ELSEVIER

European Journal of Operational Research 99 (1997) 552-564

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

Optimal clustering of frequency-constrained maintenance jobs with shared set-ups

Gerhard van Dijkhuizen *, Aart van Harten

Faculty of Technology & Management, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

Abstract

Since maintenance jobs often require one or more set-up activities, joint execution or clustering of maintenance jobs is a powerful instrument to reduce shut-down costs. We consider a clustering problem for frequency-constrained maintenance jobs, i.e. maintenance jobs that must be carried out with a prescribed (or higher) frequency. For the clustering of maintenance jobs with identical, so-called common set-ups, several strong dominance rules are provided. These dominance rules are used in an efficient dynamic programming algorithm which solves the problem in polynomial time. For the clustering of maintenance jobs with partially identical, so-called shared set-ups, similar but less strong dominance rules are available. Nevertheless, a surprisingly well-performing greedy heuristic and a branch and bound procedure have been developed to solve this problem. For randomly generated test problems with 10 set-ups and 30 maintenance jobs, the heuristic was optimal in 47 out of 100 test problems, with an average deviation of 0.24% from the optimal solution. In addition, the branch and bound method found an optimal solution in only a few seconds computation time on average. © 1997 Elsevier Science B.V.

Keywords: Maintenance; Optimization; Dynamic programming; Heuristics; Branch and bound

1. Introduction

Many preventive maintenance jobs (inspections, replacements) of production systems require shut-down of the units involved. If these units are used continuously, as is the case in process industry, shut-downs can be very costly and management will try to minimize their duration and frequency. Since maintenance jobs often share one or more preparatory set-up activities, joint execution or clustering of maintenance jobs is generally seen as a powerful instrument to reduce shut-down costs.

The clustering of maintenance jobs can be modelled on a long-term and on a short-term basis. In the long term, maintenance jobs are combined into so-called maintenance packages that are executed at fixed intervals (see Fig. 1). In the short term, typical circumstances such as maintenance opportunities and manpower requirements are taken into account. Typically, short-term clustering is used in an operational planning phase, and long-term clustering is applied to strategical decision-making. In this paper, we focus on the long-term clustering possibilities.

In literature, much attention has been paid to the planning of preventive maintenance jobs, where correlation between various jobs is essential in view of set-up avoidance. For a literature review on mathemat-

* Corresponding author.
Email: g.c.vandijkhuizen@sms.utwente.nl.

PII S0377-2217(96)00320-7

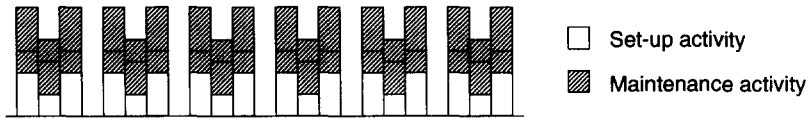

Fig. 1. Long-term clustering of maintenance activities.

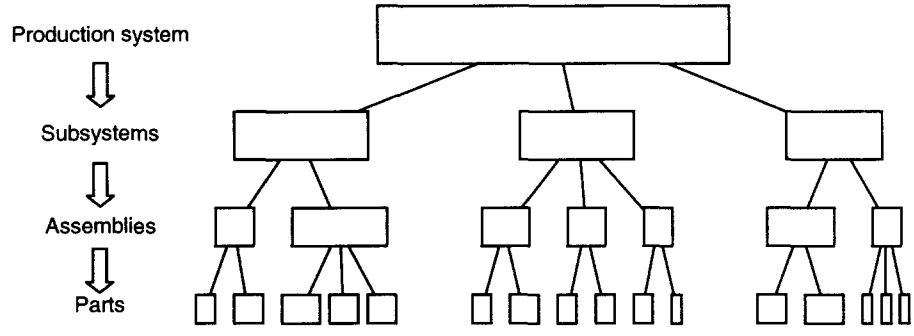

Fig. 2. Example of a production system tree consisting of four hierarchical levels.

ical models applied to maintenance, we refer to the surveys of Pierskalla and Voeller [8], Valdez-Flores and Veldman [10], and Cho and Parlar [2]. One of the problems encountered in practice, is that for large numbers of components, mathematical models are difficult to analyse (cf. Vanneste [12]). Besides, optimal policies are often much too complex to be implemented in decision support systems for maintenance planning. Therefore, a decomposition approach is to be preferred, in which the outcomes of mathematical models for individual components are used as inputs in a comprehensive model.

The clustering of frequency-constrained maintenance jobs, as presented in this paper, can be seen as such a comprehensive model, since the frequency of each maintenance job must be known in advance. Typically, the frequency of a maintenance job is determined with the use of mathematical models, such as the age or block replacement model (cf. Barlow and Proschan [1]), and the delay-time model (cf. Christer and Waller [3]). In many practical situations, however, the use of mathematical models is not possible owing to a lack of historical data. In these cases, frequencies are usually based on subjective data or expert opinions. Finally, the frequency of a maintenance job might also be based on safety restrictions or legislation. The use of limitative frequencies, or so-called frequency constraints, enables us to integrate these frequencies, based on either objective or subjective data, in one and the same mathematical model.

Pioneering work on this subject has been carried out by Gits [6,7], who considers the clustering of frequency-constrained maintenance jobs with identical, so-called *common* set-ups. Dekker, Wildeman and Smit [4,13] recently considered an approach in which the frequency constraints are replaced by frequency-dependent costs. Although they focus on the clustering of maintenance jobs in an operational planning phase, their methods are also applicable to long-term clustering problems.

In our opinion, the assumption of common set-ups cannot be sustained, as production systems become more and more complicated. Nevertheless, many production systems can be decomposed into several subsystems, which in turn can be decomposed into several assemblies, parts, components, and so on. This decomposition leads to a tree-like structure as represented in Fig. 2, which will be referred to as the production system tree. In general, with each node of the production system tree we can associate a certain set-up activity, and a number of maintenance jobs that could be performed if that particular set-up was carried out. Considering the tree-like structure of the production system, it is clear that some maintenance jobs may not share all setup-activities, as is the case with common set-ups, but only a subset of them. For this reason, we consider the clustering of frequency-constrained maintenance jobs with partially identical,

so-called *shared* set-ups.

It is clear that this provides a richer and more realistic modelling framework in comparison with the requirement of completely coinciding paths, as is the case with common set-ups. Practical examples of shared, but not common, set-ups can be found in various areas, for example in airline maintenance, maintenance of nuclear power plants, and off-shore maintenance. A real application of shared set-ups is presented in an article of Sculli and Suraweera [9], which deals with opportunistic tramcar maintenance.

In the applications above, the hierarchical structure of set-up activities and maintenance jobs is due to the tree-like structure of the production system. This is, however, not strictly necessary, as the following example shows. Consider a melting furnace which is subject to several periodic preventive maintenance jobs. Due to safety restrictions, different maintenance jobs require different furnace temperatures. The furnace has to be cooled down to the required temperature before any maintenance job can be carried out. If we associate a set-up activity with each of the required temperatures, then the set of different temperatures also reflects a shared set-up structure.

The outline of this paper is as follows. In Section 2, a mathematical formulation of the clustering problem is given and the complexity of this problem is briefly discussed. In Section 3, the clustering problem with common set-ups is considered. Several dominance rules are provided and an efficient dynamic programming algorithm is developed, which solves this problem in polynomial time. In Section 4, a greedy heuristic and a branch and bound algorithm are presented for the clustering problem with shared set-ups, for which similar but less strong dominance rules are available. Computational results in Section 5 show that the heuristic generates near-optimal solutions, and that the branch and bound algorithm finds an optimal solution within acceptable computation time. Finally, in Section 6, some conclusions are summarized and possibilities for further research are indicated.

2. General approach

In this section, a proper definition and a mathematical formulation of the clustering problem are given, and the complexity of this problem is briefly discussed.

2.1. Problem definition

Within this context, a maintenance job is defined as a preventive maintenance activity on a single component or a set of preventive maintenance activities on a set of components. Furthermore, a frequency-constrained maintenance job is defined as a maintenance job that must be carried out at fixed intervals with a prescribed or higher frequency. Finally, a maintenance package is defined as a set of maintenance jobs that are combined into a single maintenance job. Consequently, the frequency of a maintenance package must be at least as high as the frequency of each corresponding maintenance job.

It is assumed that fixed costs must be made for each set-up and for each maintenance job. These costs may consist of maintenance-related costs (e.g. salaries, spare parts, tools, materials) as well as production-related costs (e.g. production loss, productivity loss, delay penalties). Given a limitative frequency (frequency constraint) for each maintenance job, the clustering problem is concerned with the partitioning (clustering) of a set of maintenance jobs into subsets of maintenance packages (clusters), so that preventive maintenance costs per unit of time are minimized in the long run. Note that the reduction in corrective maintenance costs, as a positive side-effect of clustering, is not contained in our analysis. If so, clustering would become even more profitable.

In our analysis, we assume that the costs of a cluster can be computed from the costs of the individual set-up activities and maintenance jobs, and that the costs of a clustering can be computed from the costs of the individual clusters. In other words, we use an overall additive cost structure, as will be explicitly stated in the following section. From a practical point of view, this means that (i) parallel execution of maintenance jobs within a cluster, and (ii) simultaneous execution of clusters within a clustering (e.g. in an operational planning phase) are not allowed. Other assumptions would lead to other interesting versions of the clustering problem, but are left for future investigations.

As a starting point of our analysis, the production system tree and the set of frequency-constrained maintenance jobs are converted into a so-called maintenance tree. The root of this maintenance tree corresponds to the production system in operating condition; maintenance jobs are represented by the leafs and

set-up activities by the remaining nodes of the tree (see Fig. 3). Each set-up node in the maintenance tree can be identified with one of the nodes of the production system tree. Furthermore, each maintenance job is represented as a leaf of its parental set-up node. This parental node can be identified with the lowest-level node in the production system tree containing all components involved in the maintenance job. The communality of set-ups is determined by the joint part of the paths connecting the nodes to the root of the tree. These are the basic rules for the conversion; further details are skipped since they are not so relevant for what follows.

In our opinion, this transformation of a production system tree into a maintenance tree is justified in many real-life situations. There may be cases, however, where a maintenance tree does not capture all possible set-up properties (e.g. consider a somewhat artificial situation where some high-level maintenance jobs require shut-down of the production system, and other low-level maintenance jobs do not). If a proper transformation of the production system tree into a maintenance tree is not possible, we suggest the use of other methods. However, our approach can still be used as an approximation.

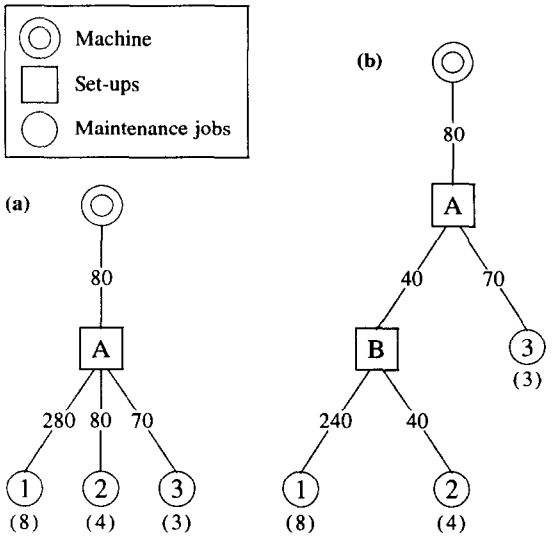

Fig. 3. Examples of a maintenance tree with (a) common set-ups and (b) shared set-ups. Set-up and maintenance costs are shown at the arcs, frequencies in brackets at the corresponding nodes.

2.2. Mathematical formulation

Consider a set of preparatory set-ups $I = \{1, \ldots, m\}$ and a set of frequency-constrained maintenance jobs $J = \{1, \ldots, n\}$. Let $I_j \subseteq I$ denote the collection of set-ups needed for maintenance job $j \in J$, and $s_i > 0$ the costs of set-up $i \in I$. Furthermore, let $f_j > 0$ denote the frequency, and $t_j > 0$ the costs of maintenance job $j \in J$.

A *cluster* of maintenance jobs is defined as a subset $U \subseteq J$. Similar to the definitions above, let $f(U) > 0$ denote the frequency, $s(U) > 0$ the set-up costs and $t(U) > 0$ the maintenance costs of a cluster $U \subseteq J$. Then the costs per unit time $\lambda(U)$ associated with U are defined as:

$$\lambda(U) = f(U)(s(U) + t(U)). \tag{1}$$

As mentioned before, the frequency $f(U)$ of a cluster of maintenance jobs $U \subseteq J$ must be at least as high as the frequency f_j of each maintenance job $j \in U$. From a cost-optimal point of view, we obtain:

$$f(U) = \max_{j \in U} f_j. \tag{2}$$

The set-up costs $s(U)$ of a cluster $U \subseteq J$ depend on the collection of set-ups needed for all maintenance jobs $j \in U$. Hence, $s(U)$ is given by:

$$s(U) = \sum_{i \in \bigcup_{j \in U} I_j} s_i. \tag{3}$$

Obviously, the maintenance costs $t(U)$ of a cluster $U \subseteq J$ are defined as:

$$t(U) = \sum_{j \in U} t_j. \tag{4}$$

A *clustering* of maintenance jobs is defined as a partitioning Ω of J. Since our objective is to minimize total costs per unit time, we are interested in the clustering Ω^* which minimizes:

$$\Lambda(\Omega) = \sum_{U \in \Omega} \lambda(U) = \sum_{U \in \Omega} f(U)(s(U) + t(U)). \tag{5}$$

Remark 1. Note that s_i $(i \in I)$ and t_j $(j \in J)$ might as well be defined as the expected values of some stochastic variables, since this would not affect the linearities in $\Lambda(\Omega)$.

Table 1
Values of $A(n)$ for increasing values of n

n	2	6	12	20	30
$A(n)$	2	203	$4.21 \cdot 10^6$	$5.17 \cdot 10^{13}$	$8.47 \cdot 10^{23}$

Let $A(n)$ denote the number of different clusterings of $\{1,\ldots,n\}$ and define $B(n,k)$ as the number of different clusterings of $\{1,\ldots,n\}$ into k clusters $(1 \leqslant k \leqslant n)$. Then $A(n) = \sum_k B(n,k)$, whereas $B(n,1) = B(n,n) = 1$ and $B(n,k) = kB(n-1,k) + B(n-1,k-1)$ for $1 < k < n$ (cf. Van Harten [11]). Some values of $A(n)$ for increasing values of n are presented in Table 1. Apparently, the search space of the clustering problem grows exponentially with the number of maintenance jobs n. In the following sections, it is shown that the complexity of the clustering problem can be reduced significantly, by exploiting its special structure.

3. The clustering problem with common set-ups

In this section, the clustering problem for maintenance jobs with common set-ups is considered. First, an example is given and several strong dominance rules are provided. With these dominance rules, an efficient dynamic programming algorithm is developed, which solves this problem in polynomial time. In the clustering problem with common set-ups, there is only one set-up $(m = 1)$. Consequently, $s(U) = S$ for all $U \subseteq J$, where $S > 0$ represents the common set-up costs.

3.1. Example

Consider the clustering problem with common set-ups, as shown in Fig. 3(a), with $I = \{A\}$, $J = \{1,2,3\}$, $I_1 = I_2 = I_3 = \{A\}$, $S = s_A = 80$, $(t_1\ t_2\ t_3) = (280\ 80\ 70)$, and $(f_1\ f_2\ f_3) = (8\ 4\ 3)$. The costs $\lambda(U)$ for all possible clusters U and the costs $\Lambda(\Omega)$ for all possible clusterings Ω are as given in Table 2. Apparently, the optimal clustering is determined by $\Omega^* = \{\{1\},\{2,3\}\}$, with corresponding costs $\Lambda(\Omega^*) = 3800$.

Table 2
Possible clusters U and clusterings Ω of J for the example of Fig. 3(a), with corresponding costs $\lambda(U)$ and $\Lambda(\Omega)$

U	$f(U)$	$s(U)$	$t(U)$	$\lambda(U)$
$\{1\}$	8	80	280	2880
$\{2\}$	4	80	80	640
$\{3\}$	3	80	70	450
$\{1,2\}$	8	80	360	3520
$\{1,3\}$	8	80	350	3440
$\{2,3\}$	4	80	150	920
$\{1,2,3\}$	8	80	430	4080

Ω	$\lambda(U) \mid U \in \Omega$	$\Lambda(\Omega)$
$\{\{1\},\{2\},\{3\}\}$	$\{2880,640,450\}$	3970
$\{\{1,2\},\{3\}\}$	$\{3520,450\}$	3970
$\{\{1,3\},\{2\}\}$	$\{3440,640\}$	4080
$\{\{1\},\{2,3\}\}$	$\{2880,920\}$	3800
$\{\{1,2,3\}\}$	$\{4080\}$	4080

3.2. Problem reduction

Let us now derive some dominance rules, with which optimal clusterings can be characterized.

Theorem 2. *Consider an optimal clustering Ω^* and let $Q_j \in \Omega^*$ denote the cluster corresponding to maintenance job $j \in J$. Then the following must be satisfied:*

(i) $\forall i,j \in J$: $f(Q_i) = f(Q_j) \rightarrow Q_i = Q_j$,
(ii) $\forall i,j \in J$: $f_i \geqslant f_j \rightarrow f(Q_i) \geqslant f(Q_j)$,
(iii) $\forall j \in J$: $f(Q_j) \leqslant f_j \cdot (S + t_j)/t_j$,
(iv) $\forall i,j \in J$: $f_i = f_j \rightarrow Q_i = Q_j$,
(v) $\forall i,j,k \in J$: $f_i \geqslant f_k \geqslant f_j \wedge Q_i = Q_j \rightarrow Q_i = Q_j = Q_k$.

Proof. If Ω^* violates (i), then $f(Q_i) = f(Q_j)$ and $Q_i \neq Q_j$ for some $i,j \in J$. In that case, combination of clusters Q_i and Q_j results in a clustering Ω' with $\Lambda(\Omega') = \Lambda(\Omega^*) - f(Q_i) \cdot S < \Lambda(\Omega^*)$. Hence, Ω^* is not optimal.

If Ω^* violates (ii), then $f_i \geqslant f_j$ and $f(Q_i) < f(Q_j)$ for some $i,j \in J$. In that case, moving job j from cluster Q_j to Q_i results in a clustering Ω' with $\Lambda(\Omega') \leqslant \Lambda(\Omega^*) - f(Q_j) \cdot t_j + f(Q_i) \cdot t_j < \Lambda(\Omega^*)$, since $f(Q_i) \geqslant f_i \geqslant f_j$. Hence, Ω^* cannot be optimal.

If Ω^* violates (iii), then $f(Q_j) \cdot t_j > f_j \cdot (S + t_j)$ for some $j \in J$. In that case, removing job j from cluster

Table 3
Values of $A(n,p)$ versus $A(n)$ for increasing values of n and p

n	2	6	12	20	30
p	2	3	4	5	6
$A(n,p)$	2	4	8	16	32
$A(n)$	2	203	$4,21 \cdot 10^6$	$5,17 \cdot 10^{13}$	$8,47 \cdot 10^{23}$

Q_j and creating a new cluster $\{j\}$ results in a clustering Ω' with $\Lambda(\Omega') \leqslant \Lambda(\Omega^*) - f(Q_j) \cdot t_j + f_j \cdot (S + t_j) < \Lambda(\Omega^*)$. Again, it follows that Ω^* is not optimal.

Properties (iv) and (v) follow directly from properties (i) and (ii). □

Now let $A(n,p)$ denote the number of clusterings of $\{1,\ldots,n\}$, which satisfy dominance rules (i) and (ii) of Theorem 2, given that there are p different frequencies. Then it can easily be seen that $A(n,p) = 2^{p-1}$ (cf. Van Harten [11]). Some values of $A(n,p)$ for increasing values of n and p are given in Table 3. Clearly, the complexity of the clustering problem with common set-ups is reduced significantly, even if property (iii) of Theorem 2 is left out of consideration.

3.3. A dynamic programming algorithm

Using dominance rule (iv) of Theorem 2, we can assume -without loss of generality- that $f_1 > \cdots > f_n$, since jobs with the same frequencies are always contained in the same cluster. Hence, jobs i and j with $f_i = f_j$ can as well be replaced by a single job $k = \{i\} \cup \{j\}$ with $f_k = f_i = f_j$ and $t_k = t_i + t_j$. Let $f^j_{\max}$ denote the maximal frequency of maintenance job $j \in J$, according to dominance rule (iii) of Theorem 2:

$$f^j_{\max} = f_j \cdot (S + t_j)/t_j. \tag{6}$$

Furthermore, let $F(k)$ denote the minimal costs for clustering jobs $1,\ldots,k$ $(1 \leqslant k \leqslant n)$. For notational convenience, define $F(0) = 0$. Using dominance rule (v) of Theorem 2, $F(k)$ can be determined by means of the following dynamic programming equation:

$$F(k) = \min_{1 \leqslant i \leqslant k : f_i \leqslant f^k_{\max}} \left\{ F(i-1) + f_i \cdot \left(S + \sum_{j=i}^{k} t_j \right) \right\}. \tag{7}$$

Note that the clustering problem may have alternative optimal solutions. Obviously, this dynamic programming algorithm requires $1/2\, n(n+1)$ computations in the worst case (i.e. $f^j_{\max} \geqslant f_1$ for all $j \in J$) and is an $O(n^2)$ algorithm. In other words, the clustering problem with common set-ups can be solved in polynomial time.

3.4. Example (continued)

Consider the example of Section 3.1 again, for which it can easily be verified that $f^1_{\max} = 10\frac{2}{7}$, $f^2_{\max} = 8$, and $f^3_{\max} = 6\frac{3}{7}$. With this in mind, the dynamic programming algorithm results in:

$$\begin{aligned}
F(0) &= 0,\\
F(1) &= F(0) + f_1(S + t_1)\\
&= 0 + 8 \cdot (80 + 280) = 2880,\\
F(2) &= \min\{F(0) + f_1(S + t_1 + t_2),\\
&\qquad F(1) + f_2(S + t_2)\}\\
&= \min\{0 + 8 \cdot (80 + 280 + 80),\\
&\qquad 2880 + 4 \cdot (80 + 80)\}\\
&= \min\{3520, 3520\} = 3520,\\
F(3) &= \min\{F(1) + f_2(S + t_2 + t_3),\\
&\qquad F(2) + f_3(S + t_3)\}\\
&= \min\{2880 + 4 \cdot (80 + 80 + 70),\\
&\qquad 3520 + 3 \cdot (80 + 70)\}\\
&= \min\{3800, 3970\} = 3800.
\end{aligned}$$

Hence, the optimal solution is $\Omega^* = \{\{1\},\{2,3\}\}$ with corresponding costs $\Lambda(\Omega^*) = 3800$, which was also found in Section 3.1.

4. The clustering problem with shared set-ups

In this section, the clustering problem for maintenance jobs with shared set-ups is considered. First of all, an example is given and several dominance rules are provided. These dominance rules are used in an efficient branch and bound algorithm, which is developed next. Finally, this branch and bound algorithm is illustrated by an example.

Table 4
Possible clusters U and clusterings Ω of J for the example of Fig. 3(b), with corresponding costs $\lambda(U)$ and $\Lambda(\Omega)$

U	$f(U)$	$s(U)$	$t(U)$	$\lambda(U)$
{1}	8	120	240	2880
{2}	4	120	40	640
{3}	3	80	70	450
{1,2}	8	120	280	3200
{1,3}	8	120	310	3440
{2,3}	4	120	110	920
{1,2,3}	8	120	350	3760

Ω	$\lambda(U) \mid U \in \Omega$	$\Lambda(\Omega)$
{{1},{2},{3}}	{2880,640,450}	3970
{{1,2},{3}}	{3200,450}	3650
{{1,3},{2}}	{3440,640}	4080
{{1},{2,3}}	{2880,920}	3800
{{1,2,3}}	{3760}	3760

4.1. Example

Consider the clustering problem with shared set-ups, as shown in Fig. 3(b), with $I = \{A, B\}$, $J = \{1, 2, 3\}$, $(s_A\ s_B) = (80\ 40)$, $I_1 = I_2 = \{A, B\}$, $I_3 = \{A\}$, $(t_1\ t_2\ t_3) = (240\ 40\ 70)$, and $(f_1\ f_2\ f_3) = (8\ 4\ 3)$. The costs $\lambda(U)$ for all possible clusters U and the costs $\Lambda(\Omega)$ for all possible clusterings Ω are given in Table 4. Apparently, the optimal clustering is determined by $\Omega^* = \{\{1,2\},\{3\}\}$, with corresponding costs $\Lambda(\Omega^*) = 3650$.

4.2. Problem reduction

As in Section 3.2, let us derive some dominance rules, with which optimal clusterings can be characterized. First of all, let s_{jk} denote the shared set-up costs of jobs $j, k \in J$:

$$s_{jk} = \sum_{i \in I_j \cap I_k} s_i. \tag{8}$$

First of all, note that $s_{jk} = 0$ for some $j, k \in J$ implies that J can be partitioned into non-empty subsets J_1 and J_2 (i.e. $J_1 \cap J_2 = \emptyset$, $J_1 \cup J_2 = J$), such that $s_{jk} = 0$ for all $j \in J_1$ and $k \in J_2$. Hence, J_1 and J_2 can be treated separately and -without loss of generality- we can assume that $s_{jk} > 0$ for all $j, k \in J$. With this assumption in mind, Theorem 2 can be generalized as follows. But first, we need the following lemma.

Lemma 3. *Consider a clustering Ω and let $Q_j \in \Omega$ denote the cluster corresponding to maintenance job $j \in J$. Furthermore, for all $j, k \in J$, let θ_j and δ_{jk} be defined as:*

$$\theta_j = s_{jj} + t_j - \max_{i \neq j} s_{ij}, \tag{9}$$

$$\delta_{jk} = \frac{s_{jj} + t_j - s_{jk}}{\theta_j}. \tag{10}$$

Then: $\delta_{jk}^{-1} \cdot f(Q_j) > f(Q_k) \geqslant f_j$ for some $j, k \in J \rightarrow \Omega$ is not optimal.

Proof. Suppose that $f(Q_k) \geqslant f_j$ for some $j, k \in J$. Then removing job j from cluster Q_j results in a cost decrement of at least $\Delta^- = f(Q_j) \cdot (s_{jj} + t_j - \max_{i \neq j} s_{ij}) = f(Q_j) \cdot \theta_j$. Similarly, moving job j to cluster Q_k results in a cost increment of at most $\Delta^+ = f(Q_k)(s_{jj} + t_j - s_{jk})$, since $f(Q_k) \geqslant f_j$ and $k \in Q_k$ by assumption. Since $\delta_{jk}^{-1} \cdot f(Q_j) > f(Q_k)$ is equivalent to $\Delta^- > \Delta^+$, this completes the proof. □

Lemma 3 should be interpreted as follows. Firstly, $\theta_j = s_{jj} + t_j - \max_{i \neq j} s_{ij}$ represents the minimal decrease in $t(U)$ if job j is removed from an arbitrary cluster $U \subseteq J$ with $j \in U$. Secondly, $s_{jj} + t_j - s_{jk}$ represents the maximal increase in $t(U)$ if job j is added to an arbitrary cluster $U \subseteq J$ with $j \notin U$ and $k \in U$. Hence, δ_{jk} reflects the ratio of these cost components. Clearly, $\delta_{jk} \geqslant 1$ for all $j, k \in J$. Furthermore, $\delta_{jk} = 1$ if $s_{jj} = s_{jk} = s_{kk}$ for some $j, k \in J$, i.e. if jobs j and k require identical (common) set-ups. In general terms, Lemma 3 provides conditions for the cluster frequencies $\{f(Q_j), f(Q_k)\}$ of each pair $\{j, k\}$ of maintenance jobs.

Theorem 4. *Consider an optimal clustering Ω^* and let $Q_j \in \Omega^*$ denote the cluster corresponding to maintenance job $j \in J$. Then the following must be satisfied:*

(i) $\forall i, j \in J$: $f(Q_i) = f(Q_j) \rightarrow Q_i = Q_j$,
(ii) $\forall i, j \in J$: $f_i \geqslant f_j \rightarrow f(Q_i) \geqslant \delta_{ji}^{-1} \cdot f(Q_j)$,
(iii) $\forall j \in J$: $f(Q_j) \leqslant f_j \cdot (s_{jj} + t_j)/\theta_j$,
(iv) $\forall i, j \in J$: $f_i = f_j \rightarrow \delta_{ij}^{-1} \leqslant \dfrac{f(Q_j)}{f(Q_i)} \leqslant \delta_{ji}$,

(v) $\forall i,j,k \in J$: $f_i \geqslant f_k \geqslant f_j \wedge Q_i = Q_j$
$$\rightarrow \delta_{jk}^{-1} \leqslant \frac{f(Q_k)}{f(Q_i)} = \frac{f(Q_k)}{f(Q_j)} \leqslant \min\{\delta_{ki}, \delta_{kj}\}.$$

Proof. If Ω^* violates (i), then $f(Q_i) = f(Q_j)$ and $Q_i \neq Q_j$ for some $i,j \in J$. In that case, combination of clusters Q_i and Q_j results in a clustering Ω' with $\Lambda(\Omega') < \Lambda(\Omega^*)$, since $s_{jk} > 0$ for all $j,k \in J$ by assumption. Hence, Ω^* is not optimal.

If Ω^* violates (ii), then $f_i \geqslant f_j$ and $\delta_{ji}^{-1} f(Q_j) > f(Q_i)$ for some $i,j \in J$. Since $f(Q_i) \geqslant f_i \geqslant f_j$, this yields $\delta_{ji}^{-1} f(Q_j) > f(Q_i) \geqslant f_j$, and it follows from Lemma 3 that Ω^* is not optimal.

If Ω^* violates (iii), then $f(Q_j) \cdot \theta_j > f_j \cdot (s_{jj} + t_j)$ for some $j \in J$. In that case, removing job j from cluster Q_j and creating a new cluster $\{j\}$ results in a clustering Ω' with $\Lambda(\Omega') \leqslant \Lambda(\Omega^*) - f(Q_j) \cdot \theta_j + f_j \cdot (s_{jj} + t_j) < \Lambda(\Omega^*)$. Again, it follows that Ω^* is not optimal.

Properties (iv) and (v) follow directly from properties (i) and (ii). □

Remark 5. In the case of common set-ups, $\theta_j = t_j$ for all $j \in J$ and $\delta_{jk} = 1$ for all $j,k \in J$. Using dominance rule (i), which coincides for Theorem 2 and Theorem 4, it can easily be verified that Theorem 4 is indeed a generalization of Theorem 2.

4.3. A branch and bound algorithm

In this section, a branch and bound algorithm (cf. Winston [14]) is presented, which uses the dominance rules provided by Theorem 4. The branch and bound tree is designed in such a way, that a node at depth m ($0 \leqslant m \leqslant n$) contains a partial clustering Ω_m of jobs $\{1,\ldots,m\}$. Consequently, a branch from a node (m, Ω_m) corresponds to the assignment of job $m+1$ to an already existing cluster $U \in \Omega_m$, or the creation of a new cluster $\{m+1\}$. Without loss of generality, we can assume that $f_1 \geqslant \cdots \geqslant f_n$, as a result of which the frequency of a cluster remains unchanged after it has been created. The branch and bound procedure can be described by the following steps:

- Use an heuristic method to determine an initial solution Ω^* and start with an empty clustering $(0, \emptyset)$ in the root of the branch and bound tree.
- Select one of the open nodes (m, Ω_m) according to a certain selection rule. If the clustering Ω_m contained in this node is final ($m = n$), update the best clustering found so far (i.e. $\Omega^* \leftarrow \Omega_n$) if necessary (i.e $\Lambda(\Omega_n) < \Lambda(\Omega^*)$). Otherwise, if this node is not in conflict with any dominance rules, calculate a lower bound $LB(m, \Omega_m)$ for the best clustering that can still be obtained, given the partial clustering Ω_m contained in the selected node. If this lower bound exceeds the best clustering found so far (i.e. $LB(m, \Omega_m) \geqslant \Lambda(\Omega^*)$), close the selected node. Otherwise, create a new node for each extension of the partial clustering Ω_m contained in the selected node. Repeat this procedure until no open nodes are left.
- The final clustering Ω^* is optimal.

4.3.1. Initial solution

The heuristic starts with each job in a separate cluster and proceeds with greedy improvements. In each iteration of the heuristic, the two clusters U^* and V^* whose union leads to the highest decrease in $\Lambda(\cdot)$ are combined into a new cluster $U^* \cup V^*$. This procedure is repeated until no improvements can be obtained anymore.

4.3.2. Selection rule

We implemented a combination of the well-known depth-first and best-first selection rules (cf. Winston [14]). In terms of the branch and bound tree, the node with the best (= lowest) lower bound at the deepest level of the branch and bound tree is selected in each iteration of the branch and bound procedure.

4.3.3. Dominance rules

In addition to the dominance rules of Theorem 4, two other dominance rules were used in the branch and bound algorithm. Before presenting these dominance rules for partial clusterings, let us first consider another two dominance rules for final clusterings, which they are based upon.

Theorem 6. *Consider an optimal clustering Ω^* and let $Q_j \in \Omega^*$ denote the cluster corresponding to maintenance job $j \in J$. Furthermore, for all $j \in J$ and $U \subseteq J$, let $\theta_j(U) = t(U \cup \{j\}) - t(U \setminus \{j\})$ be defined as:*

$$\theta_j(U) = s_{jj} + t_j - \max_{i \in U \setminus \{j\}} s_{ij}. \tag{11}$$

Then the following must be satisfied:

(i) $\forall j \in J$: $f(Q_j) \cdot \theta_j(Q_j) \leqslant f_j \cdot (s_{jj} + t_j)$,
(ii) $\forall j \in J$: $f(Q_j) \cdot \theta_j(Q_j) \leqslant \min_{U \in \Omega^*:\ f(U) \geqslant f_j} f(U) \cdot \theta_j(U)$.

Proof. If Ω^* violates (i), then $f(Q_j) \cdot \theta_j(Q_j) > f_j \cdot (s_{jj} + t_j)$ for some $j \in J$. Similar to the proof of property (iii) of Theorem 4, removing job j from cluster Q_j and creating a new cluster $\{j\}$ results in a clustering Ω' with $\Lambda(\Omega') < \Lambda(\Omega^*)$. Hence, Ω^* is not optimal.

If Ω^* violates (ii), then $f(Q_j) \cdot \theta_j(Q_j) > f(U) \cdot \theta_j(U)$ and $f(U) \geqslant f_j$ for some $j \in J$ and $U \in \Omega^*$. Similar to the proof of Lemma 3, moving job j from cluster Q_j to U results in a clustering Ω' with $\Lambda(\Omega') < \Lambda(\Omega^*)$. Hence, Ω^* is not optimal. □

Unfortunately, the dominance rules of Theorem 6 are only applicable to final clusterings and thus, cannot be used in the branch and bound algorithm. Hence, a generalization of Theorem 6 for partial clusterings is desirable. Similar to the definition in Section 4.2, let $f^j_{\max}$ denote the maximum frequency of maintenance job $j \in J$, according to dominance rule (iii) of Theorem 4:

$$f^j_{\max} = f_j \cdot (s_{jj} + t_j)/\theta_j. \tag{12}$$

Let us now consider a partial clustering Ω_m of jobs $\{1, \ldots, m\}$, and let $Q_{m,j} \in \Omega_m$ denote the cluster corresponding to maintenance job $j \in \{1, \ldots, m\}$. Furthermore, let $\Delta Q_{m,j} = \{k \in \{m+1, \ldots, n\} \mid f^k_{\max} \geqslant f(Q_{m,j})\}$ denote the collection of maintenance jobs $k \in \{m+1, \ldots, n\}$ that can still be included in cluster $Q_{m,j}$ without violation of dominance rule (iii) of Theorem 4. Then, with Theorem 6 in mind, a node (m, Ω_m) can be skipped if for some maintenance job $j \in \{1, \ldots, m\}$:

$$f(Q_{m,j}) > \min \left\{ \frac{f_j \cdot (s_{jj} + t_j)}{\theta_j(Q_{m,j} \cup \Delta Q_{m,j})}, \min_{U \in \Omega_m:\ f(U) \geqslant f_j} \frac{f(U) \cdot \theta_j(U)}{\theta_j(Q_{m,j} \cup \Delta Q_{m,j})} \right. . \tag{13}$$

From now on, these dominance rules will be referred to as the global dominance rules, since they are evaluated for all jobs $j \in \{1, \ldots, m\}$ in a given node (m, Ω_m). During the implementation of the branch and bound algorithm, it was found that these global dominance rules required a relatively large amount of computation time. Therefore, we decided to introduce so-called local dominance rules, which only evaluate for job m in a given node (m, Ω_m). In Section 5, it is investigated which dominance rules are to be preferred.

4.3.4. Lower bounds

We developed and implemented two lower bounds, which are based on the same general idea. Given a node (m, Ω_m) with a partial clustering Ω_m of jobs $\{1, \ldots, m\}$, a lower bound $\Delta_j(m, \Omega_m)$ is calculated for the increase in $\Lambda(.)$ due to each of the remaining jobs j $(m < j \leqslant n)$, given that the clustering Ω_m of jobs $\{1, \ldots, m\}$ is known, but the clustering of jobs $\{m+1, \ldots, j-1\}$ is not. A lower bound $LB(m, \Omega_m)$ for a clustering of jobs $\{1, \ldots, n\}$, given the partial clustering Ω_m of jobs $\{1, \ldots, m\}$, is then constructed as:

$$LB(m, \Omega_m) = \Lambda(\Omega_m) + \sum_{j=m+1}^{n} \Delta_j(m, \Omega_m). \tag{14}$$

In this expression, $\Delta_j(m, \Omega_m)$ gives a lower bound for each job j $(m < j \leqslant n)$ separately. In the following theorem, a so-called static lower bound is presented. This lower bound is called static, because its value does not depend on Ω_m and consequently, can be calculated in advance of the branch and bound procedure.

Theorem 7. *A static lower bound $\Delta_j(m, \Omega_m)$ is given by:*

$$\Delta_j(m, \Omega_m) = \min \left\{ f_j \cdot (s_{jj} + t_j), \min_{i \in \{1, \ldots, j-1\}:\ f_i \leqslant f^j_{\max}} f_i \cdot (s_{jj} + t_j - \max_{k \in \{i, \ldots, j-1\}} s_{jk}) \right. . \tag{15}$$

Proof. Consider an arbitrary clustering of jobs $\{1,\ldots,j-1\}$, for which no further information is available. Let us now deal with job j, for which two decisions can be made:

- Create a new cluster $\{j\}$. Then total costs increase with exactly $f_j \cdot (s_{jj} + t_j)$.
- Add job j to an already existing cluster $U \subseteq \{1,\ldots,j-1\}$. Then, according to dominance rule (iii) of Theorem 4, we can restrict ourselves to clusters of the form $\{i,\ldots,j-1\}$ with $i < j$ and $f_i \leqslant f^j_{\max}$, in which case total costs increase with $f_i \cdot (s_{jj} + t_j - \max\{s_{jk} \mid k \in \{i,\ldots,j-1\}\})$.

Given these possible decisions, Theorem 7 follows easily. □

The following theorem comprises a so-called dynamic lower bound. This lower bound is called dynamic, because its value depends on Ω_m and consequently, must be calculated during the branch and bound procedure.

Theorem 8. *A dynamic lower bound $\Delta_j(m, \Omega_m)$ is given by:*

$$\Delta_j(m,\Omega_m) = \min\left\{\begin{array}{l} f_j \cdot (s_{jj}+t_j), \\ \min\limits_{U\in\Omega_m:\ f(U)\leqslant f^j_{\max}} f(U)\cdot(s_{jj}+t_j - \max\limits_{k\in U\cup\{m+1,\ldots,j-1\}} s_{jk}), \\ \min\limits_{i\in\{m+1,\ldots,j-1\}:\ f_i\leqslant f^j_{\max}} f_i\cdot(s_{jj}+t_j - \max\limits_{k\in\{i,\ldots,j-1\}} s_{jk}). \end{array}\right. \quad (16)$$

Proof. In contrast with the proof of Theorem 7, consider an arbitrary clustering of jobs $\{1,\ldots,j-1\}$, for which the partial clustering Ω_m of jobs $\{1,\ldots,m\}$ is known, but for which no information is available on the partial clustering of jobs $\{m+1,\ldots,j-1\}$. With respect to job j, three decisions can be made:

- Create a new cluster $\{j\}$. Then total costs increase with exactly $f_j \cdot (s_{jj} + t_j)$.
- Add job j to an already existing cluster $V \subseteq \{1,\ldots,j-1\}$ with $U \subseteq V$ for some $U \in \Omega_m$. Then, according to dominance rule (iii) of Theorem 4, we can restrict ourselves to clusters of the form $U \cup \{m+1,\ldots,j-1\}$ with $U \in \Omega_m$ and $f(U) \leqslant f^j_{\max}$, in which case total costs increase with $f(U) \cdot (s_{jj} + t_j - \max\{s_{jk} \mid k \in U \cup \{m+1,\ldots,j-1\}\})$.
- Add job j to an already existing cluster $V \subseteq \{m+1,\ldots,j-1\}$. Then, for the same reasons as in (ii), we can restrict ourselves to clusters of the form $\{i,\ldots,j-1\}$ with $i \in \{m+1,\ldots,j-1\}$ and $f_i \leqslant f^j_{\max}$, in which case total costs increase with $f_i \cdot (s_{jj} + t_j - \max\{s_{jk} \mid k \in \{i,\ldots,j-1\}\})$.

Given this set of possible decisions, Theorem 8 follows easily. □

It is to be expected that the static lower bound requires more nodes but less time per node than the dynamic lower bound. In Section 5, it is investigated which lower bound is to be preferred.

4.4. Example (*continued*)

Consider the example of Section 4.1 again, for which it can easily be verified that $f^1_{\max} = 12$, $f^2_{\max} = 16$ and $f^3_{\max} = 6^3/_7$. The branch and bound algorithm, in which we used the dominance rules of Theorem 4 and the dynamic lower bound, results in:

- The heuristic starts with $\Omega^* = \{\{1\},\{2\},\{3\}\}$ and $\Lambda(\Omega^*) = 3970$. With $\Lambda(\{\{1,2\},\{3\}\}) = 3650$, $\Lambda(\{\{1,3\},\{2\}\}) = 4080$ and $\Lambda(\{\{2,3\},\{1\}\}) = 3800$, the heuristic continues with $\Omega^* = \{\{1,2\},\{3\}\}$ in the next iteration. With $\Lambda(\{\{1,2,3\}\}) = 3760$, the heuristic is terminated with $\Omega^* = \{\{1,2\},\{3\}\}$ and $\Lambda(\Omega^*) = 3650$.
- The branch and bound procedure starts with a node $A = (0,\emptyset)$ with lower bound $LB(A) = 3480$ in the root of the branch and bound tree. Since $LB(A) < \Lambda(\Omega^*)$, a new node $B = (1,\{\{1\}\})$ with $LB(B) = 3480$ is created. Since $LB(B) < \Lambda(\Omega^*)$, new nodes $C = (2,\{\{1,2\}\})$ with $LB(C) = 3800$ and $D = (2,\{\{1\},\{2\}\})$ with $LB(D) = 3650$ are created. Since $LB(C) > LB(D) = \Lambda(\Omega^*)$, the remaining nodes C and D are closed.
- The optimal solution is given by $\Omega^* = \{\{1,2\},\{3\}\}$ with $\Lambda(\Omega^*) = 3650$, which was also found in Section 4.1.

Remark 9. It can easily be shown that the heuristic always finds an optimal solution in problems with three or less maintenance jobs.

Table 5
Performance of the heuristic for 100 randomly generated test problems with 10 set-ups and $1 \leqslant p \leqslant 6$ maintenance jobs per set-up

	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$
% optimal	92 %	80 %	46 %	38 %	21 %	12 %
% deviation	0.04 %	0.17 %	0.36 %	0.39 %	0.54 %	0.57 %

Table 6
Average computation time (in seconds) and total number of nodes needed by the branch and bound algorithm for 100 randomly generated test problems with 10 set-ups and $1 \leqslant p \leqslant 3$ maintenance jobs per set-up: performance of local vs. global dominance rules, and static vs. dynamic lower bound

		$p = 1$		$p = 2$		$p = 3$	
dominance rules	lower bound	time	nodes	time	nodes	time	nodes
local	static	0.01	13.64	0.18	152.0	1.88	1424
local	dynamic	0.01	9.47	0.19	81.6	2.45	639
global	static	0.02	13.33	0.41	136.6	6.34	1069
global	dynamic	0.01	9.38	0.31	78.7	4.32	527

5. Computational results

We implemented the heuristic and the branch and bound procedure in Borland Pascal 7.0 on a personal computer with a 80486 microprocessor, operating at a clock speed of 40 MHz. To determine which dominance rules and which lower bounds are to be preferred, we tested our algorithms on a number of artificial maintenance trees with 10 set-ups ($m = 10$) and $1 \leqslant p \leqslant 6$ jobs per set-up ($n = 10, 20, 30, 40, 50, 60$) respectively, as represented in Fig. 4. For each tree structure, we randomly generated 100 test problems.

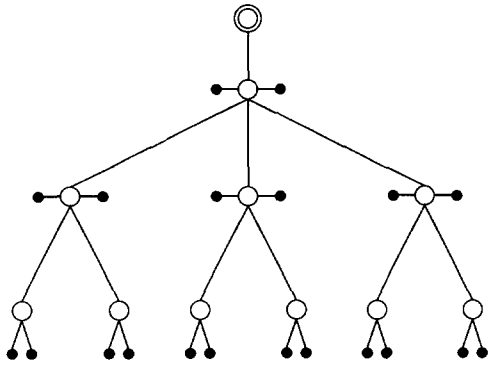

Fig. 4. Maintenance tree for test problems ($p = 2$).

In each test problem, values for s_i ($i \in I$) and t_j ($j \in J$) were drawn from a uniform distribution on $\{1,\ldots,100\}$, and values for f_j ($j \in J$) from a uniform distribution on $\{1,\ldots,12\}$.

For the initial solutions found by the heuristic, we measured which percentage was already optimal. Furthermore, we calculated the average deviation from the optimal solution. For the branch and bound algorithm, we focussed on the total number of nodes that was branched upon and the amount of CPU time needed. The results are presented in Tables 5, 6 and 7.

Table 5 shows that, somewhat surprisingly, the greedy heuristic performs very well. For problems with 10 maintenance jobs, the heuristic found an optimal solution in 92 out of 100 test problems, with an average deviation of 0.04% from the optimal solution. As was expected, an increase in the number of maintenance jobs reduced the performance of the heuristic: for problems with 60 maintenance jobs, the heuristic found an optimal solution in 12 out of 100 test problems, with a deviation of 0.57% from the optimal solution on average. Apparently, the performance of our heuristic reduces with the size of the problem, which is quite natural.

Table 6 indicates that the local dominance rules and the static lower bound give the best overall per-

Table 7
Average computation time (in minutes) and number of nodes needed by the branch and bound algorithm for 100 randomly generated test problems with 10 set-ups and $4 \leqslant p \leqslant 6$ maintenance jobs per set-up

		$p = 4$		$p = 5$		$p = 6$	
dominance rules	lower bound	time	nodes	time	nodes	time	nodes
local	static	0:20	$1.26 \cdot 10^4$	2:43	$1.57 \cdot 10^5$	26:40	$1.01 \cdot 10^6$

formance, at least for the test problems we considered. We have no indications, however, that experiments with other test problems would lead to different conclusions. Although the dynamic lower bound and the global dominance rules strongly reduced the total number of nodes (both with about a factor 2), the branch and bound procedure still requires higher computation times. Apparently, the reduction in nodes is canceled out by an increase in the required computation time per node.

From the results in Table 7, in which the static lower bound and the local dominance rules were used, it can be concluded that the computation time of the branch and bound algorithm is exponential in the number of maintenance jobs n. While formulating the same test problems as Integer Linear Programs, and solving them with a standard ILP solver, different results were found. Although computation times for small values of n were significantly larger (e.g. 0.2 versus 35 seconds for $n = 20$), the computation times of the ILP solver seemed to increase quadratically with the number of maintenance jobs n, indicating that the clustering problem might not be NP hard. Although this phenomenon may be caused by e.g. the limited choice of frequencies, a standard ILP solver will probably outperform our branch and bound procedure for large values of n.

On the other hand, our branch and bound procedure can still be improved significantly. For example, it is possible to decompose the clustering problem into a number of smaller subproblems, corresponding to second-level subtrees of the maintenance tree, by conditioning on the $2^{|F|-1}$ subsets of the set F of frequencies. It is to be expected that this decomposition will lead to significant reductions in computation times. For large problems, e.g. with 50 or more maintenance jobs, we suggest to use the heuristic, which finds near-optimal solutions in negligible computation times.

6. Concluding remarks

In this paper, we showed that the clustering problem for frequency-constrained maintenance jobs with common set-ups can be solved in polynomial time by means of an efficient dynamic programming algorithm. Furthermore, we developed a greedy heuristic and a branch and bound algorithm to determine an optimal clustering of frequency-constrained maintenance jobs with shared set-ups. We provided several dominance rules, inspired by the case of common set-ups, which strongly reduced the complexity of the clustering problem. Computational results indicated that the heuristic generates near-optimal solutions, and that the branch and bound algorithm requires acceptable computation times for many practical situations ($n \leqslant 50$).

Summarizing, we believe that our methods can be applied in many practical situations, and are a step forward into effective and efficient maintenance planning. Although minor improvements might still be obtained in an operational planning phase (e.g. by means of opportunistic maintenance policies), the determination of maintenance packages can as least serve as a basis for further optimization models. Our methods can be incorporated in an interactive decision support system with which strategical maintenance planning can be supported. For this reason, a number of generalizations of our methods are still under consideration.

For instance, the positive effect of clustering on the need for corrective maintenance was not taken into account. With the clustering of maintenance jobs, preventive maintenance is carried out more frequently, and consequently, corrective maintenance jobs will be reduced. It is possible to introduce frequency-dependent costs instead of frequency constraints, so that clustering becomes even more profitable in com-

parison with our approach. Unfortunately, frequency-dependent costs are hard to obtain in practice owing to a lack of historical data; this in contrast to the frequency constraints that are required for our methods.

Another possibility is to allow parallel execution of maintenance jobs and simultaneous execution of maintenance packages. In our model, it was assumed that no cost reductions can be obtained by carrying out maintenance jobs in parallel, or maintenance packages simultaneously. This lead to an overall additive cost structure. Other assumptions will lead to other interesting versions of the clustering problem.

It might also be worthwhile to develop more advanced heuristics, and to investigate their performance. For instance, it is possible to start with a bottom-up approach, i.e. first combining at the lowest-level and then working upwards. This might especially be worthwhile if the set of frequencies is limited. Finally, it is also possible to generalize the dominance rules for identical jobs (i.e. jobs with identical frequencies and identical set-ups) to dominance rules for identical subtrees. Since identical subtrees correspond to identical equipment at the same level of the production system tree, this might be an interesting generalization in many real-life applications. In future work, these suggestions will be studied more specifically.

References

[1] Barlow, R.E., and Proschan, F., *Mathematical Theory of Reliability*, Wiley, New York, 1965.

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

[3] Christer, A.H., and Waller, W.M., Delay time models for industrial maintenance problems, *Journal of the Operational Research Society* 35 (1984) 401–406.

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

[5] Gertsbakh, I.B., *Statistical Reliability Theory*, Marcel Dekker, New York, 1989.

[6] Gits, C.W., On the maintenance concept for a technical system: a framework for design, Ph.D. Dissertation, University of Eindhoven, The Netherlands, 1984.

[7] Gits, C.W., On the maintenance concept for a technical system: III. Design framework, *Maintenance Management International* 6 (1986) 223–237.

[8] Pierskalla, W.P., and Voelker, J.A., A survey of maintenance models: the control and surveillance of deteriorating systems, *Naval Research Logistics Quarterly* 23 (1976) 353–388.

[9] Sculli, D., and Suraweera, A.W., Tramcar maintenance, *Journal of the Operational Research Society* 30 (1979) 809–814.

[10] Valdez-Flores, C., Feldman, R.M., A survey of preventive maintenance models for stochastically deteriorating single-unit systems, *Naval Research Logistics* 36 (1989) 419–446.

[11] Van Harten, A., On the sequencing and clustering of maintenance jobs, Internal Report, University of Twente, The Netherlands, 1994.

[12] Vanneste, S.G., Maintenance policies for complex systems, Ph.D. Dissertation, University of Tilburg, The Netherlands, 1992.

[13] Wildeman, R., Dekker, R., and Smit, A.C.J.M., Combining maintenance activities in an operational planning phase: a dynamic programming approach, Technical Report 9424/A, Econometric Institute, Erasmus University Rotterdam, The Netherlands.

[14] Winston, W.L., *Operations Research: Applications and Algorithms*, 2nd edn., Wadsworth Publishing Company, Belmont, CA, 1991.