentiable, we can apply a golden-section search. (For a description of these methods, see Chapter 8 of Bazaraa, Sherali and Shetty [2]).

In all cases it is convenient to have an upper bound on an optimal value $T(R)$. To this end we assume without loss of generality that $1/x_n^* \leq 1/x_{n-1}^* \leq \cdots \leq 1/x_1^*$. The following lemma proves that $T(R) \leq 1/x_1^*$.

Lemma 3 *If we assume without loss of generality that $1/x_n^* \leq 1/x_{n-1}^* \leq \cdots \leq 1/x_1^*$, then for an optimal $T(R)$ of (R) it holds that $T(R) \leq 1/x_1^*$.*

PROOF: For all $T > 1/x_1^*$ it follows that

$$ST + \sum_{i=1}^{n} g_i^{(R)}(T) = ST + \sum_{i=1}^{n} \Phi_i(x_i^*) > \frac{S}{x_1^*} + \sum_{i=1}^{n} \Phi_i(x_i^*) = \frac{S}{x_1^*} + \sum_{i=1}^{n} g_i^{(R)}(1/x_1^*).$$

Hence $T(R) \leq 1/x_1^*$. □

Once (R) is solved we have an optimal $T(R)$. In some cases this value of $T(R)$ is also an optimal T for (Q) as is established by the following lemma.

Lemma 4 *The value $T < 1/x_n^*$ is a local optimal solution of problem (R) if and only if T is a local optimal solution of problem (Q). Moreover, if for an optimal $T(R)$ of problem (R) it holds that $T(R) \leq 1/x_n^*$, then $T(Q) = T(R)$ is optimal for problem (Q).*

PROOF: Since the functions $g_i(\cdot)$ and $g_i^{(R)}(\cdot)$ (see (3) and (4)) are identical on $(0,1/x_n^*]$ (by Assumption 2), $T < 1/x_n^*$ is a local optimal solution of problem (R) if and only if T is a local optimal solution of problem (Q).

To prove the second part, we observe that if $T(R)$ is optimal for problem (R), it holds that

$$ST(R) + \sum_{i=1}^{n} g_i^{(R)}(T(R)) = v(R) \leq v(Q) \leq ST(R) + \sum_{i=1}^{n} g_i(T(R)).$$

Since $T(R) \leq 1/x_n^*$ and hence $g_i(T(R)) = g_i^{(R)}(T(R))$, the upper and lower bound in the above inequality are equal, which implies that $T(R)$ is also optimal for (Q). □

An immediate result of this lemma is the following corollary, that yields a lower bound on $T(Q)$.

Corollary 1 *If there does not exist a local optimal solution of (R) within $(0,1/x_n^*)$, then it holds for the optimal $T(Q)$ of (Q) that $T(Q) \geq 1/x_n^*$.*

Notice that if (R) is a convex-programming problem it is easy to check whether there exists a local optimal solution within $(0, 1/x_n^*)$. When $T(R) \leq 1/x_n^*$ then $T(R)$ is a local (in that case even global) optimal solution and we have that $T(Q) = T(R)$ is optimal for (Q) according to Lemma 4. When $T(R) > 1/x_n^*$, we calculate the objective function of (R) in $1/x_n^*$. When it equals $v(R)$ then $1/x_n^*$ is also optimal for (R), so that $T(Q) = 1/x_n^*$ is also optimal for (Q) again according to Lemma 4. When the objective function of (R) in $1/x_n^*$ is larger than $v(R)$, there does not exist a local optimal solution of (R) within $(0, 1/x_n^*)$ and it holds by Corollary 1 that $T(Q) > 1/x_n^*$. Consequently, we proved the following corollary.

Corollary 2 *Suppose (R) is a convex-programming problem. If the objective function of (R) in $1/x_n^*$ is larger than $v(R)$ and $T(R) > 1/x_n^*$, then $T(Q) > 1/x_n^*$. Otherwise, there is a global optimal solution in $(0, 1/x_n^*]$; an optimal $T(Q)$ is then given by $T(Q) = \min\{1/x_n^*, T(R)\}$.*

Notice that if $T(R) \leq 1/x_n^*$ the corresponding values of k_i are all equal to 1 because of (3). This implies that not only $T(R)$ is optimal for (Q) but the corresponding k_i as well, since the functions $g_i(t)$ and $g_i^{(R)}(t)$ are equal for $t \leq 1/x_n^*$. If $T(R) > 1/x_n^*$, this may not be an optimal T for problem (Q). Besides, the values of k_i corresponding with $T(R)$ are not necessarily integer, which implies that the optimal solution of (R) is in general not feasible for (Q) when $T(R) > 1/x_n^*$. Consequently, the first thing to do when $T(R) > 1/x_n^*$ is to find a feasible solution for (Q).

4.2 Feasibility Procedures for (Q)

A straightforward way to find a feasible solution for (Q) is to substitute the value of $T(R)$ in (3). We saw in Section 3 that an optimal value of k_i depends on the interval $I_i^{(k)}$ that contains $T(R)$. Using this, we can obtain a feasible solution with the following Feasibility Procedure (FP).

Feasibility Procedure

For each $i = 1, \ldots, n$ do the following:

1. Compute $k = \lfloor T(R)x_i^* \rfloor$. This is the value for which $T(R) \in I_i^{(k)}$.
2. If $k = 0$, then $k_i(FP) = 1$ is the optimal k_i-value for (Q) corresponding with $T(R)$ (use (3)).
3. If $k \geq 1$ then $k_i(FP) = k$ or $k_i(FP) = k + 1$ is an optimal value, depending on whether $\Phi_i(k/T(R)) \leq \Phi_i((k+1)/T(R))$ or $\Phi_i(k/T(R)) \geq \Phi_i((k+1)/T(R))$ (use (3)).

Let $v(FP)$ be the objective-function value of (Q) corresponding with this feasible solution. Then we have

$$v(FP) = ST(R) + \sum_{i=1}^{n} g_i(T(R)) \geq v(Q) \geq v(R).$$

Hence we can check the quality of this solution; if $v(FP)$ is close to $v(R)$ then it is also close to the optimal objective value $v(Q)$.

If it is not close enough we may find a better solution by application of a procedure that is similar to the iterative approach of Goyal et al. We call this the Improved-Feasibility Procedure (IFP).

Improved-Feasibility Procedure

1. Let $k_i(IFP) = k_i(FP)$, $i = 1, \ldots, n$, with $k_i(FP)$ the values given by the feasibility procedure FP.
2. Solve the optimisation problem

$$\inf_{T>0} \left\{ ST + \sum_{i=1}^{n} \Phi_i(k_i(IFP)/T) \right\}, \tag{6}$$

and let $T(IFP)$ be an optimal value for T.

3. Determine new constants $k_i(IFP)$ by substitution of $T(IFP)$ in (3). This implies the application of the feasibility procedure FP to the value $T(IFP)$. Let $v(IFP)$ be the corresponding objective-function value.

For the value $v(IFP)$ generated by the IFP it follows that

$$\begin{aligned}
v(IFP) &= ST(IFP) + \sum_{i=1}^{n} g_i(T(IFP)) \\
&= ST(IFP) + \sum_{i=1}^{n} \Phi_i(k_i(IFP)/T(IFP)) \\
&\leq ST(IFP) + \sum_{i=1}^{n} \Phi_i(k_i(FP)/T(IFP)) && \text{(Step 3.)} \\
&\leq ST(R) + \sum_{i=1}^{n} \Phi_i(k_i(FP)/T(R)) && \text{(Step 2.)} \\
&= ST(R) + \sum_{i=1}^{n} g_i(T(R)) \\
&= v(FP).
\end{aligned}$$

Consequently, the solution generated by the IFP is at least as good as the solution obtained with the FP.

The IFP can in principle be repeated with in step 1 the new constants $k_i(IFP)$, and this can be done until no improvement is found. This procedure differs from the iterative algorithm of Goyal et al. in two aspects. The first difference concerns the way integer values of k_i are found given a value of T. We explained in Section 3 that in the algorithm of Goyal and Kusy [10] optimal k_i are found by searching in a table that is made in advance for each i. This becomes inefficient when the values of k_i are large, since then searching in the table takes much time. Besides, this has to be done in each iteration again. Goyal and Gunasekaran [9] find for each i an optimal *real-valued* k_i that is rounded to the nearest integer. This may not be optimal. In our procedure we can identify optimal values of k_i for a given value of T *immediately*, by substitution of T in (3) (that is, by application of the FP).

The second difference concerns the initialisation of the k_i. Goyal et al. initialise each $k_i = 1$ and then find a corresponding optimal T. This often results in a solution that cannot be improved upon by the algorithm but is not optimal, that is, the algorithm is then stuck in a local optimal solution (see Van Egmond, Dekker and Wildeman [19]). In the IFP we start with a value of T that is optimal for (R) and hence might be a good solution for (Q) as well; this may be a much better initialisation for the algorithm (we will investigate this in Section 5).

However, we cannot guarantee that with this alternative initialisation the IFP will not suffer from local optimality. If the procedure terminates and the generated solution $v(IFP)$ is not close to $v(R)$, then we cannot guarantee that it is a good solution. In that case we will apply a global-optimisation algorithm.

Observe that for the models mentioned before (with an increasing rate of occurrence of failures for the minimal-repair model) the IFP is easily solvable since (6) is a convex-programming problem (the functions $\Phi_i(1/t)$ are then convex). Otherwise, the IFP may not be useful since (6) can be a difficult problem to solve. In that case we will not apply the IFP but we will use a global-optimisation algorithm immediately after application of the FP when $v(FP)$ is not close enough to $v(R)$.

To apply a global optimisation we first need an interval that contains an optimal $T(Q)$.

4.3 Upper and Lower Bounds on $T(Q)$

In this subsection we will derive lower and upper bounds on $T(Q)$. Corollary 1 already provides a lower bound $1/x_n^*$ when there is no local optimal solution of (R) within $(0, 1/x_n^*)$. We saw that if (R) is a convex-programming problem this results in Corollary 2 that states that if $T(R) > 1/x_n^*$ and the objective function of (R) in $1/x_n^*$ is larger than $v(R)$ we have $T(Q) > 1/x_n^*$, while otherwise an optimal $T(Q)$ can be identified immediately by $T(Q) = \min\{T(R), 1/x_n^*\}$.

If the functions $M_i(\cdot)$ are convex and differentiable it is easy to see that the functions $g_i^{(R)}$ are differentiable and that (R) is a differentiable convex-programming problem. Moreover, if at least one of the functions $M_i(\cdot)$ is strictly convex we can prove a better lower bound than $1/x_n^*$ on $T(Q)$. This is established by the following lemma.

Lemma 5 *Consider the optimisation problem:*

$$(Q_1) \qquad \inf_{T>0} \left\{ ST + \sum_{i=1}^{n} \Phi_i(1/T) \right\},$$

with $v(Q_1)$ the optimal objective value and $T(Q_1)$ an optimal T. If for each $i = 1, \ldots, n$ the function $M_i(\cdot)$ is convex and differentiable on $(0, \infty)$, and for at least one $i \in \{1, \ldots, n\}$ the function $M_i(\cdot)$ is strictly convex on $(0, \infty)$, and the differentiable convex-programming problem (R) has no global optimal solution within $(0, 1/x_n^*]$, then $T(Q) \geq T(Q_1) > 1/x_n^*$.

PROOF: If there does not exist a global optimal solution of (R) within $(0, 1/x_n^*]$, then it is shown analogously to Lemma 4 and Corollary 1 that $T(Q_1) > 1/x_n^*$.

To prove the inequality $T(Q) \geq T(Q_1)$, notice first that (Q_1) equals the optimisation problem (Q) when all k_i are fixed to the value 1. Consequently, (Q_1) is a more restricted problem than (Q) and it is easy to verify that $v(Q) \leq v(Q_1)$. Furthermore, if $T(Q)$ and certain values of k_i are optimal for (Q), then it is easy to see that if the functions $\Phi_i(\cdot)$ are differentiable the following holds:

$$S - \sum_{i=1}^{n} \frac{d}{dt} \Phi_i(k_i/t) \bigg|_{t=T(Q)} = 0,$$

so that

$$S = \sum_{i=1}^{n} \frac{k_i}{(T(Q))^2} \Phi_i'(k_i/T(Q)).$$

Substitution of this in the optimal objective value of (Q) yields:

$$v(Q) = ST(Q) + \sum_{i=1}^{n} \Phi_i(k_i/T(Q)) = \sum_{i=1}^{n} \left\{ \frac{k_i}{T(Q)} \Phi_i'(k_i/T(Q)) + \Phi_i(k_i/T(Q)) \right\}.$$

It is easily verified that

$$\sum_{i=1}^{n} \left\{ \frac{k_i}{T(Q)} \Phi_i'(k_i/T(Q)) + \Phi_i(k_i/T(Q)) \right\} = \sum_{i=1}^{n} M_i'(k_i/T(Q)),$$

so that

$$v(Q) = \sum_{i=1}^{n} M_i'(k_i/T(Q)). \qquad (7)$$

Analogously, it can be shown for the optimal objective value of (Q_1) that

$$v(Q_1) = \sum_{i=1}^{n} M_i'(1/T(Q_1)). \qquad (8)$$

Suppose now that the inequality $T(Q) \geq T(Q_1)$ does not hold, that is, $T(Q) < T(Q_1)$. Since the functions $M_i(\cdot)$ are (strictly) convex and, consequently, the functions $M_i'(\cdot)$ are (strictly) increasing, this implies that (use (7) and (8))

$$v(Q) = \sum_{i=1}^{n} M_i'(k_i/T(Q)) \geq \sum_{i=1}^{n} M_i'(1/T(Q)) > \sum_{i=1}^{n} M_i'(1/T(Q_1)) = v(Q_1),$$

which is in contradiction with $v(Q) \leq v(Q_1)$. Consequently, $T(Q) \geq T(Q_1)$. □

A rough upper bound on $T(Q)$ is obtained by the following lemma.

Lemma 6 *For an optimal $T(Q)$ of (Q) it holds that*

$$T(Q) \leq (1/S)\left\{v(FP) - \sum_{i=1}^{n} \Phi_i(x_i^*)\right\},$$

with $v(FP)$ the objective value corresponding with the feasible solution of (Q) generated by the FP.

PROOF: For every $T > 0$ if holds that

$$ST + \sum_{i=1}^{n} \Phi_i(x_i^*) \leq ST + \sum_{i=1}^{n} g_i^{(R)}(T) \leq ST + \sum_{i=1}^{n} g_i(T).$$

Consequently, we have for every T with $ST + \sum_{i=1}^{n} g_i(T) \leq v(FP)$ that

$$ST + \sum_{i=1}^{n} \Phi_i(x_i^*) \leq v(FP),$$

which implies that

$$T \leq (1/S)\left\{v(FP) - \sum_{i=1}^{n} \Phi_i(x_i^*)\right\}.$$

Since $T(Q)$ is such a T for which $ST + \sum_{i=1}^{n} g_i(T) \leq v(FP)$, the lemma follows. □

We can obtain a better upper bound if (R) is a convex-programming problem. This is established by the following lemma.

Lemma 7 *Let T_{up} be the smallest $T \geq T(R)$ for which the objective function of (R) equals $v(FP)$. If (R) is a convex-programming problem then T_{up} is an upper bound on $T(Q)$. Moreover, this upper bound is smaller than or equal to the upper bound according to Lemma 6.*

PROOF: For all $T > 1/x_1^*$ it follows that

$$ST + \sum_{i=1}^{n} g_i^{(R)}(T) = ST + \sum_{i=1}^{n} \Phi_i(x_i^*),$$

where the function in the right-hand side of the equation is an increasing function in T that tends to infinity if T tends to infinity. Consequently, there exists a value $T_{up} \geq T(R)$ such that $ST + \sum_{i=1}^{n} g_i^{(R)}(T) = v(FP)$. For values of $T > T_{up}$ the objective function of (R) is larger than or equal to $v(FP)$, since (R) is a convex-programming problem and the minimum is obtained in $T(R)$. Since (R) is a relaxation of (Q), the objective function of (Q) is also larger than or equal to $v(FP)$ for values of $T > T_{up}$, so that T_{up} is a lower bound on $T(Q)$.

It is easy to see that this upper bound is at least as good as the upper bound of Lemma 6, since for $T = (1/S)\{v(FP) - \sum_{i=1}^{n} \Phi_i(x_i^*)\}$ it holds that

$$\begin{aligned} ST + \sum_{i=1}^{n} g_i^{(R)}(T) &\geq ST + \sum_{i=1}^{n} \Phi_i(x_i^*) \\ &= S\left[(1/S)\left\{v(FP) - \sum_{i=1}^{n} \Phi_i(x_i^*)\right\}\right] + \sum_{i=1}^{n} \Phi_i(x_i^*) \\ &= v(FP) \\ &= ST_{up} + \sum_{i=1}^{n} g_i^{(R)}(T_{up}). \end{aligned}$$

That is, in $T = (1/S)\{v(FP) - \sum_{i=1}^{n} \Phi_i(x_i^*)\}$ the objective function of (R) is not smaller than in T_{up}. Since (R) is a convex-programming problem and T_{up} is the smallest $T \geq T(R)$ for which the objective function of (R) equals $v(FP)$, we have that $T \geq T_{up}$. □

Notice that the upper bound T_{up} can easily be found with a bisection on the interval $[T(R), (1/S)\{v(FP) - \sum_{i=1}^{n} \Phi_i(x_i^*)\}]$.

It cannot generally be proved that the objective function of (R) is equal to $v(FP)$ for a value of $T \leq T(R)$, but if it is, we have a lower bound T_{low} on $T(Q)$ analogously.

Lemma 8 *If there is a $T \leq T(R)$ for which the objective function of (R) is equal to $v(FP)$, let then T_{low} be the largest $T \leq T(R)$ for which this holds. If (R) is a convex-programming problem then T_{low} is a lower bound on $T(Q)$.*

PROOF: For values of $T < T_{low}$ the objective function of (R) is larger than or equal to $v(FP)$, since (R) is a convex-programming problem and the minimum is obtained in $T(R)$. Since (R) is a relaxation of (Q), the objective function of (Q) is also larger than or equal to $v(FP)$ for values of $T < T_{low}$, so that T_{low} is a lower bound on $T(Q)$. □

In Figure 5 it is illustrated how the bounds T_{low} and T_{up} are generated. If (R) is a convex-programming problem and the lower bound T_{low} exists, then it can easily be found as follows. We first check whether $T_{low} \geq 1/x_n^*$, with $1/x_n^*$ the lower bound given by Corollary 2. To this end we compute the objective function of (R) in $1/x_n^*$ and check whether it is smaller than $v(FP)$. If so, then $T_{low} < 1/x_n^*$ and otherwise $T_{low} \geq 1/x_n^*$. In the latter case we can easily find T_{low} with a bisection on the interval $[1/x_n^*, T(R)]$.

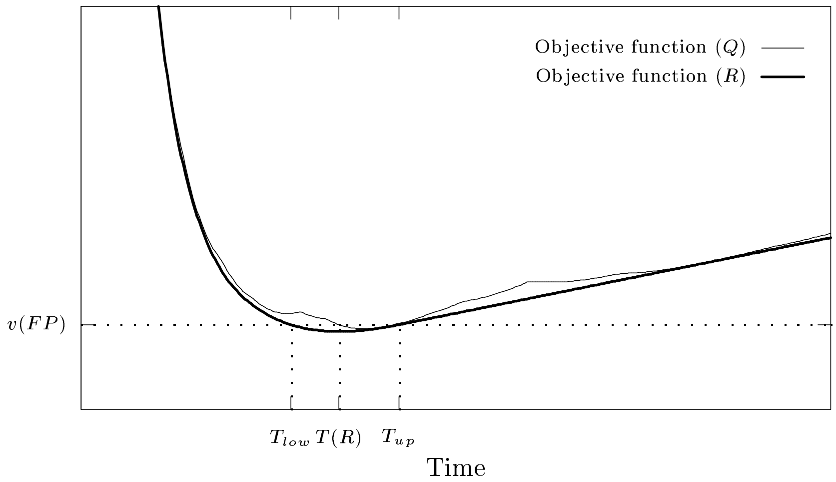

Figure 5. A lower bound T_{low} and an upper bound T_{up} on an optimal $T(Q)$ are found where the objective function of relaxation (R) equals $v(FP)$, the value of the objective function of problem (Q) in $T(R)$.

Notice that if (R) is a convex-programming problem, it can be useful to apply the IFP. In that case the bounds T_{up} and T_{low} derived above may be improved when the objective value $v(FP)$ is replaced by $v(IFP)$, since $v(IFP) \leq v(FP)$.

In this subsection we derived a number of upper and lower bounds on $T(Q)$. The results are summarised in Table I.

From Table I we can find the bounds that can be used dependent on certain conditions. For example, for the special cases of Goyal et al., the minimal-repair model with a unimodal rate of occurrence of failures and the inspection model, we showed in Theorem 2 that (R) is a convex programming problem. This is already sufficient to use all bounds of Table I, except the lower bound $T(Q_1)$. To use the bound $T(Q_1)$, each $M_i(\cdot)$ must be convex on $(0, \infty)$ and at least one $M_i(\cdot)$ must be strictly convex. We

Table I
Lower and Upper Bounds on $T(Q)$

Lower bound	Condition
$1/x_n^*$	(R) has no local optimal solution in $(0, 1/x_n^*]$
$T(Q_1)$	Each $M_i(\cdot)$ is convex on $(0, \infty)$ and at least one $M_i(\cdot)$ is strictly convex on $(0, \infty)$
T_{low}	(R) is a convex-programming problem and its objective function equals $v(FP)$ for a $T \leq T(R)$
Upper bound	**Condition**
$\{v(FP) - \sum_{i=1}^{n} \Phi_i(x_i^*)\}/S$	none
T_{up}	(R) is a convex-programming problem

showed in the proof of Theorem 1 that each $M_i(\cdot)$ is convex on $(0, \infty)$ for the models described above (with an increasing rate of occurrence of failures for the minimal-repair model). For the special cases of Goyal et al. each $M_i(\cdot)$ is even strictly convex on $(0, \infty)$, so that the bound $T(Q_1)$ can then always be used. For the minimal-repair and inspection model at least one $M_i(\cdot)$ must be strictly convex.

Let now T_l be the largest lower bound and T_u be the smallest upper bound that can be used for a specific problem, then we have that $T(Q) \in [T_l, T_u]$. Consequently, it is sufficient to apply a global-optimisation technique on the interval $[T_l, T_u]$ to find a value for $T(Q)$.

4.4 Global-Optimisation Techniques

What remains to be specified is the usage of a global-optimisation technique for (Q) on the interval $[T_l, T_u]$ when the feasible solution to (Q) found after application of the FP (or the IFP) is not good enough.

Lipschitz Optimisation

Efficient global-optimisation techniques exist for the case that the objective function of (Q) is *Lipschitz*. A univariate function is said to be Lipschitz if for each pair x and y the absolute difference of the function values in these points is smaller than or equal to a constant (called the *Lipschitz constant*) multiplied by the absolute distance between x and y. More formally:

Definition 2 *A function $f(x)$ is said to be Lipschitz on the interval $[a, b]$ with Lipschitz constant L, if for all $x, y \in [a, b]$ it holds that $|f(x) - f(y)| \leq L|x - y|$.*

If the objective function of (Q) is Lipschitz on the interval $[T_l, T_u]$, then global-optimisation techniques can be applied in this interval to obtain a solution with a corresponding objective value that is arbitrarily close to the optimal objective value $v(Q)$ (see the chapter on Lipschitz optimisation in Horst and Pardalos [13]). For the special cases of Goyal et al., the minimal-repair model with an increasing rate of occurrence of failures, and the inspection model, we can prove that the objective function of (Q) is Lipschitz on $[T_l, T_u]$, and we can derive a Lipschitz constant (see appendix).

There are several Lipschitz-optimisation algorithms (see Horst and Pardalos [13]), and we implemented some of them. The simplest one, called the *passive algorithm*, evaluates the function to be minimised at the points $a + \epsilon/L$, $a + 3\epsilon/L$, $a + 5\epsilon/L, \ldots$, and takes the point at which the function is minimal. The function value in this point does not differ more than ϵ from the minimal value in $[a, b]$. We implemented the algorithm of Evtushenko that is based on the passive algorithm, but that takes a following step larger than $2\epsilon/L$ if the current function value is larger than 2ϵ above the current *best* known value, which makes the algorithm faster than the passive algorithm. However, this algorithm can still be very time consuming, especially when the Lipschitz constant L is large. The algorithm of Evtushenko and the other algorithms described in Horst and Pardalos [13] turned out to be too time consuming, and were therefore not of practical use to our problem.

Fortunately, however, the shape of the objective function of problem (Q) is such that the Lipschitz constant is decreasing in T (this is shown in the appendix). Using this, the algorithm of Evtushenko can easily be extended to deal with a *dynamic* Lipschitz constant; after a certain number of steps (going from left to right) the Lipschitz constant is recomputed, such that larger steps can be taken. This is repeated after the same number of steps, and so on, until the interval $[a, b]$ is covered. This approach turned out to work very well for our problem; the increment in speed was sometimes a factor 1000 compared to the version of Evtushenko, and this made Lipschitz optimisation of practical use to our problem.

Golden-Section Search Heuristic

We can also apply alternative methods that do not use the notion of Lipschitz optimisation. One such a method is golden-section search. Golden-section search is usually applied (and is optimal) for functions that are strictly unimodal, which the objective function of (Q) is generally not. However, we will apply an approach in which the interval $[T_l, T_u]$ is divided into a number of subintervals of equal length, on each of which a golden-section search is applied. The best point of these intervals is taken as solution. We then divide the subintervals into intervals that are twice as small and apply on each a golden-section search again. This doubling of intervals is repeated until no improvement is found. We refer to this approach as the *multiple-interval golden-section search heuristic*, the results of which are given in Section 5.

4.5 A Solution Procedure for (Q)

We are now ready to formulate a solution procedure for (Q). We consider first a solution procedure for the special cases of Goyal et al., the minimal-repair model with a unimodal rate of occurrence of failures, and the inspection model, in which cases problem (R) is a convex-programming problem. Subsequently, we indicate the changes when, for example, block replacement is used.

We can summarise the results in this section in the formulation of the following solution procedure for (Q):

1. Solve the convex-programming problem (R) using that $T(R) \leq 1/x_1^*$. An optimal value $T(R)$ can be found by application of a bisection technique if the objective function of (R) is differentiable, and otherwise golden-section search can be applied.

2. If $T(R) \leq 1/x_n^*$ then $T(Q) = T(R)$ is optimal for (Q); stop.

3. If $T(R) > 1/x_n^*$, check whether the objective function of (R) in $1/x_n^*$ equals $v(R)$. If so, $T(Q) = 1/x_n^*$ is optimal for (Q); stop.

4. Otherwise, we have that $T(Q) > 1/x_n^*$ and we first find a feasible solution for problem (Q) by applying the FP or IFP. If the corresponding objective value is close enough to $v(R)$, then it is also close to $v(Q)$, so that we have a good solution; stop.

5. If the solution is not good enough, apply a global-optimisation technique on the interval $[T_l, T_u]$ to find a value for $T(Q)$.

If this solution procedure is applied to the block-replacement model, some details have to be modified slightly. The first modification concerns the solution of the relaxation (R) that is in general not a convex-programming problem, but, since it has fewer local minima, is much easier to solve than problem (Q). Therefore, to find a solution to problem (R), we will apply a single iteration of the multiple-interval golden-section search heuristic described in the previous subsection, that is, the number of subintervals is fixed and will not be doubled until no improvement is found.

Though the optimisation problem (6) is not a convex-programming problem for the block-replacement model and is therefore more difficult to solve, we will still use the IFP with a single golden-section search applied to solve problem (6); even as such the IFP outperforms the approach of Goyal and Kusy, as will be shown by the experiments in the next section.

Since the nice results that we derived for the special cases of Goyal et al., minimal repair and inspection do not generally hold for the block-replacement model, determination of a Lipschitz constant becomes more difficult, if possible at all. Therefore, we will not apply Lipschitz optimisation to determine a value of $v(Q)$. Instead, we will use the multiple-interval golden-section search heuristic as described in the previous subsection.

5 Numerical Results

In this section the solution procedure described in the previous section will be investigated and it will be compared with the iterative approach of Goyal et al. This will first be done for the special case of Goyal and Kusy, the minimal-repair model with an increasing rate of occurrence of failures, and the inspection model, in which cases an optimal solution can be found by Lipschitz optimisation. This makes it possible to make a good comparison and also to investigate the performance of the multiple-interval golden-section search heuristic. Subsequently, the performance of the solution procedure for the block-replacement model is investigated, using the golden-section search heuristic. Since the iterative algorithm of Goyal and Kusy [10] performs better than that of Goyal and Gunasekaran [9], we chose the first one to compare it with our solution procedure. All algorithms are implemented in Borland Pascal version 7.0 on a 66 MHz personal computer.

For all models we have 6 different values for the number n of components and 7 different values for the set-up cost S. This yields 42 different combinations of n and S, and for each of these combinations 100 random problem instances are taken by choosing random values for the remaining parameters. For the minimal-repair, inspection and block-replacement model the lifetime distribution for component i is given by a Weibull-(λ_i, β_i) distribution (a Weibull-(λ, β) distributed stochastic variable has a cumulative distribution function $F(t) = 1 - e^{-(t/\lambda)^\beta}$). The data are summarised in Table II.

Table II
Values for the Parameters in the Four Models

$n = 3, 5, 7, 10, 25, 50$	
$S = 10, 50, 100, 200, 500, 750, 1000$	
$c_i^p \in [1, 500]$ (random)	
Special Case of Goyal and Kusy:	*Minimal-Repair Model:*
$f_i \in [15, 50]$ (random)	$\lambda_i \in [1, 20]$ (random)
$v_i \in [1, 20]$ (random)	$\beta_i \in [1.5, 4]$ (random)
$e \in [1, 4]$ (random)	$c_i^r \in [1, 250]$ (random)
Inspection Model:	*Block-Replacement Model:*
$\lambda_i \in [1, 20]$ (random)	$\lambda_i \in [1, 20]$ (random)
$\beta_i \in [1.5, 4]$ (random)	$\beta_i \in [1.5, 4]$ (random)
$c_i^u \in [c_i^p/\mu_i + 1, 1000]$ (random)	$c_i^f \in [2c_i^p/(1 - \sigma_i^2/\mu_i^2) + 1, 5000]$ (random)

The variables μ_i and σ_i in this table (for the block-replacement and inspection model) are the expectation and the standard deviation of the the lifetime distribution of component i. Notice that for the inspection model we take $c_i^u \geq c_i^p/\mu_i + 1$ and for the block-replacement model $c_i^f \geq 2c_i^p/(1 - \sigma_i^2/\mu_i^2) + 1$. This guarantees the existence of a finite minimum x_i^* for the individual average-cost function $\Phi_i(\cdot)$. In Dekker [4] it is shown that for the inspection model a finite minimum for $\Phi_i(\cdot)$ exists if $c_i^p < c_i^u \mu_i$, and, a forteriori, if $c_i^u \geq c_i^p/\mu_i + 1$. For the block-replacement model it can be shown (see also Dekker [4]) that a finite minimum exists if $c_i^f > 2c_i^p/(1 - \sigma_i^2/\mu_i^2)$. Notice finally that since $\beta_i \geq 1$, the rate of occurrence of failures for the minimal repair model is increasing.

Results for the special case of Goyal and Kusy, the minimal-repair model and the inspection model

For the special case of Goyal and Kusy, the minimal-repair model and the inspection model, the value $v(Q)$ can be determined by Lipschitz optimisation with an arbitrary deviation from the optimal value; we allowed a relative deviation of 10^{-4} (i.e., 0.01%). In Table III the relevant results of the 4200 problem instances for each model are given.

Table III
Results of 4200 Random Examples for the Special Case of Goyal and Kusy, the Minimal-Repair Model and the Inspection Model

	GoyKus	MinRep	Inspec
Average running time relaxation (sec.)	0.01	0.01	0.06
Improved Feasibility Procedure (IFP):			
Average running time IFP (sec.)	0.07	0.05	1.32
Average deviation IFP (v(IFP)-v(Q))/v(Q)	0.443%	0.065%	0.129%
Minimum deviation IFP	0.000%	0.000%	0.000%
Maximum deviation IFP	10.842%	4.250%	7.184%
Golden-Section Search (GSS):			
Average running time GSS (sec.)	0.72	0.41	11.81
Average deviation GSS (v(GSS)-v(Q))/v(Q)	0.001%	0.000%	0.000%
Minimum deviation GSS	0.000%	0.000%	-0.001%
Maximum deviation GSS	0.334%	0.152%	0.107%
Iterative Algorithm Goyal and Kusy (GK):			
Average running time GK (sec.)	0.07	0.12	4.64
Average deviation GK (v(GK)-v(Q))/v(Q)	0.829%	0.421%	1.253%
Minimum deviation GK	0.000%	-0.001%	0.000%
Maximum deviation GK	11.654%	18.289%	66.188%

From the table it can be seen that solving the relaxation takes very little time. Applying the IFP also takes little time. (In all cases the running times for the inspection model are higher than for the special case of Goyal and Kusy and the minimal-repair model, since for the inspection model a numerical routine has to be applied for each function evaluation, whereas for the other two models the cost functions can be computed analytically.) Notice that some deviations are negative. This is due to the relative deviation of 0.01% allowed in the optimal objective value determined by the Lipschitz optimisation; a heuristic can give a solution with objective value up to 0.01% smaller than that according to the Lipschitz-optimisation procedure.

In all cases the IFP (that is an intelligent modification of the approach of Goyal et al.) outperforms the iterative algorithm of Goyal and Kusy, while the running times of the IFP are equal or faster. The differences are smallest for the special case of Goyal and Kusy. This can be explained from the fact that in the model of Goyal and Kusy there is little variance possible in the lifetime distributions of the components, mainly because the exponent e has to be the same for all components. In the inspection model, however, there can be large differences in the individual lifetime distributions, and this can cause much larger deviations for the iterative algorithm of Goyal and Kusy; the average deviation for Goyal and Kusy's algorithm is then 1.253% and the maximum deviation even 66.188%, which is much higher than the deviations for the IFP. The IFP performs well for all models.

Since for many examples the algorithm of Goyal and Kusy and the IFP find the optimal solution, the average deviations of these algorithms do not differ so much (in many cases the deviation is zero percent). However, there is a considerable difference in the number of times that large deviations were generated. This is illustrated in Table IV that gives the percentage of the examples in which the deviation was larger than 1% and 5% for the three models discussed in this subsection. From this table it is clear that the IFP performs much better than the algorithm of Goyal and Kusy and that if the algorithm of Goyal and Kusy does not give the optimal solution, the deviation can be large. The conclusion is that solving the relaxation and subsequently the improved feasibility procedure is better than and at least as fast as the iterative algorithm of Goyal and Kusy. This also implies that the algorithm of Goyal and Kusy can be improved considerably if another initialisation of the k_i and T is taken, viz. according to the solution of the relaxation.

The deviation of 66.188% in Table III occurs for one of the problem instances of the inspection model with $n = 5$ and $S = 10$. The parameters and results are given in Table V. The large deviation for

Table IV
Percentage of the Examples Where the IFP and the Algorithm of Goyal and Kusy Generated Deviations of More Than 1% and 5%.

Algorithm	Deviation > 1%	Deviation > 5%
	Special Case of Goyal and Kusy	
IFP	12.86	1.79
Goyal and Kusy	27.50	2.10
	Minimal-Repair Model	
IFP	1.57	—
Goyal and Kusy	12.38	1.64
	Inspection Model	
IFP	3.12	0.05
Goyal and Kusy	26.50	6.69

Table V
Parameters and Results for the Problem Instance of the Inspection Model for Which the Algorithm of Goyal and Kusy Performs Worst

	$S = 10$				
Component	c_i^p	c_i^u	λ_i	β_i	x_i^*
1	247.00	962.00	1	3.50	0.83
2	472.00	475.00	9	3.45	5.99
3	344.00	511.00	20	1.71	7.04
4	459.00	528.00	14	3.90	8.45
5	225.00	541.00	17	2.47	6.45

Optimal solution:
$T = 0.85$
$k_i = 1, 7, 8, 10, 8$
corresponding objective value $v(Q) = 706.29$

Solution of Goyal and Kusy's algorithm:
$T = 5.87$
$k_i = 1, 1, 1, 1, 1$
corresponding objective value $v(GK) = 1173.77$
$100\% \times (v(GK) - v(Q))/v(Q) = 66.188\%$

the algorithm of Goyal and Kusy can be explained as follows. In the first iteration of the algorithm all k_i are initialised at the value 1. The corresponding T is then determined; it equals 5.87. In the following iteration it is investigated for each component i whether a larger integer value for k_i given $T = 5.87$ yields lower individual average costs. This is not the case, as can also be expected considering the individual x_i^* in the last column of Table V. Take, for example, $k_2 = 2$ for component 2. This implies that component 2 is inspected each $2 \times 5.87 = 11.74$ time units, whereas its optimal inspection interval has length $x_2^* = 5.99$. The value 5.87 turns out to be a better alternative than 11.74, which also turns out to be the case for the other components. Consequently, the algorithm terminates with $T = 5.87$ and all k_i equal to 1. For component 1 this implies that it is inspected each 5.87 time units whereas the optimal inspection interval has length 0.83. Since for component 1 the failure cost c_1^u per unit time is relatively large, this implies a large deviation; the individual average-cost function of component 1 is relatively steep. It would be much better to take a smaller T and to increase the k_i for components 2,3,4,5 accordingly, which is indeed reflected by the optimal T that equals 0.85.

From the results of Table III it can further be seen that the multiple-interval golden-section search heuristic performs very well in all cases. The average deviation is almost zero, and the maximum deviation is relatively small. The heuristic is initialised with four subintervals and this number is doubled until no improvement is found. It turned out that four subintervals is mostly sufficient. The running time of the heuristic is also quite moderate: less than a second for the special case of Goyal and Kusy and the minimal-repair model, and almost 12 seconds for the inspection model (where a numerical routine has to be applied for each function evaluation).

Usually, Lipschitz optimisation can take much time. For the special cases in this subsection, Lipschitz optimisation can be made much faster by application of a dynamic Lipschitz constant, as was explained in the previous subsection. For the special case of Goyal and Kusy, Lipschitz optimisation then took on average 5.83 seconds, for the minimal-repair model only 0.82 seconds, and for the inspection model 23.82 seconds. This is more than the golden-section search heuristic, but still not very much. The running time depends on the number of components and on the set-up cost. In Table VI the average running times are given for the hundred random examples that were taken for each of the 42 combinations of n and S for the minimal-repair model. As can be seen from this table, the running time increases

Table VI
Average Running Times (sec.) of the Lipschitz-Optimisation Procedure
for the Minimal-Repair Model

	$n = 3$	$n = 5$	$n = 7$	$n = 10$	$n = 25$	$n = 50$
$S = 10$	0.29	0.36	0.53	0.82	2.01	4.95
$S = 50$	0.13	0.22	0.36	0.58	1.28	3.35
$S = 100$	0.10	0.20	0.30	0.47	1.25	2.55
$S = 200$	0.07	0.17	0.23	0.38	0.97	2.16
$S = 500$	0.07	0.13	0.21	0.31	0.96	1.91
$S = 750$	0.05	0.13	0.19	0.31	0.91	2.14
$S = 1000$	0.05	0.10	0.18	0.29	0.90	1.96
average	0.11	0.19	0.29	0.45	1.18	2.71

approximately linearly in the number n of components and decreases in the set-up cost S. The fact that the running time decreases if S increases is due to a steeper objective function for larger S. A larger S causes smaller upper bounds for $T(Q)$ and, as a result, smaller intervals on which Lipschitz optimisation has to be applied. The running time also depends on the precision that is required. For less precision Lipschitz optimisation becomes much faster. Future generations of computers will make the advantage of the golden-section search heuristic over Lipschitz optimisation less important.

We can conclude that if a solution is required in little time, we can solve the relaxation and apply the improved feasibility procedure to obtain a solution for the above problem with a deviation of less than one percent on average. The improved feasibility procedure outperforms the algorithm of Goyal and Kusy not only by time and average deviation, but the maximum deviation is also much smaller.

When precision is more important, we can apply the golden-section search heuristic to obtain a solution for the above problems with a deviation of almost zero percent on average. When optimality must be guaranteed or when running time is less important, Lipschitz optimisation can be applied to obtain a solution with arbitrary precision.

Results for the block-replacement model

For the solution of (R) we applied one iteration of the multiple-interval golden-section search heuristic, that is, we do not double the number of subintervals until no improvement is found. Since in the previous subsection it turned out that four subintervals is mostly sufficient to find a solution for problem (Q), we chose the number four here as well.

In Table VII the relevant results of the 4200 problem instances are given (for the renewal function in the block-replacement model we used the approximation of Smeitink and Dekker [16]). The solutions of the IFP and of the algorithm of Goyal and Kusy are now compared with the values of $v(Q)$ obtained by the multiple-interval golden-section search heuristic.

Table VII

Results of 4200 Random Examples for the Block-Replacement Model

Average running time relaxation (sec.)	0.23
Improved Feasibility Procedure (IFP):	
Average running time IFP (sec.)	1.30
Average deviation IFP (v(IFP)-v(Q))/v(Q)	0.051%
Minimum deviation IFP	-0.002%
Maximum deviation IFP	5.921%
Golden-Section Search (GSS):	
Average running time GSS (sec.)	10.26
Iterative Algorithm Goyal and Kusy (GK):	
Average running time GK (sec.)	3.72
Average deviation GK (v(GK)-v(Q))/v(Q)	0.658%
Minimum deviation GK	-0.222%
Maximum deviation GK	39.680%

The average running time of the relaxation is again very small. It is larger than the average running time of, for example, the inspection model, since golden-section search is not applied once but four times, according to one iteration of the multiple-interval golden-section search heuristic.

From the results it is evident that the IFP outperforms the iterative algorithm of Goyal and Kusy. The average deviation is 0.658% for the algorithm of Goyal and Kusy and only 0.051% for the IFP. Besides, the maximum deviation for the IFP is quite moderate, 5.921%, whereas for the algorithm of Goyal and Kusy this can be as large as 39.680%. It can happen that the algorithm of Goyal and Kusy sometimes performs slightly better than the IFP, reflected in the minimum deviations of -0.222% and -0.002%, respectively.

The golden-section search heuristic applied to solve problem (Q) needed again four intervals in most cases. The average running time of the heuristic is 10.26 seconds. Remember that the solutions of the algorithm of Goyal and Kusy and of the IFP are compared with the solutions according to the golden-section search heuristic. Notice that the negative deviations of -0.222% and -0.002% imply that both the algorithm of Goyal and Kusy and the IFP can in some cases be better than the golden-section search heuristic, though the differences are small. This implies that the golden-section search heuristic is not optimal, but that was already clear from the results in the previous subsection. However, in most cases the heuristic is better than the other two approaches, regarding the average deviations of 0.658% and 0.051% for the algorithm of Goyal and Kusy and the IFP compared to the heuristic.

The conclusion here is again that when a solution is required in little time, we can solve the relaxation and apply the improved feasibility procedure; this is better than the algorithm of Goyal and Kusy (especially the maximum deviation is much smaller). When precision is more important, we can apply the golden-section search heuristic, at the cost of more time.

6 Conclusions

In this paper we presented a general approach for the coordination of maintenance frequencies. We extended an approach by Goyal et al. that deals with components with a very specific deterioration structure and that does not indicate how good the obtained solutions are. Extension of this approach enabled incorporation of well-known maintenance models like minimal repair, inspection and block replacement. We presented an alternative solution approach that can solve these models to optimality (except the block-replacement model, for which our approach is used as a heuristic).

The solution of a relaxed problem followed by the application of a feasibility procedure yields a solution in little time and less than one percent above the minimal value. This approach outperforms the approach of Goyal et al. When precision is more important, a heuristic based on golden-section search can be applied to obtain a solution with a deviation of almost zero percent. For the special cases of Goyal et al., the minimal-repair model and the inspection model, application of a procedure using a dynamic Lipschitz constant yields a solution with an arbitrarily small deviation from an optimal solution, with running times somewhat larger than those of the golden-section search heuristic.

In the solution approach of this paper many maintenance-optimisation models can be incorporated. Not only the minimal-repair, inspection and block-replacement models, but many others can be handled as well. It is also easily possible to combine *different* maintenance activities, for example to combine the inspection of a component with the replacement of another. Altogether, the approach presented here is a flexible and powerful tool for the coordination of maintenance frequencies for multiple components.

Appendix: Determination of Lipschitz Constant

We will prove here that the objective function of problem (Q) is Lipschitz on the interval $[T_l, T_u]$ for the special cases of Goyal et al., the minimal-repair model with an increasing rate of occurrence of failures, and the inspection model. Furthermore, we derive a Lipschitz constant L.

It is obvious that if L_i is a Lipschitz constant for the function $g_i(\cdot)$ (see (3)), then the Lipschitz constant L for the objective function of (Q) equals

$$L = S + \sum_{i=1}^{n} L_i, \tag{9}$$

with S the set-up cost. Consequently, we have to find an expression for L_i.

To do so, consider an arbitrary $i \in \{1, \ldots, n\}$ and determine which of the intervals $I_i^{(k)}$ (see Section 3) overlap with the interval $[T_l, T_u]$. Clearly, this is for each k with $\lfloor T_l x_i^* \rfloor \leq k \leq \lfloor T_u x_i^* \rfloor$. Now define $L_i^{(k)}$ as the Lipschitz constant of $g_i(\cdot)$ on $I_i^{(k)}$ for each of these $k \geq 1$. If $\lfloor T_l x_i^* \rfloor = 0$, then let $L_i^{(0)}$ be the Lipschitz constant of $g_i(\cdot)$ on $[1/x_n^*, 1/x_i^*]$. We will show that

$$L_i = \max\{L_i^{(k)}\}, \quad \text{where } k \text{ ranges from } \max\{0, \lfloor T_l x_i^* \rfloor\} \text{ to } \lfloor T_u x_i^* \rfloor. \tag{10}$$

To prove this, observe first that if t_1, t_2 belong to the same interval $I_i^{(k)}$, then by definition

$$|g_i(t_1) - g_i(t_2)| \leq L_i^{(k)} |t_1 - t_2| \leq L_i |t_1 - t_2|.$$

If t_1, t_2 do not belong to the same interval then assume without loss of generality that $g_i(t_1) \geq g_i(t_2)$. For $t_1 < t_2$ with t_1 belonging to $I_i^{(k)}$ it then follows that

$$\begin{aligned} 0 \leq g_i(t_1) - g_i(t_2) &\leq g_i(t_1) - \Phi_i(x_i^*) \\ &= g_i(t_1) - g_i((k+1)/x_i^*) \\ &\leq L_i^{(k)}((k+1)/x_i^* - t_1) \end{aligned}$$

$$
\begin{aligned}
&\leq L_i^{(k)}(t_2 - t_1) \\
&\leq L_i|t_1 - t_2|.
\end{aligned}
$$

The other case $t_2 < t_1$ can be derived in a similar way and so we have shown that

$$|g_i(t_1) - g_i(t_2)| \leq L_i|t_1 - t_2|,$$

with L_i according to (10).

If we now find an expression for the Lipschitz constant $L_i^{(k)}$, then with (9) and (10) we have an expression for the Lipschitz constant L. In the proof of Lemma 2 we showed that if $M_i(t)$ is convex on $(0,\infty)$, then $\Phi_i(1/t)$ is also convex on $(0,\infty)$. We saw in the proof of Theorem 1 that $M_i(t)$ is convex on $(0,\infty)$ for the special cases of Goyal et al., the minimal-repair model with an increasing rate of occurrence of failures, and the inspection model. Consequently, for these models $\Phi_i(1/t)$ is convex on $(0,\infty)$. This implies that the derivative of the function $\Phi_i(1/t)$ is increasing, and consequently we obtain that for all $t_1 \leq t_2 \in [1/x_n^*, 1/x_i^*]$:

$$
\begin{aligned}
|g_i(t_1) - g_i(t_2)| &= |\Phi_i(1/t_1) - \Phi_i(1/t_2)| \\
&\leq -\frac{d}{dt}\Phi_i(1/t)\bigg|_{t=t_1} \cdot |t_1 - t_2| \\
&\leq -\frac{d}{dt}\Phi_i(1/t)\bigg|_{t=1/x_n^*} \cdot |t_1 - t_2| \\
&= (x_n^*)^2\Phi_i'(x_n^*)|t_1 - t_2|,
\end{aligned}
$$

so that

$$L_i^{(0)} = (x_n^*)^2\Phi_i'(x_n^*). \tag{11}$$

By the same argument we find that for $k \geq 1$

$$L_i^{(k)} = \max\left\{-\frac{d}{dt}\Phi_i((k+1)/t)\bigg|_{t=k/x_i^*}, \frac{d}{dt}\Phi_i(k/t)\bigg|_{t=(k+1)/x_i^*}\right\},$$

and so

$$L_i^{(k)} = \max\left\{\frac{k+1}{k^2}(x_i^*)^2\Phi_i'\left(\frac{k+1}{k}x_i^*\right), -\frac{k}{(k+1)^2}(x_i^*)^2\Phi_i'\left(\frac{k}{k+1}x_i^*\right)\right\}. \tag{12}$$

Notice that both arguments in (12) are decreasing in k since $\Phi_i'(\cdot)$ is increasing. This implies that $L_i^{(k)}$ is maximal for $k = 1$ or $k = 0$. Consequently, (10) becomes

$$L_i = \begin{cases} L_i^{(\lfloor T_l x_i^* \rfloor)} & \text{if } \lfloor T_l x_i^* \rfloor \geq 1, \\ \max\{L_i^{(1)}, L_i^{(0)}\} & \text{if } \lfloor T_l x_i^* \rfloor = 0, \end{cases}$$

with $L_i^{(k)}$ given by (11) and (12).

This analysis also shows that the Lipschitz constant L is decreasing in T. That is, if L_1, L_2 are the Lipschitz constants for the objective function of (Q) on the intervals $[T_1, T_u]$ and $[T_2, T_u]$ respectively, with $T_1 \leq T_2 \leq T_u$, then $L_1 \geq L_2$.

References

[1] Bäckert, W. and D.W.T. Rippin. The determination of maintenance strategies for plants subject to breakdown. *Computers and Chemical Engineering*, **9**:113–126, 1985.

[2] Bazaraa, M.S., H.D. Sherali, and C.M. Shetty. *Nonlinear Programming Theory and Algorithms.* Wiley, New York, 1993.

[3] Cho, D.I. and M. Parlar. A survey of maintenance models for multi-unit systems. *European Journal of Operational Research*, **51**:1–23, 1991.

[4] Dekker, R. Integrating optimisation, priority setting, planning and combining of maintenance activities. *European Journal of Operational Research*, **82**:225–240, 1995.

[5] Dekker, R., J.B.G. Frenk, and R.E. Wildeman. An efficient optimal solution method for the joint replenishment problem. Technical Report 9542/A, Econometric Institute, Erasmus University Rotterdam (submitted for publication), 1995.

[6] Dekker, R. and I.F.K. Roelvink. Marginal cost criteria for preventive replacement of a group of components. *European Journal of Operational Research*, **84**(2):467–480, 1995.

[7] Goyal, S.K. Determination of economic packaging frequency for items jointly replenished. *Management Science*, **20**:293–298, 1973.

[8] Goyal, S.K. Determination of optimum packaging frequency of items jointly replenished. *Management Science*, **21**:436–443, 1974.

[9] Goyal, S.K. and A. Gunasekaran. Determining economic maintenance frequency of a transport fleet. *International Journal of Systems Science*, **4**(4):655–659, 1992.

[10] Goyal, S.K. and M.I. Kusy. Determining economic maintenance frequency for a family of machines. *Journal of the Operational Research Society*, **36**(12):1125–1128, 1985.

[11] Goyal, S.K. and A.T. Satir. Joint replenishment inventory control: Deterministic and stochastic models. *European Journal of Operational Research*, **38**:2–13, 1989.

[12] Hiriart-Urruty, J.-B. and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I: Fundamentals,* volume 305 of *A Series of Comprehensive Studies in Mathematics*. Springer-Verlag, Berlin, 1993.

[13] Horst, R. and P.M. Pardalos. *Handbook of Global Optimization*. Kluwer Academic Publishers, 1995.

[14] Howard, R.A. *Dynamic Programming and Markov Processes*. Wiley, 1960.

[15] Martos, B. *Nonlinear Programming Theory and Methods*. Akademiai Kiado, Budapest, 1975.

[16] Smeitink, E. and R. Dekker. A simple approximation to the renewal function. *IEEE Transactions on Reliability*, **39**(1):71–75, 1990.

[17] Van der Duyn Schouten, F.A. and S.G. Vanneste. Analysis and computation of (n, N)-strategies for maintenance of a two-component system. *European Journal of Operational Research*, **48**(2):260–274, 1990.

[18] Van Dijkhuizen, G. and A. Van Harten. Optimal clustering of repetitive frequency-constrained maintenance jobs with shared setups. Technical report, University of Twente, The Netherlands, 1995.

[19] Van Egmond, R., R. Dekker, and R.E. Wildeman. Correspondence: Determining economic maintenance frequency of a transport fleet. *International Journal of Systems Science*, **26**:1755–1757, 1995.

[20] Van Eijs, M.J.G., R.M.J. Heuts, and J.P.C. Kleijnen. Analysis and comparison of two strategies for multi-item inventory systems with joint replenishment costs. *European Journal of Operational Research*, **59**:405–412, 1992.