How to Determine Maintenance Frequencies for Multi-Component Systems? A General Approach

Rommert Dekker, Hans Frenk and Ralph E. Wildeman

Econometric Institute, Erasmus University Rotterdam, 3000 DR Rotterdam, The Netherlands

Summary. 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, multi-component, optimisation

1. Introduction

A technical system (such as a transportation fleet, a machine, a road, or a building) mostly contains many different components. The cost of maintaining a component of such a technical system often consists of a cost that depends on the component involved and of a fixed cost that only depends on the system. The system-dependent cost is called the *set-up cost* and is shared by all maintenance activities carried out simultaneously on components of the system. The set-up cost can consist of, for example, 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 on different components are executed simultaneously, since execution of a group of activities requires only one set-up. This can yield considerable cost savings, and therefore the development of optimisation models for multiple components is an important research issue.

For a literature overview of the field of maintenance of multi-component systems, we refer to Van der Duyn Schouten (1996) in this volume. Another review is given by Cho and Parlar (1991). 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 1960, 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 1985). For problems with many components heuristic methods can be applied. For instance, Dekker and Roelvink (1995) present a heuristic replacement criterion in case always a fixed group of components is replaced. Van der Duyn Schouten and Vanneste (1990) study structured strategies, viz. (n, N)-strategies, but provide an algorithm for only two identical components. Summarising, these models are of limited practical use, since reasonable numbers of components cannot be handled.

An approach that can handle many components was introduced by Goyal and Kusy (1985) and Goyal and Gunasekaran (1992). 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 more difficult to fit practical situations and 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 et al. 1995).

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 (1973); the joint-replenishment problem can be considered as a special case of the maintenance problem of Goyal and Kusy (1985). A method to solve the joint-replenishment problem to optimality was presented by Goyal (1974). 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 1989). But again, it is not clear how these heuristics will perform in case of the more general maintenance cost functions.

In this chapter we present a general approach for the coordination of maintenance frequencies, thereby pursuing the idea of Goyal and Gunesekaran (1992) and Goyal and Kusy (1985). With the approach we can easily solve the model of Goyal et al. to optimality, but we can also incorporate other maintenance models like minimal repair, inspection and block replacement.

We can also efficiently solve the joint-replenishment problem to optimality (see Dekker et al. 1995).

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 it is not good enough, 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. For the block-replacement model we will apply a good heuristic.

This chapter is structured as follows. In the next section we give the problem formulation. In Section 3 we rewrite the problem and we introduce a relaxation, which enables us 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, $i = 1, \dots, n$. 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 chapter we consider preventive maintenance activities of the block type, that is, the determination of 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 or modified block-replacement policies.

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 cumulative 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 and 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

$$\Phi_i(x) = \frac{c_i^p + M_i(x)}{x}, \; x > 0. \tag{2.1}$$

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. This implies that component i is preventively maintained each k_iT time units, $k_i \in \mathbb{N}$. The idea of modelling maintenance at fixed intervals that are integer multiples of a basis interval originates from inventory theory, see Goyal (1973). It was introduced in maintenance by Goyal and Kusy (1985) and further developed by Goyal and Gunasekaran (1992).

The objective now 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 $\Phi_i(k_iT)$ of each component i. The determination of the average set-up cost depends on how often an occasion for maintenance is actually used.

In the context of inventory theory a discussion in the literature has taken place on how to deal with so-called *empty* occasions that occur when the smallest integer k_i is larger than one. For example, suppose that there are two components and that $k_1 = 2$ and $k_2 = 3$, then two out of six occasions will not be used for maintenance. Dagpunar (1982) suggests that in that case on average only $4/6^{th}$ of the set-up cost is incurred. He proposes to use a correction factor $\Delta(k)$, $k = (k_1, \ldots, k_n)$. For example, if $k = (2,3)$ then $\Delta(k) = 4/6$. Dagpunar gives the following general expression for $\Delta(k)$:

$$\Delta(k) = \sum_{i=1}^{n} (-1)^{i+1} \sum_{\{\alpha \subset \{1,\ldots,n\} \,:\, |\alpha|=i\}} \{\mathrm{lcm}(k_{\alpha_1}, \ldots, k_{\alpha_i})\}^{-1}, \tag{2.2}$$

where $\mathrm{lcm}(k_{\alpha_1}, \ldots, k_{\alpha_i})$ denotes the least common multiple of the integers $k_{\alpha_1}, \ldots, k_{\alpha_i}$. Notice that $\Delta(k) \leq 1$ and that $\Delta(k) \geq (\min_i\{k_i\})^{-1}$. Consequently, if $\min_i\{k_i\} = 1$, then $\Delta(k) = 1$.

Goyal (1982), however, criticises the formulation of Dagpunar (1982). In the maintenance context (see Goyal and Kusy 1985 and Goyal and Gunasekaran 1992), but also in the formulation of the joint-replenishment problem found in the inventory literature, the correction factor is usually neglected, or equivalently, assumed to be equal to 1. This is correct under the assumption that the set-up cost is also incurred at occasions at which no actual maintenance is carried out.

We will consider here two different problem formulations, one with the correction factor and another without. With the correction factor we have the following problem:

$$\inf \left\{ \frac{S\Delta(k)}{T} + \sum_{i=1}^{n} \Phi_i(k_iT) \;:\; k_i \in \mathbb{N},\ T > 0 \right\}, \tag{2.3}$$

where $\Delta(k)$ is given by (2.2). If the correction factor $\Delta(k)$ is neglected, we have:

$$\inf\left\{\frac{S}{T}+\sum_{i=1}^{n}\Phi_i(k_iT)\ :\ k_i\in\mathbb{N},\ T>0\right\}. \tag{2.4}$$

Computation of the correction factor $\Delta(k)$ is in general time consuming. As was also pointed out by Goyal (1982) in the inventory context, minimisation of a cost function becomes considerably more complex if the correction factor $\Delta(k)$ is included. Together with the observation that problem (2.3) is a mixed continuous-integer programming problem, this makes (2.3) a very difficult problem to solve.

That is why we will focus in this chapter on problem (2.4). Although this problem is easier than problem (2.3), it is in general still difficult to solve. Approaches published so far include only computationally prohibitive enumeration methods or heuristics. However, in this chapter we will show that in many cases problem (2.4) can efficiently be solved to optimality. For the cases that this is not possible we present some heuristics that perform better than previously published ones. We will also discuss some results for problem (2.3). However, we will not consider a solution procedure for this problem. Observe that a solution of problem (2.4) can always be used as a feasible solution of problem (2.3) and we will show by numerical experiments in Section 5 that a feasible solution thus obtained will in many cases be sufficiently good.

In the modelling approach of Goyal and Kusy (1985) and Goyal and Gunasekaran (1992) only a very specific deterioration-cost function $M_i(\cdot)$ for component i, $i=1,\ldots,n$, is allowed. Here we allow more general deterioration-cost functions, so that this modelling approach can also be applied to well-known preventive-maintenance strategies of the block type, 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 1995, who provides an extensive list of these models; here we only mention some important ones).

Special Case of Goyal and Kusy. Goyal and Kusy (1985) use the following deterioration-cost function: $M_i(x)=\int_0^x(f_i+v_it^e)dt$, 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 the inventory literature, see also Dekker et al. (1995) (in that case the deterioration costs are holding costs).

Special Case of Goyal and Gunasekaran. The deterioration-cost function used by Goyal and Gunasekaran (1992) is slightly different from that of Goyal and Kusy (1985). They take $M_i(x)=\int_0^{Y_i(x-X_i)}(a_i+b_it)dt$, 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 1995), 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 1995), 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 1995), 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 construct a relaxation of the optimisation problems given by (2.3) and (2.4), and to simplify (2.4). Observe that the optimisation problems (2.3) and (2.4) allow for each component another function $M_i(\cdot)$. Thus it is possible to mix the different models above. It is possible, for instance, 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

To make optimisation problems (2.3) and (2.4) mathematically more tractable, we substitute T by $1/T$. Using this transformation, the relaxation for both problems that will be introduced in the next subsection 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. This result will be proved in Section 4 and there it will also be shown that without this transformation the relaxation is in general not a convex-programming problem. As will be seen later, this result is very useful in a solution procedure to solve problem (2.4).

Clearly, by the transformation $T \to 1/T$, the optimisation problem (2.3) is equivalent with

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

and optimisation problem (2.4) is equivalent with

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

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

Denote now by $v(P_c)$, $v(P)$ the optimal objective value of (P_c), (P) respectively, and by $T(P_c)$, $T(P)$ an optimal T (if it exists) for these problems. Notice that if $T(P_c)$ and $(k_1(T(P_c)), k_2(T(P_c)), \ldots, k_n(T(P_c))) \in \mathbb{N}^n$ are optimal for (P_c), then $T = 1/T(P_c)$ and the same values of k_i, $i = 1, \ldots, n$, are optimal for the optimisation problem (2.3). Analogously, if $T(P)$ and $(k_1(T(P)), k_2(T(P)), \ldots, k_n(T(P))) \in \mathbb{N}^n$ are optimal for (P), then $T = 1/T(P)$ and $(k_1(T(P)), k_2(T(P)), \ldots, k_n(T_P)))$ are optimal for problem (2.4).

3.1 A Relaxation of (P_c) and (P)

We will first introduce a relaxation of problem (P). As will be shown subsequently, the optimal objective value of this relaxation is also a lower bound on $v(P_c)$.

If we replace in (P) the constraints $k_i \in \mathbb{N}$ by $k_i \geq 1$, then we have the following optimisation problem:

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

Let $v(P_{rel})$ be the optimal objective value of (P_{rel}) and let $T(P_{rel})$ be a corresponding optimal solution of (P_{rel}) (if it exists).

For this relaxation it clearly follows that $v(P) \geq v(P_{rel})$. Without any assumptions on $\Phi_i(\cdot)$, it can be shown that $v(P_{rel})$ is also a lower bound on $v(P_c)$. This is established in the following lemma.

Lemma 3.1. *It follows that* $v(P) \geq v(P_c) \geq v(P_{rel})$.

Proof. Since for every vector $k = (k_1, \ldots, k_n)$ it holds that $\Delta(k) \leq 1$, the first inequality follows immediately. To prove the second inequality, we observe that for every $\epsilon > 0$ there exists a vector $(T_\epsilon, k_1(T_\epsilon), \ldots, k_n(T_\epsilon))$ satisfying

$$\begin{aligned} v(P_c) &\geq S\Delta(k(T_\epsilon))T_\epsilon + \sum_{i=1}^{n} \Phi_i\left(\frac{k_i(T_\epsilon)}{T_\epsilon}\right) - \epsilon \\ &= S\Delta(k(T_\epsilon))T_\epsilon + \sum_{i=1}^{n} \Phi_i\left(\frac{k_i(T_\epsilon)\Delta(k(T_\epsilon))}{\Delta(k(T_\epsilon))T_\epsilon}\right) - \epsilon. \end{aligned}$$

Since $\Delta(k(T_\epsilon)) \geq (\min_i\{k_i(T_\epsilon)\})^{-1}$, we have that $k_i(T_\epsilon)\Delta(k(T_\epsilon)) \geq 1$ for every i, and consequently

$$\begin{aligned} v(P_c) &\geq S\Delta(k(T_\epsilon))T_\epsilon + \sum_{i=1}^{n} \inf\left\{\Phi_i\left(\frac{k_i}{\Delta(k(T_\epsilon))T_\epsilon}\right) \ : \ k_i \geq 1\right\} - \epsilon \\ &\geq \inf_{T>0}\left\{ST + \sum_{i=1}^{n} \inf\left\{\Phi_i\left(\frac{k_i}{T}\right) \ : \ k_i \geq 1\right\}\right\} - \epsilon \\ &= v(P_{rel}) - \epsilon. \end{aligned}$$

Since $\epsilon > 0$ is arbitrary, the desired result follows. □

Since $v(P_{rel})$ equals $\inf_{T>0}\{ST + \sum_{i=1}^{n} \inf\{\Phi_i(k_i/T) \ : \ k_i \geq 1\}\}$, it is natural to impose the following assumption.

Assumption 3.1. *For each $i = 1, \ldots, n$ the optimisation problem (P_i) given by $\inf\{\Phi_i(x) \ : \ x > 0\}$ has a finite optimal solution $x_i^* > 0$.*

The problems (P_i) introduced in Assumption 3.1 are often easy to solve. For many single-component maintenance models the function $\Phi_i(\cdot)$ has a unique minimum and is strictly decreasing left of this minimum and strictly increasing right of it (i.e., the function $\Phi_i(\cdot)$ is *strictly unimodal*). In that case optimisation can be carried out with, for example, golden-section search (see Chapter 8 of Bazaraa et al. 1993). A more efficient algorithm to identify an optimal solution for a large class of single-component models is presented by Barros et al. (1995).

To continue our analysis, if optimisation problem (P_{rel}) can be solved and $T(P_{rel})$ is an optimal solution, then we can construct a feasible solution of (P_c) and (P) in the following way. Introduce the interval $I_i^{(k)} := [k/x_i^*, (k+1)/x_i^*]$, $k = 0, 1, \ldots$, and define the function $g_i(\cdot)$ as follows:

$$g_i(t) := \begin{cases} \Phi_i(1/t) & \text{if } t \in I_i^{(0)} \\ \min\{\Phi_i(k/t), \Phi_i((k+1)/t)\} & \text{if } t \in I_i^{(k)}, \ k = 1, 2, \ldots \end{cases} \tag{3.1}$$

Notice that for a given t, the value $g_i(t)$ and the corresponding integers $k_i(t)$ can easily be calculated once an optimal solution x_i^* of (P_i) is known. A given t lies within the interval $I_i^{(k)}$ for which $k = \lfloor tx_i^* \rfloor$, 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 $g_i(t)$, and $k_i(t)$ equals 1. Otherwise, if $k \geq 1$, two function evaluations are necessary and $k_i(t)$ equals k or $k+1$, depending on whether $\Phi_i(k/t) \leq \Phi_i((k+1)/t)$ or $\Phi_i(k/t) \geq \Phi_i((k+1)/t)$.

Using (3.1), it is easy to calculate the integers $k_i(T(P_{rel}))$ corresponding with $T(P_{rel})$, and it is clear that $(T(P_{rel}), k_1(T(P_{rel})), \ldots, k_n(T(P_{rel})))$ is a feasible solution for both (P_c) and (P).

In Section 4 we show that under certain conditions on the functions $\Phi_i(\cdot)$ it holds that $g_i(t) = \inf\{\Phi_i(k_i/t) \ : \ k_i \in \mathbb{N}\}$. However, without any conditions on these functions we only have that $\inf\{\Phi_i(k_i/t) \ : \ k_i \in \mathbb{N}\} \leq g_i(t)$ for every $t > 0$ and so by the definition of (P) it follows that $v(P) \leq ST(P_{rel}) + \sum_{i=1}^{n} g_i(T(P_{rel}))$. Hence, if the value $ST(P_{rel}) + \sum_{i=1}^{n} g_i(T(P_{rel}))$ is close enough to $v(P_{rel})$, we can decide, due to

$$v(P_{rel}) \leq v(P_c) \leq v(P) \leq ST(P_{rel}) + \sum_{i=1}^{n} g_i(T(P_{rel})),$$

that $(T(P_{rel}), k_1(T(P_{rel})), \ldots, k_n(T(P_{rel})))$ is a reasonable feasible solution of problem (P_c) and of problem (P).

To analyse now under Assumption 3.1 the optimisation problem (P_{rel}), observe that for every $T \geq 1/x_i^*$ it holds that $\inf\{\Phi_i(k_i/T) \ : \ k_i \geq 1\} = \Phi_i(x_i^*)$. By this observation the following result is easy to prove. This result will be used in a procedure to solve (P_{rel}).

Lemma 3.2. *If we assume without loss of generality that $1/x_n^* \leq 1/x_{n-1}^* \leq \cdots \leq 1/x_1^*$, then for any optimal $T(P_{rel})$ of (P_{rel}) it follows that $T(P_{rel}) \leq 1/x_1^*$.*

Proof. For all $T > 1/x_1^*$ we obtain that

$$\begin{aligned} ST + \sum_{i=1}^{n} \inf\{\Phi_i(k_i/T) \ : \ k_i \geq 1\} &= ST + \sum_{i=1}^{n} \Phi_i(x_i^*) \\ > \ \frac{S}{x_1^*} + \sum_{i=1}^{n} \Phi_i(x_i^*) &= \frac{S}{x_1^*} + \sum_{i=1}^{n} \inf\{\Phi_i(k_i x_1^*) \ : \ k_i \geq 1\}, \end{aligned}$$

and so for every $T > 1/x_1^*$ the objective function of (P_{rel}) evaluated in T is larger than the objective function evaluated in the point $1/x_1^*$. This implies $T(P_{rel}) \leq 1/x_1^*$ and the desired result is proved. □

In Section 4 we will simplify the objective function of problem (P_{rel}) by imposing some assumptions on the functions $\Phi_i(\cdot)$. In order to simplify the objective function of problem (P), we also need some assumptions on the same functions $\Phi_i(\cdot)$. However, before introducing these assumptions, we discuss the literature on problem (P).

3.2 Literature on Problem (P)

Goyal and Kusy (1985) and Goyal and Gunasekaran (1992) apply an iterative algorithm to solve problem (2.4) in the previous section (equivalent with (P)) 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 (2.4) as a function of T equal to zero. Subsequently, the authors find for each i a value of k_i, in two different ways. Goyal and Kusy (1985) 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 (1992) find for each i the optimal *real-valued* k_i by setting the derivative of the cost function of (2.4) as a function of k_i to zero and 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 et al. 1995). 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 of the obtained solutions.

In the inventory theory literature many heuristics have appeared for the special cost functions in the joint-replenishment problem (see Goyal and Satir 1989). Although some heuristics can be modified to deal with the cost functions of maintenance optimisation, the performance of these heuristics cannot be guaranteed.

Altogether, the literature does not provide an efficient and reliable approach to solve problem (P). That is why we will focus in this chapter on an alternative solution approach that is based on the global optimisation of (P). In order to do so, we need to simplify the objective function of problem (P), which will be done in the next subsection. In a solution procedure for (P) (discussed in Section 4) we first find a solution to problem (P_{rel}) and by using (3.1) we then obtain a feasible solution to (P) (and hence also to (P_c)). Since

$v(P_{rel})$ is a lower bound on both $v(P_c)$ and $v(P)$, we can decide whether this feasible solution is good enough.

If this feasible solution is not good enough, we subsequently apply a global-optimisation procedure to the simplified problem (P) in an interval that is obtained by the relaxation and that contains an optimal $T(P)$. For the special cases of Goyal et al., the minimal-repair model and the inspection model it is then possible to find in little time a solution to (P) with an objective value that has an arbitrarily small deviation from the optimal value $v(P)$. 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. Our approach can also be applied to find an optimal solution to the joint-replenishment problem, see Dekker et al. (1995). In that case the procedure can be made even more efficient, since the cost functions in that problem have a very simple form.

With a solution to problem (P), we then have an improved upper bound $v(P)$ on $v(P_c)$. If this is close to $v(P_{rel})$, then it is by Lemma 3.1 also close to $v(P_c)$ and so we have a good solution of (P_c) as well.

We will now simplify under certain conditions the objective function of problem (P).

3.3 Simplification of Problem (P)

To simplify the objective function of problem (P), we introduce the following definition and assumption (for Definition 3.1 see also Chapter 3 of Avriel et al. 1988).

Definition 3.1. *A function $f(x)$, $x \in (0,\infty)$, is called unimodal on $(0,\infty)$ 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 on $(0,\infty)$ with respect to $b = 0$.

Assumption 3.2. *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 the function $\Phi_i(\cdot)$ is unimodal on $(0,\infty)$ with respect to x_i^*.*

By Assumption 3.2 the objective function of problem (P) can be simplified considerably. To this end consider the interval $I_i^{(k)} := [k/x_i^*, (k+1)/x_i^*]$, $k = 0,1,\ldots$, introduced in Section 3.1 and observe that if $t \in I_i^{(k)}$ and $k \geq 1$, then 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 3.2 the function $\Phi_i(\cdot)$ is unimodal on $(0,\infty)$ 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

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

Analogously, if $t \in I_i^{(0)}$ and $t > 0$, then it holds that $x_i^* \leq 1/t$, so that

$$x_i^* \leq 1/t \leq 2/t \leq 3/t \leq \cdots,$$

and consequently

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

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

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

Consequently, if we define $g_i(\cdot)$ as in (3.1), if follows that

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

It is not difficult to verify that by Assumption 3.2 and the fact that $\Phi_i(\cdot)$ is continuous, the function $g_i(\cdot)$ is also continuous. In Figure 3.1 an example of the function $g_i(\cdot)$ is given.

Under Assumption 3.2 the optimisation problem (P) has the following simplified representation:

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

with $g_i(\cdot)$ given by (3.1).

Below we introduce a class of functions $\Phi_i(\cdot)$ that satisfy Assumption 3.2. To this end we need the next result.

Lemma 3.3. *If the function $M_i(\cdot)$ is convex on (b_i,∞) for some $b_i \geq 0$, then it follows that $c := \lim_{x\to\infty} M_i(x)/x \leq \infty$ exists. Moreover, the function $M_i(x) - xc$ is decreasing on (b_i,∞).*

Proof. Since the function $M_i(\cdot)$ is convex on (b_i,∞) if follows by applying the well-known criterion of increasing slopes valid for convex functions (see Proposition 1.1.4 in Chapter I of Hiriart-Urruty and Lemaréchal 1993) that for any fixed $y > b_i$ the function $x \to (M_i(x)-M_i(y))/(x-y)$ is increasing on $(b_i,\infty)\setminus\{y\}$. This implies that $\infty \geq \lim_{x\to\infty}(M_i(x) - M_i(y))/(x-y)$ exists and clearly this limit equals $c := \lim_{x\to\infty} M_i(x)/x$. To prove the second part

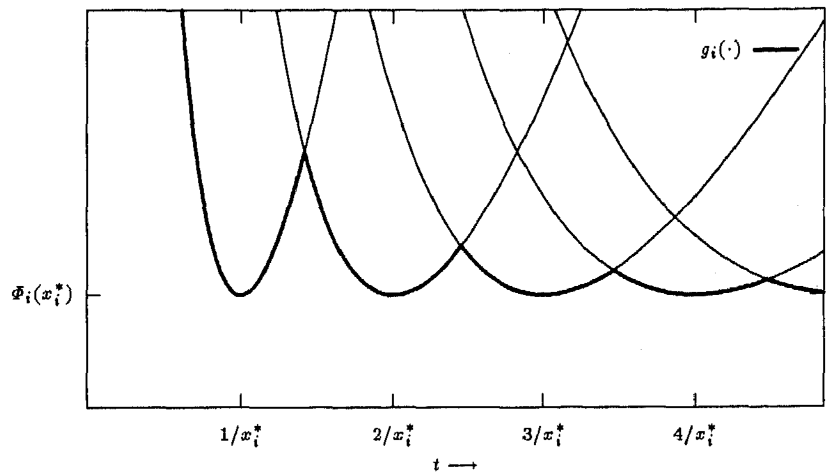

Fig. 3.1. An example of the function $g_i(\cdot)$. The thin lines are the graphs of the functions $\Phi_i(1/t), \Phi_i(2/t), \ldots, \Phi_i(5/t)$. The (bold) graph of $g_i(\cdot)$ is the lower envelope of these functions.

we only need to consider $c < \infty$. Observe now that for any y with $b_i < y < x$ it holds that

$$(M_i(x) - xc) - (M_i(y) - yc) = \left(\frac{M_i(x) - M_i(y)}{x - y} - c\right)(x - y).$$

Since by the first part of this lemma we have that the function $x \longrightarrow (M_i(x) - M_i(y))/(x-y)$ is increasing on (y, ∞) and $\lim_{x\to\infty}(M_i(x)-M_i(y))/(x-y) = c$, it follows by the above equality that $M_i(x) - xc \leq M_i(y) - yc$. □

Using Lemma 3.3 we can show the following result. Observe that the first part of this lemma improves a result given by Dekker (1995).

Lemma 3.4. *If $M_i(\cdot)$ is concave on $(0, b_i)$ and convex on (b_i, ∞) for some $b_i \geq 0$, then the set of optimal solutions of the optimisation problem (P_i) given by $\inf\{\Phi_i(x) \ : \ x > 0\}$ is nonempty and compact if and only if $\lim_{x\to\infty} M_i(x) - xc < -c_i^p$, with $c := \lim_{x\to\infty} M_i(x)/x$. Moreover, it follows for any optimal solution x_i^* of (P_i) that $x_i^* \geq b_i$ and that the function $\Phi_i(\cdot)$ is unimodal on $(0, \infty)$ with respect to x_i^*.*

Proof. If for some $b_i > 0$ the function $M_i(\cdot)$ is concave on $(0, b_i)$ then the function $c_i^p + M_i(\cdot)$ is also concave on $(0, b_i)$. This implies for every $0 < z_1 < z_2 < b_i$ 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 (2.1) it follows that $\Phi_i(z_1) > \Phi_i(z_2)$ and, consequently, that $\Phi_i(\cdot)$

is strictly decreasing on $(0, b_i)$. By this observation it follows that if $x_i^* > 0$ is an optimal solution of (P_i), then necessarily $x_i^* \geq b_i$. On the other hand, if $b_i = 0$, then by the feasibility of x_i^* we also have that $x_i^* \geq b_i$ and this proves the second part of the lemma (that is, $x_i^* \geq b_i$ for any optimal solution x_i^*).

To verify the 'only-if' proposition, observe that the optimal objective value $v(P_i)$ of (P_i) is smaller than $\Phi_i(\infty)$, since the optimal solution set of (P_i) is nonempty and compact. By the first part of the proof and the continuity of $\Phi_i(\cdot)$ this yields the existence of some $x_0 > b_i$ such that $\Phi_i(x_0) < \Phi_i(\infty)$. The first part of Lemma 3.3 shows that $c := \lim_{x\to\infty} M_i(x)/x$ exists and since $\Phi_i(\infty) = \lim_{x\to\infty} M_i(x)/x = c$ and $\Phi_i(x_0) < \Phi_i(\infty)$, it follows that $(c_i^p + M_i(x_0))/x_0 < c$ or, equivalently, $M_i(x_0) - x_0 c < -c_i^p$. Using now that the function $M_i(x) - xc$ is decreasing on (b_i, ∞) (Lemma 3.3), we have that $\lim_{x\to\infty} M_i(x) - xc < -c_i^p$. To verify the other inclusion (the 'if' proposition), observe that $\lim_{x\to\infty} M_i(x) - xc < -c_i^p$ implies that there exists some $x_0 \in (0, \infty)$ satisfying $M_i(x_0) - x_0 c < -c_i^p$ or, equivalently, $\Phi_i(x_0) < \Phi_i(\infty)$. This yields that $v(P_i) < \Phi_i(\infty)$ and since also $\Phi_i(0) = \infty$ and $\Phi_i(\cdot)$ is continuous, this implies that the optimal solution set of (P_i) is nonempty and compact.

To verify the last result, observe since $M_i(\cdot)$ is convex on (b_i, ∞), that the function $c_i^p + M_i(\cdot)$ is also convex on (b_i, ∞). By Theorem 3.51 of Martos (1975) this implies that $\Phi_i(t) = (c_i^p + M_i(t))/t$ is a so-called quasiconvex function. Using that $\inf\{\Phi_i(x) \ : \ x \geq b_i\}$ has an optimal solution $x_i^* \geq b_i$, we then obtain by Proposition 3.8 of Avriel et al. (1988) that $\Phi_i(\cdot)$ is unimodal on (b_i, ∞) with respect to x_i^*. Together with the result that the continuous function $\Phi_i(\cdot)$ is strictly decreasing on $(0, b_i)$ the desired result follows, that is, $\Phi_i(\cdot)$ is unimodal on $(0, \infty)$ with respect to x_i^*. □

Now we can show that in general the special cases of Goyal et al., the minimal-repair model and the inspection model satisfy Assumption 3.2 when the optimisation problem (P_i) has a finite optimal solution $x_i^* > 0$.

Theorem 3.1. *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 3.2 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 3.4 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 3.4 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 3.4 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 3.4 we showed that for this 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 (P_{rel}) of (P) is a convex-programming problem (see Lemma 4.2).
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 3.4 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 3.2 is satisfied. □

Observe that by Lemma 3.4 an easy necessary and sufficient condition for the existence of only finite optimal solutions of (P_i) is presented for both cases 3 and 4 above.

In Figure 3.2 an example of the objective function of problem (P) under Assumption 3.2 is given. In general this objective function 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 problem (P_{rel}) is considered, often a much easier problem is obtained; for the special cases of Goyal et al., the minimal-repair model and the inspection model the relaxation (P_{rel}) turns out to be a single-variable convex-programming problem and so it is easy to solve.

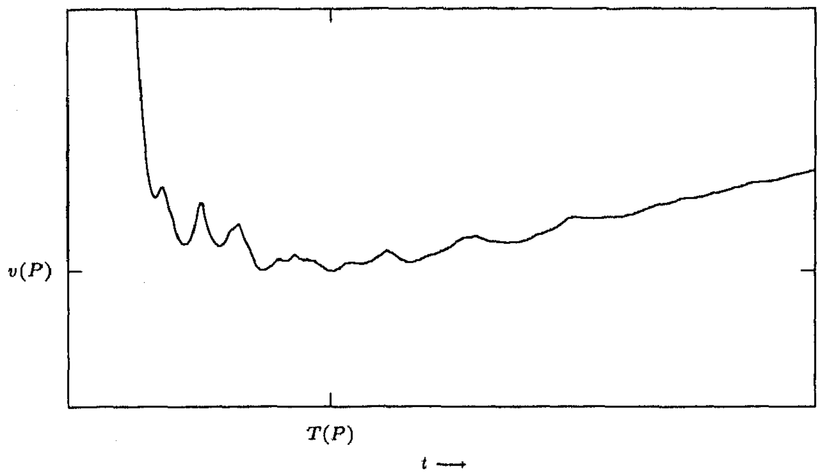

Fig. 3.2. An example of the objective function of problem (P); there are many local minima.

4. Solving Problem (P)

In this section we discuss under some additional assumptions on the functions $\Phi_i(\cdot)$ a computationally fast solution procedure for problem (P). This yields an optimal solution $(T(P), k_1(T(P)), \ldots, k_n(T(P)))$ of (P). With respect to problem (P_c) we observe that the optimal solution of (P) is also feasible for (P_c). Moreover, if there exists a $(T(P_c), k_1(T(P_c)), \ldots, k_n(T(P_c)))$ that is optimal for (P_c) with $\Delta(k(T(P_c))) = 1$, it follows by Lemma 3.1 that $v(P) = v(P_c)$, so that in that case $(T(P), k_1(T(P)), \ldots, k_n(T(P)))$ is also an optimal solution of (P_c). For this solution it follows automatically that $\Delta(k(T(P))) = 1$, so that if for the generated optimal solution $(T(P), k_1(T(P)), \ldots, k_n(T(P)))$ of (P) it holds that $\Delta(k(T(P))) < 1$, then this implies that $v(P_c) < v(P)$.

To start our approach to tackle problem (P), we first find out under which conditions the relaxation (P_{rel}) introduced in Section 3.1 is easy to solve.

4.1 Analysis of (P_{rel})

To simplify problem (P_{rel}) we only need a much weaker assumption than Assumption 3.2 discussed in the previous section.

Assumption 4.1. *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 $i = 1, \ldots, n$ it holds that $\Phi_i(\cdot)$ is increasing on (x_i^*, ∞).*

Theorem 3.1 showed for the special cases of Goyal et al., the minimal-repair model with a unimodal rate of occurrence of failures and the inspection model, that Assumption 3.2 is satisfied when (P_i) has a finite solution $x_i^* > 0$. As a result, also Assumption 4.1 is satisfied for these models.

By Assumption 4.1 the objective function of problem (P_{rel}) can be simplified. Analogously to equation (3.1) we have for

$$g_i^{(R)}(t) := \begin{cases} \Phi_i(1/t) & \text{if } t \leq 1/x_i^* \\ \Phi_i(x_i^*) & \text{if } t \geq 1/x_i^* \end{cases} \tag{4.1}$$

that $g_i^{(R)}(t) = \inf\{\Phi_i(k_i/t) \ : \ k_i \geq 1\}$. In Figure 4.1 an example of the function $g_i^{(R)}(\cdot)$ is given.

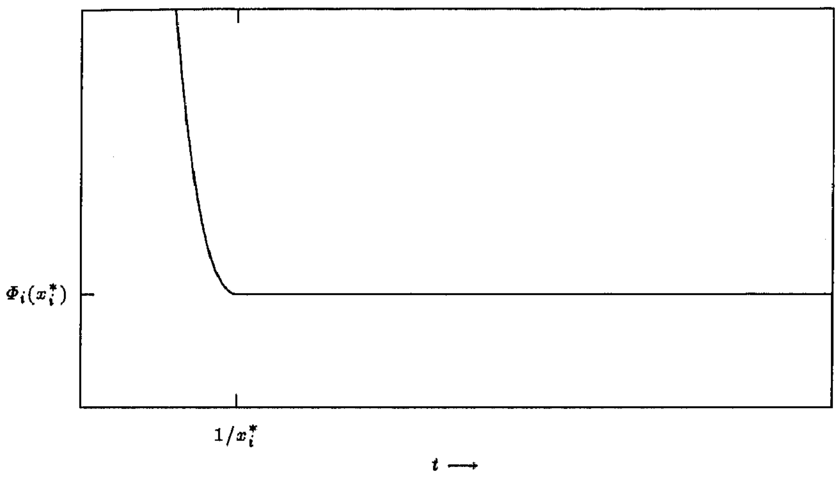

Fig. 4.1. An example of the function $g_i^{(R)}(\cdot)$. Notice the similarity with the graph of $g_i(\cdot)$ in Figure 3.1.

Now (P_{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 4.1 it follows that $v(R) = v(P_{rel})$ and $T(R) = T(P_{rel})$, since (R) and (P_{rel}) are equivalent under this assumption. Remember, if we use (R) we always assume that Assumption 4.1 holds.

We will now consider a class of functions $\Phi_i(\cdot)$ that satisfy Assumption 4.1.

Lemma 4.1. *If the optimisation problem (P_i) given by $\inf\{\Phi_i(x) \,:\, x > 0\}$ has a finite optimal solution $x_i^* > 0$ and the function $M_i(\cdot)$ is convex on (x_i^*, ∞), then the function $\Phi_i(\cdot)$ is increasing on (x_i^*, ∞).*

Proof. Since the function $M_i(\cdot)$ is convex on (x_i^*, ∞), if follows by Theorem 3.51 of Martos (1975) that $\Phi_i(t) = (c_i^p + M_i(t))/t$ is a so-called quasiconvex function on (x_i^*, ∞). Since $\inf\{\Phi_i(x) \,:\, x > 0\}$ has an optimal solution $x_i^* > 0$, the desired result follows by Proposition 3.8 of Avriel et al. (1988). □

Under the same condition as imposed in Lemma 4.1, one can prove additionally that the function $g_i^{(R)}(\cdot)$ is convex. Consequently, if the condition of Lemma 4.1 holds for each i, the optimisation problem (R) is a univariate convex-programming problem and so it is easy to solve. The convexity of the function $g_i^{(R)}(\cdot)$ is established by the following lemma.

Lemma 4.2. *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 $g_i^{(R)}(\cdot)$ is convex it is sufficient to show that the function $t \rightarrow \Phi_i(1/t)$ is convex on $(0, 1/x_i^*)$. If $t \rightarrow \Phi_i(1/t)$ 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 the minimal value of $\Phi_i(\cdot)$ on $(0, \infty)$, that $t \rightarrow \Phi_i(1/t)$ is also decreasing on $(0, 1/x_i^*)$ and then it follows from the definition of $g_i^{(R)}(\cdot)$ (see (4.1)) that $g_i^{(R)}(\cdot)$ is convex on $(0, \infty)$.

So we have to prove that $t \rightarrow \Phi_i(1/t)$ is convex on $(0, 1/x_i^*)$. We will prove that $t \rightarrow \Phi_i(1/t)$ is convex on $(0, 1/b_i)$ if $M_i(\cdot)$ is convex on (b_i, ∞). So let $M_i(\cdot)$ be convex on (b_i, ∞), then $c_i^p + M_i(t) = t\Phi_i(t)$ is also convex on (b_i, ∞). Define now 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 $g(t) := \Phi_i(1/t)$. It is easy to verify that

$$s_f(t, t_0) = \Phi_i(t_0) - (1/t_0)s_g(1/t, 1/t_0). \tag{4.2}$$

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 1993), yields 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 $t_0 > b_i$. By (4.2) 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 we obtain that $g(t) = \Phi_i(1/t)$ is convex on $(0, 1/b_i)$.

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

We can now apply the above results to the special cases of Goyal et al., the minimal-repair model and the inspection model.

Theorem 4.1. *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 problem (P_{rel}) is equivalent with problem (R) and (R) is a convex-programming problem.*

Proof. In the proof of Theorem 3.1 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 4.2 the corresponding $g_i^{(R)}(\cdot)$ are convex so that problem (R) is a convex-programming problem. □

In Figure 4.2 an example of the objective function of problem (R) is given.

We can now explain why we applied in the previous section the transformation of T into $1/T$ in the original optimisation problem (2.4). We saw that (R) is a convex-programming problem if each function $g_i^{(R)}(\cdot)$ is convex on $(0, \infty)$. In the proof of Lemma 4.2 we showed that this is the case if each function $t \longrightarrow \Phi_i(1/t)$ is convex on $(0, 1/x_i^*)$. We showed furthermore that the function $t \longrightarrow \Phi_i(1/t)$ is convex on $(0, 1/x_i^*)$ if $M_i(\cdot)$ is convex on (x_i^*, ∞) (which is generally the case for the models described before). If we did not apply the transformation of T into $1/T$, we would obtain that the corresponding relaxation is a convex-programming problem only if each function $\Phi_i(\cdot)$ is convex on (x_i^*, ∞). This is a much more restrictive condition and it is in general not true (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

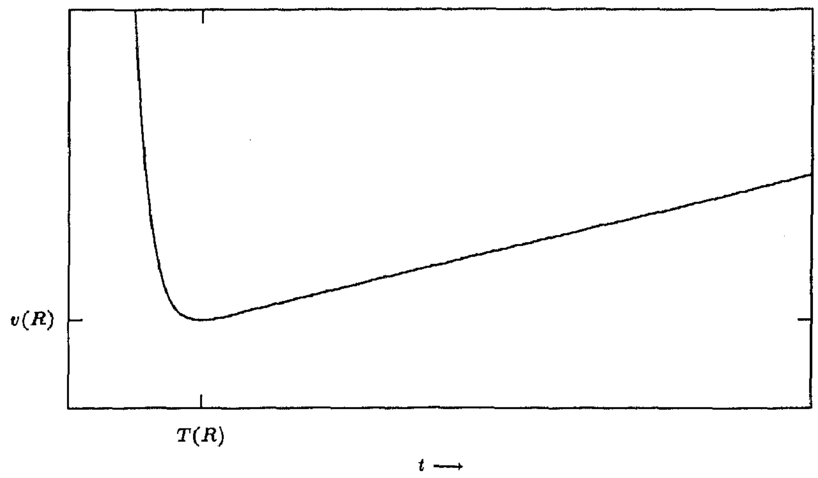

Fig. 4.2. An example of the objective function of problem (R).

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 et al. 1993).

To apply these procedures it is necessary to have a lower and an upper bound on an optimal value $T(R)$. If we assume again without loss of generality that $1/x_n^* \leq 1/x_{n-1}^* \leq \cdots \leq 1/x_1^*$, then for any optimal $T(R)$ of (R) it follows by Lemma 3.2 that $0 < T(R) \leq 1/x_1^*$.

Once (R) is solved we have an optimal $T(R)$. If additionally (R) is a convex-programming problem, it is possible to derive an easy dominance result for the optimal solution of (P). In order to do so, we first need the following lemma.

Lemma 4.3. *If $T(R) \leq 1/x_n^*$ is an optimal solution of problem (R) (and Assumption 4.1 holds), then $(T(R), 1, \ldots, 1)$ is an optimal solution of (P_c) and of (P). Moreover, if there does not exist an optimal solution of (R) within the interval $(0, 1/x_n^*)$, then any optimal solution $T(P)$ of (P) is bounded from below by $1/x_n^*$.*

Proof. Since $T(R) \leq 1/x_n^*$ is an optimal solution of problem (R) it follows by Assumption 4.1 that the optimal scalars $k_i(T(R))$, $i = 1, \ldots, n$, are equal to one and so $(T(R), k_1(T(R)), \ldots, k_n(T(R)))$ is also a feasible solution of problem (P_c) and (P). Hence we obtain that $v(R) = ST(R) + \sum_{i=1}^{n} \Phi_i(1/T(R)) \geq$

$v(P)$ and this yields by Lemma 3.1 that $v(R) = v(P_c) = v(P)$, implying that $(T(R), 1, \ldots, 1)$ is also an optimal solution of (P_c) and of (P).

To prove the second part, observe since the functions $g_i(\cdot)$ and $g_i^{(R)}(\cdot)$ (see (3.1) and (4.1)) are identical on $(0, 1/x_n^*]$ (by Assumption 4.1), that $T < 1/x_n^*$ is a local optimal solution of problem (R) if and only if T is a local optimal solution of problem (P). Hence, if there is no local optimal solution of (R) within the interval $(0, 1/x_n^*)$, then there is no local optimal solution of problem (P) within $(0, 1/x_n^*)$, and this yields $T(P) \geq 1/x_n^*$. □

If (R) is a convex-programming problem, then Lemma 4.3 yields the following result. If $T(R) \leq 1/x_n^*$ then $T(R)$ is an optimal solution of (P_c) and (P) and the optimal scalars k_i, $i = 1, \ldots, n$, are equal to one. If $T(R) > 1/x_n^*$, we evaluate the objective function of (R) in $1/x_n^*$ and if this value equals $v(R)$ then $1/x_n^*$ is also an optimal solution of (R), so that by Lemma 4.3 it follows that $1/x_n^*$ is an optimal solution of (P_c) and (P) as well. Finally, if the objective function of (R) in $1/x_n^*$ is larger than $v(R)$, then there does not exist a local optimal solution of (R) within $(0, 1/x_n^*)$ (since the objective function of (R) is convex) and thus it follows by Lemma 4.3 that $T(P) \geq 1/x_n^*$. Consequently, we have shown the following corollary.

Corollary 4.1. *Suppose (R) is a convex-programming problem. If $T(R) > 1/x_n^*$ and the objective function of (R) in $1/x_n^*$ is larger than $v(R)$, then for any optimal solution $T(P)$ of (P) it follows that $T(P) \geq 1/x_n^*$. Otherwise, an optimal $T(P)$ is given by $T(P) = \min\{1/x_n^*, T(R)\}$.*

Observe for $T(R) > 1/x_n^*$ that $T(R)$ may not be an optimal solution of problem (P). Besides, the values of k_i corresponding with $T(R)$ are not necessarily integer, implying that the optimal solution of (R) is in general not feasible for (P) 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 (P) (which is consequently also a feasible solution for problem (P_c)).

4.2 Feasibility Procedures for (P_c) and (P)

A straightforward way for finding a feasible solution for (P_c) and (P) is to substitute the value of $T(R)$ in (3.1). This is specified by 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 (P) corresponding with $T(R)$ (use (3.1)).

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 (3.1)).

If $v(FP)$ denotes the objective value $ST(R)+\sum_{i=1}^{n} g_i(T(R))$ then clearly by the definition of $g_i(\cdot)$ and Assumption 4.1 it follows that

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

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

If it is not close enough we may find a better solution by applying a procedure that is similar to the iterative approach of Goyal et al. We will call this procedure 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
$$\inf_{T>0} \left\{ ST + \sum_{i=1}^{n} \Phi_i(k_i(IFP)/T) \right\}, \tag{4.3}$$
and let $T(IFP)$ be an optimal value for T.
3. Determine new constants $k_i(IFP)$ by substitution of $T(IFP)$ in (3.1). This implies the application of the feasibility procedure FP to the value $T(IFP)$. Let $v(IFP)$ be the corresponding objective value.

Under Assumption 3.2 it follows for the value $v(IFP)$ generated by the IFP that

$$\begin{aligned}
v(IFP) &= ST(IFP) + \sum_{i=1}^{n} g_i(T(IFP)) \\
&= ST(IFP) + \sum_{i=1}^{n} \Phi_i(k_i(IFP)/T(IFP)) \quad \text{(Assumption 3.2)} \\
&\leq ST(IFP) + \sum_{i=1}^{n} \Phi_i(k_i(FP)/T(IFP)) \quad \text{(Step 3.)} \\
&\leq ST(R) + \sum_{i=1}^{n} \Phi_i(k_i(FP)/T(R)) \qquad \text{(Step 2.)} \\
&= ST(R) + \sum_{i=1}^{n} g_i(T(R)) \\
&= v(FP).
\end{aligned}$$

Consequently, if Assumption 3.2 holds, 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 (1985) 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 (1992) 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 under Assumption 3.2 optimal values of k_i for a given value of T *immediately*, by substitution of T in (3.1) (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 et al. 1995). In the IFP we start with a value of T that is optimal for (R) and hence might be a good solution for (P) 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 the alternative initialisation the IFP does 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 the solution is good. 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 (4.3) is a convex-programming problem (the functions $\Phi_i(1/t)$ are then convex). Otherwise, the IFP may not be useful since (4.3) 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 $T(P)$.

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

In this subsection we will derive lower and upper bounds on $T(P)$. Corollary 4.1 already provides a lower bound $1/x_n^*$ if $T(R) > 1/x_n^*$ and (R) is a convex-programming problem.

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 lower bound on $T(P)$ that is at least as good as $1/x_n^*$. This is established by the following lemma.

Lemma 4.4. *Consider the optimisation problem:*

$$(P_1) \qquad \inf_{T>0}\left\{ST + \sum_{i=1}^{n} \Phi_i(1/T)\right\},$$

with $v(P_1)$ the optimal objective value and $T(P_1)$ an optimal T. If for each $i = 1,\dots,n$ the function $M_i(\cdot)$ is convex and differentiable on $(0,\infty)$, and for at least one $i \in \{1,\dots,n\}$ 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 $T(P) \geq T(P_1) \geq 1/x_n^*$.

Proof. If there does not exist a global optimal solution of (R) in $(0, 1/x_n^*)$, then it can be shown analogously to Lemma 4.3 that $T(P_1) \geq 1/x_n^*$.

To prove the inequality $T(P) \geq T(P_1)$, notice first that (P_1) equals the optimisation problem (P) when all k_i are fixed to the value 1. Consequently, (P_1) is a more restricted problem than (P) and it is easy to verify that $v(P) \leq v(P_1)$. Furthermore, if $T(P)$ and certain values of k_i are optimal for (P), then it is easy to see that if the functions $\Phi_i(\cdot)$ are differentiable the following holds:

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

so that

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

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

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

It is easily verified that

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

so that

$$v(P) = \sum_{i=1}^{n} M_i'(k_i/T(P)). \tag{4.4}$$

Analogously, it can be shown for the optimal objective value of (P_1) that

$$v(P_1) = \sum_{i=1}^{n} M_i'(1/T(P_1)). \tag{4.5}$$

Suppose now that the inequality $T(P) \geq T(P_1)$ does not hold, that is, $T(P) < T(P_1)$. Since the functions $M_i(\cdot)$ are (strictly) convex and, consequently, the functions $M_i'(\cdot)$ are (strictly) increasing, this implies that (use (4.4) and (4.5))

$$\begin{aligned} v(P) &= \sum_{i=1}^{n} M_i'(k_i/T(P)) \\ &\geq \sum_{i=1}^{n} M_i'(1/T(P)) \\ &> \sum_{i=1}^{n} M_i'(1/T(P_1)) \\ &= v(P_1), \end{aligned}$$

which is in contradiction with $v(P) \leq v(P_1)$. Hence, $T(P) \geq T(P_1)$. □

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

Lemma 4.5. *For an optimal $T(P)$ of (P) it holds that*

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

with $v(FP)$ the objective value corresponding with the feasible solution of (P) 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 $T(P)$ is such a T for which $ST + \sum_{i=1}^{n} g_i(T) \leq v(FP)$, the lemma follows. □

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

Lemma 4.6. *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 $T(P)$. Moreover, this upper bound is smaller than or equal to the upper bound according to Lemma 4.5.*

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 (P), the objective function of (P) is also larger than or equal to $v(FP)$ for values of $T > T_{up}$, so that T_{up} is an upper bound on $T(P)$.

It is easy to see that this upper bound is at least as good as the upper bound of Lemma 4.5, 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 $T(P)$ analogously.

Lemma 4.7. *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 $T(P)$.*

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 (P), the objective function of (P) is also larger than or equal to $v(FP)$ for values of $T < T_{low}$, so that T_{low} is a lower bound on $T(P)$. □

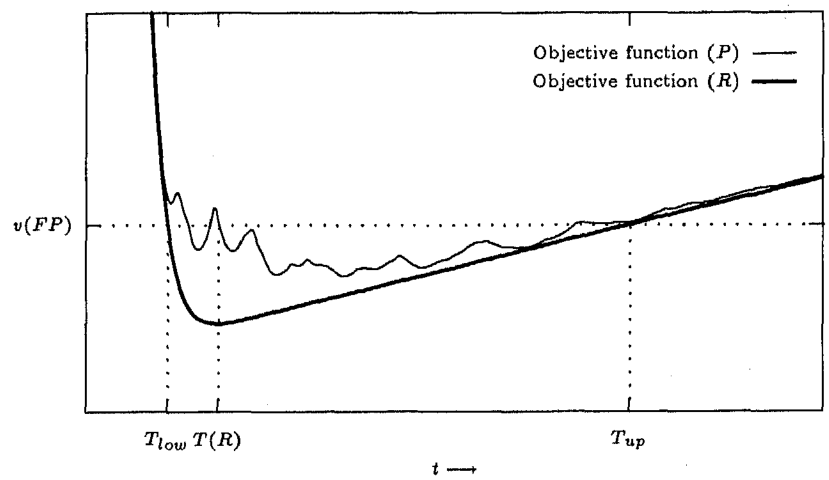

Fig. 4.3. A lower bound T_{low} and an upper bound T_{up} on an optimal $T(P)$ are found where the objective function of relaxation (R) equals $v(FP)$, the value of the objective function of problem (P) in $T(R)$.

In Figure 4.3 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 4.1. 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 $[1/x_n^*, T(R)]$.

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 lower and upper bounds on $T(P)$. The results are summarised in Table 4.1.

Table 4.1. Lower and Upper Bounds on $T(P)$

Lower bound	Condition
$1/x_n^*$	There is no optimal solution of (R) in $(0, 1/x_n^*)$
$T(P_1)$	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**
$(1/S)\{v(FP) - \sum_{i=1}^{n} \Phi_i(x_i^*)\}$	none
T_{up}	(R) is a convex-programming problem

From Table 4.1 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 4.1 that (R) is a convex programming problem. This is already sufficient to use all bounds of Table 4.1, except the lower bound $T(P_1)$. To use the bound $T(P_1)$, each $M_i(\cdot)$ must be convex on $(0,\infty)$ and at least one $M_i(\cdot)$ must be strictly convex. We showed in the proof of Theorem 3.1 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 $T(P_1)$ 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 $T(P) \in [T_l, T_u]$. Consequently, it is sufficient to apply a global-optimisation technique on the interval $[T_l, T_u]$ to find a value for $T(P)$.

4.4 Global-Optimisation Techniques

What remains to be specified is the usage of a global-optimisation technique for (P) on the interval $[T_l, T_u]$ when the feasible solution to (P) 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 (P) 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 4.1. *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 (P) 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 $v(P)$ (see the chapter on Lipschitz optimisation in Horst and Pardalos 1995). 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 (P) is Lipschitz on $[T_l, T_u]$, and we can derive a Lipschitz constant (see Appendix A).

There are several Lipschitz-optimisation algorithms (see Horst and Pardalos 1995), 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 (1995) 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 (P) is such that the Lipschitz constant is decreasing in T (this is shown in Appendix A). 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 (P) 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 we apply on each a golden-section search again. The 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 (P)

We are now ready to formulate a solution procedure for (P). 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 (P):

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 $T(P) = T(R)$ is optimal for (P); stop.
3. If $T(R) > 1/x_n^*$, check whether the objective function of (R) in $1/x_n^*$ equals $v(R)$. If so, $T(P) = 1/x_n^*$ is optimal for (P); stop.
4. Otherwise, we have that $T(P) \geq 1/x_n^*$ and we first find a feasible solution for problem (P) by applying the FP or IFP. If the corresponding objective value is close enough to $v(R)$, then it is also close to $v(P)$, 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 $T(P)$.

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 (P). 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 (4.3) 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 (4.3); even as such the IFP outperforms the approach of Goyal and Kusy (1985), 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, the determination of a Lipschitz constant becomes more difficult, if possible at all. Therefore, we will not apply Lipschitz optimisation to determine a value of $v(P)$. 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 for (P) 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 $v(P)$ of (P) 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. All algorithms are implemented in Borland Pascal version 7.0 on a 66 MHz personal computer.

By considering the gap between $v(R)$ and $v(P)$ we are by Lemma 3.1 able to say something about the optimal objective value $v(P_c)$ of (P_c). We will not investigate problem (P_c) any further, since incorporation of the correction factor $\Delta(k)$ in a solution procedure is too time consuming.

For all models we have six different values for the number n of components and seven different values for the set-up cost S. This yields forty-two different combinations of n and S, and for each of these combinations hundred 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 5.1.

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 $v(P)$ 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 5.2 the relevant results of the 4200 problem instances for each model are given.

Notice first that from this table it follows that the difference between the relaxed solution $v(R)$ and the optimal objective value $v(P)$ of problem (P) is not very large. On average the gap is approximately one per cent or less and the maximum deviation is 5.566% for the model of Goyal and Kusy and

Table 5.1. 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)	
The following parameters are taken randomly:	
Special Case of Goyal and Kusy:	*Minimal-Repair Model:*
$f_i \in [15, 50]$	$\lambda_i \in [1, 20]$
$v_i \in [1, 20]$	$\beta_i \in [1.5, 4]$
$e \in [1, 4]$	$c_i^r \in [1, 250]$
Inspection Model:	*Block-Replacement Model:*
$\lambda_i \in [1, 20]$	$\lambda_i \in [1, 20]$
$\beta_i \in [1.5, 4]$	$\beta_i \in [1.5, 4]$
$c_i^u \in [c_i^p/\mu_i + 1, 1000]$	$c_i^f \in [2c_i^p/(1 - \sigma_i^2/\mu_i^2) + 1, 5000]$

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 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 (1995) 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 1995) 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.

even smaller for the other models. By Lemma 3.1 we have that the optimal objective value $v(P_c)$ of problem (P_c) will deviate even less from $v(R)$. This implies that if one wants to find a solution to problem (P_c), it is better to solve the easier problem (P) first. Since the gap between $v(P)$ and $v(R)$ is often small, this yields a solution that will in most cases suffice. Only when the gap is considered not small enough, one can subsequently apply a heuristic to problem (P_c) to try to find an objective value that is smaller than $v(P)$.

From the table it can be seen that solving the relaxation takes very little time. A subsequent application of the FP requires only one function evaluation for each component and this takes a negligible amount of time, which is why for the FP no running times are given in Table 5.2. Applying the IFP also takes little time. (All running times in Table 5.2 are higher for the inspection model 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 an objective value up to 0.01% smaller than that according to the Lipschitz-optimisation procedure.

As can be expected, the algorithm of Goyal and Kusy outperforms the algorithm of Goyal and Gunasekaran. This is explained from the fact that Goyal and Kusy take the *optimal* k_i given a value of T, whereas Goyal and

Table 5.2. Results of 4200 Random Examples for the Special Case of Goyal and Kusy, the Minimal-Repair Model and the Inspection Model

	GoyKus	MinRep	Inspec
Relaxation (R):			
Average running time relaxation (sec.)	0.01	0.01	0.06
Deviation (R) $(v(P) - v(R))/v(R)$:			
Average deviation (R)	1.174%	0.531%	0.835%
Minimum deviation (R)	0.000%	0.000%	0.000%
Maximum deviation (R)	5.566%	3.390%	4.953%
Feasibility Procedure (FP):			
Deviation FP $(v(FP) - v(P))/v(P)$:			
Average deviation FP	1.294%	0.246%	0.398%
Minimum deviation FP	0.000%	0.000%	0.000%
Maximum deviation FP	13.666%	8.405%	7.616%
Improved Feasibility Procedure (IFP):			
Average running time IFP (sec.)	0.07	0.05	1.32
Deviation IFP $(v(IFP) - v(P))/v(P)$:			
Average deviation IFP	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
Deviation GSS $(v(GSS) - v(P))/v(P)$:			
Average deviation GSS	0.001%	0.000%	0.000%
Minimum deviation GSS	0.000%	0.000%	-0.001%
Maximum deviation GSS	0.334%	0.152%	0.107%
Goyal and Kusy (GK):			
Average running time GK (sec.)	0.07	0.12	4.64
Deviation GK $(v(GK) - v(P))/v(P)$:			
Average deviation GK	0.829%	0.421%	1.253%
Minimum deviation GK	0.000%	-0.001%	0.000%
Maximum deviation GK	11.654%	18.289%	66.188%
Goyal and Gunasekaran (GG):			
Average running time GG (sec.)	0.06	0.13	4.05
Deviation GG $(v(GG) - v(P))/v(P)$:			
Average deviation GG	0.984%	0.608%	1.910%
Minimum deviation GG	0.000%	-0.001%	0.000%
Maximum deviation GG	14.027%	18.289%	66.188%

Gunasekaran take for each k_i the *rounded* optimal real value. However, the differences between the two algorithms are small.

The feasible solution corresponding with the relaxation (i.e., obtained by application of the FP) is in most cases better than that of the algorithms of Goyal et al. Only for the special case of Goyal and Kusy the FP performs somewhat worse. For the minimal-repair and inspection model the FP performs much better.

In all cases the IFP (that is an intelligent modification of the approach of Goyal et al.) outperforms the iterative algorithms of Goyal et al., 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 algorithms of Goyal et al.; 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 algorithms of Goyal et al. 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 per cent). However, there is a considerable difference in the number of times that large deviations were generated. This is illustrated in Table 5.3 that gives the percentage of the examples in which the IFP and the algorithm of Goyal and Kusy had a deviation 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 algorithms of Goyal et al. This also implies that the algorithms of Goyal et al. can be improved considerably if another initialisation of the k_i and T is taken, viz. according to the solution of the relaxation.

Table 5.3. 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

The deviation of 66.188% in Table 5.2 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 5.4. The large deviation for 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 one. The corresponding T is then determined; it equals 5.87. In the following iteration it is investigated for each component i

Table 5.4. 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 $v(P) = 706.29$

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

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 5.4. 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 one. 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 5.2 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). This is not much compared to, for example, the algorithms of Goyal et al.

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, especially not when it is considered that Lipschitz optimisation is an optimal solution procedure and when the running times are compared to those of the heuristics discussed here.

The running time of the Lipschitz optimisation depends on the number of components and on the set-up cost. In Table 5.5 the average running times are given for the hundred random examples that were taken for each of the forty-two combinations of n and S for the minimal-repair model. As can be

Table 5.5. 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

seen from this table, the running time increases somewhat more than linearly in the number n of components and decreases in the set-up cost S. The almost linear increment of speed is a nice result when it is considered that Lipschitz optimisation is an optimal solution procedure and that alternative optimal procedures published so far in the literature (see, for example, Goyal 1974 in the inventory context) involve only enumeration methods with exponentially growing running times. 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 $T(P)$ 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 with a deviation of less than one per cent on average. The improved feasibility procedure outperforms the algorithms of Goyal et al. 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 per cent 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 (P), we chose the number four here as well.

In Table 5.6 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 1990). The solutions of the FP, IFP and the algorithms of Goyal et al. are now compared with the values of $v(P)$ obtained by the multiple-interval golden-section search heuristic.

Table 5.6. Results of 4200 Random Examples for the Block-Replacement Model

Relaxation (R):	
Average running time relaxation (sec.)	0.23
Average deviation (R) $(v(P) - v(R))/v(R)$	0.402%
Minimum deviation (R)	0.000%
Maximum deviation (R)	2.708%
Feasibility Procedure (FP):	
Average deviation FP $(v(FP) - v(P))/v(P)$	0.196%
Minimum deviation FP	0.000%
Maximum deviation FP	12.217%
Improved Feasibility Procedure (IFP):	
Average running time IFP (sec.)	1.30
Average deviation IFP $(v(IFP) - v(P))/v(P)$	0.051%
Minimum deviation IFP	-0.002%
Maximum deviation IFP	5.921%
Golden-Section Search (GSS):	
Average running time GSS (sec.)	10.26
Goyal and Kusy (GK):	
Average running time GK (sec.)	3.72
Average deviation GK $(v(GK) - v(P))/v(P)$	0.658%
Minimum deviation GK	-0.222%
Maximum deviation GK	39.680%
Goyal and Gunasekaran (GG):	
Average running time GG (sec.)	3.54
Average deviation GG $(v(GG) - v(P))/v(P)$	0.943%
Minimum deviation GG	-0.222%
Maximum deviation GG	41.003%

From this table is follows again that the gap between $v(R)$ and $v(P)$ is small: maximally 2.637% and only 0.399% on average. This implies that also for the block-replacement model it is better to solve first problem (P) than problem (P_c), since the solution thus obtained will in many cases be sufficiently good. If the gap is not small enough, one can subsequently apply a heuristic to problem (P_c).

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.

Also in this case the algorithm of Goyal and Kusy outperforms the algorithm of Goyal and Gunasekaran, though the differences are small. The FP already outperforms the algorithms of Goyal et al. and the IFP performs even better. 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% (and for the algorithm of Goyal and Gunasekaran even larger). It can happen that the algorithms of Goyal et al. sometimes perform slightly better than the IFP, reflected in the minimum deviations of -0.222% for the algorithms of Goyal et al. and -0.002% for the IFP.

The golden-section search heuristic applied to solve problem (P) needed again four intervals in most cases. The average running time of the heuristic is 10.26 seconds, which is not much compared to, for example, the algorithms of Goyal et al. Remember that the solutions of the algorithms of Goyal et al. 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 algorithms of Goyal et al. 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 algorithms, regarding the average deviations of 0.658% and 0.943% for the algorithms of Goyal et al. and 0.051% for 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 algorithms of Goyal et al. (especially the maximum deviation is much smaller). When precision is more important, we can apply the golden-section search heuristic, at the cost of somewhat more time.

6. Conclusions

In this chapter 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 per cent above the minimal value. This approach outperforms the approach of Goyal et al. When precision is more important, a fast heuristic based on golden-section search can be applied to obtain a solution with a deviation of almost zero per cent. 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 chapter 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.

References

Avriel, M., Diewert, W.E., Schaible, S., Zang, I.: *Generalized Concavity*. New York: Plenum Press 1988

Bäckert, W., Rippin, D.W.T: The Determination of Maintenance Strategies for Plants Subject to Breakdown. *Computers and Chemical Engineering* **9**, 113–126 (1985)

Barros, A.I., Dekker, R., Frenk, J.B.G., van Weeren, S.: Optimizing a General Replacement Model by Fractional Programming Techniques. Technical Report. Econometric Institute, Erasmus University Rotterdam (1995)

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

Cho, D.I., Parlar, M.: A Survey of Maintenance Models for Multi-Unit Systems. *European Journal of Operational Research* **51**, 1–23 (1991)

Dagpunar, J.S.: Formulation of a Multi Item Single Supplier Inventory Problem. *Journal of the Operational Research Society* **33**, 285–286 (1982)

Dekker, R.: Integrating Optimisation, Priority Setting, Planning and Combining of Maintenance Activities. *European Journal of Operational Research* **82**, 225–240 (1995)

Dekker, R., Frenk, J.B.G., Wildeman, R.E. : An Efficient Optimal Solution Method for the Joint Replenishment Problem. *European Journal of Operational Research*. To appear (1996)

Dekker, R., Roelvink, I.F.K.: Marginal Cost Criteria for Preventive Replacement of a Group of Components. *European Journal of Operational Research* **84**, 467–480 (1995)
Goyal, S.K.: Determination of Economic Packaging Frequency for Items Jointly Replenished. *Management Science* **20**, 293–298 (1973)
Goyal, S.K.: Determination of Optimum Packaging Frequency of Items Jointly Replenished. *Management Science* **21**, 436–443 (1974)
Goyal, S.K.: A note on Formulation of the Multi-Item Single Supplier Inventory Problem. *Journal of the Operational Research Society* **33**, 287–288 (1982)
Goyal, S.K., Gunasekaran, A.: Determining Economic Maintenance Frequency of a Transport Fleet. *International Journal of Systems Science* **4**, 655–659 (1992)
Goyal, S.K., Kusy, M.I.: Determining Economic Maintenance Frequency for a Family of Machines. *Journal of the Operational Research Society* **36**, 1125–1128 (1985)
Goyal, S.K., Satir, A.T.: Joint Replenishment Inventory Control: Deterministic and Stochastic Models. *European Journal of Operational Research* **38**, 2–13 (1989)
Hiriart-Urruty, J.-B., Lemaréchal, C.: *Convex Analysis and Minimization Algorithms I: Fundamentals.* A Series of Comprehensive Studies in Mathematics, Vol. 305. Berlin: Springer 1993
Horst, R., Pardalos, P.M.: *Handbook of Global Optimization.* Dordrecht: Kluwer 1995
Howard, R.A.: *Dynamic Programming and Markov Processes.* New York: Wiley 1960.
Martos, B. *Nonlinear Programming Theory and Methods.* Budapest: Akademiai Kiado 1975
Smeitink, E., Dekker, R.: A Simple Approximation to the Renewal Function. *IEEE Transactions on Reliability* **39**, 71–75 (1990)
Van der Duyn Schouten, F.A., Vanneste, S.G.: Analysis and Computation of (n, N)-Strategies for Maintenance of a Two-Component System. *European Journal of Operational Research* **48**, 260–274 (1990)
Van der Duyn Schouten, F. A.: Stochastic Models of Reliability and Maintenance: An Overview. In this volume (1996), pp. 117–136
Van Egmond, R., Dekker, R., Wildeman, R.E.: Correspondence on: Determining economic maintenance frequency of a transport fleet. *International Journal of Systems Science* **26**, 1755–1757 (1995)

Appendix

A. Determination of Lipschitz Constant

We will prove here that the objective function of problem (P) 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 (3.1)), then the Lipschitz constant L for the objective function of (P) equals

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

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

To do so, consider an arbitrary $i \in \{1, \dots, n\}$ 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

$$\begin{aligned} L_i = \max\{L_i^{(k)}\}, & \\ \text{where } k \text{ ranges from } \max\{0, \lfloor T_l x_i^* \rfloor\} \text{ to } \lfloor T_u x_i^* \rfloor. \end{aligned} \tag{A.2}$$

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

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

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 (A.2).

If we now find an expression for the Lipschitz constant $L_i^{(k)}$, then with (A.1) and (A.2) we have an expression for the Lipschitz constant L. In the proof of Lemma 4.2 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 3.1 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 $t_1 \leq t_2 \in [1/x_n^*, 1/x_i^*]$:

$$\begin{aligned} |g_i(t_1) - g_i(t_2)| &= |\Phi_i(1/t_1) - \Phi_i(1/t_2)| \\ &\leq -\frac{d}{dt}\Phi_i(1/t)\bigg|_{t=t_1} \cdot |t_1 - t_2| \end{aligned}$$

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

so that

$$L_i^{(0)} = (x_n^*)^2 \Phi_i'(x_n^*). \tag{A.3}$$

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

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

and so

$$\begin{aligned} L_i^{(k)} = \max\Bigg\{ &\frac{k+1}{k^2}(x_i^*)^2 \Phi_i'\left(\frac{k+1}{k}x_i^*\right), \\ &-\frac{k}{(k+1)^2}(x_i^*)^2 \Phi_i'\left(\frac{k}{k+1}x_i^*\right)\Bigg\}. \end{aligned} \tag{A.4}$$

Notice that both arguments in (A.4) 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, (A.2) becomes

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

with $L_i^{(k)}$ given by (A.3) and (A.4).

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 (P) on the intervals $[T_1, T_u]$ and $[T_2, T_u]$ respectively, with $T_1 \le T_2 \le T_u$, then $L_1 \ge L_2$.