Journal of the Operational Research Society (2006) 57, 1035–1044

www.palgrave-journals.com/jors

Scheduling preventive railway maintenance activities

G Budai*, D Huisman and R Dekker

Erasmus University Rotterdam, Rotterdam, The Netherlands

A railway system needs a substantial amount of maintenance. To prevent unexpected breakdowns as much as possible, preventive maintenance is required. In this paper we discuss the preventive maintenance scheduling problem (PMSP), where (short) routine activities and (long) unique projects have to be scheduled in a certain period. To reduce costs and inconvenience for the travellers and operators, these activities should be scheduled together as much as possible. We present two versions of the PMSP, one with fixed intervals between two consecutive executions of the same routine work, and one with only a maximum interval. Apart from giving a math programming formulation for the PMSP and for its extension we also present some heuristics. In addition, we compare the performance of these heuristics with the optimal solution using some randomly generated instances.

Journal of the Operational Research Society (2006) **57,** 1035–1044. doi:10.1057/palgrave.jors.2602085
Published online 19 October 2005

Keywords: rail transport; maintenance; scheduling; optimization; heuristics

Introduction

Reliability, that is, punctuality and safety, are important aspects in railway transport. The quality of the railway infrastructure has a major influence on the reliability of the railway system as a whole. Therefore, it is important that there is enough preventive maintenance of the infrastructure (eg rail, ballast, sleepers, switches and fasteners). However, maintenance is expensive and budgets for maintenance are always under pressure. So it is important to reduce the maintenance costs without reducing the maintenance itself. This paper considers the clustering of maintenance activities on the same link in a network in order to reduce the disturbance of railway traffic.

We introduce the preventive maintenance scheduling problem (PMSP), where a schedule for the maintenance activities has to be found for one link such that the sum of the possession costs and the maintenance costs is minimized. The possession costs are mainly determined by the possession time, which is the time that a track requires for maintenance and cannot be used for railway traffic. The focus is on medium-term planning, determining which preventive maintenance works will be performed in what time periods (month/week/hours).

The contribution of this paper is twofold. First, we will provide a math programming formulation for the PMSP and for its extension and we prove that PMSP is NP-hard. Second, we will give four heuristics to solve this problem quickly. The latter is important, since in practice a whole network needs to be optimized. Moreover, to give an insight into the quality of the heuristics, we will compare them with (a lower bound on) the optimal solution.

Although we have formulated our problem for railway maintenance, the maintenance scheduling problem also arises in other public/private sectors, since preventive maintenance of other technical systems (machine, road, bridge, building, high voltage lines, electric power stations, aeroplanes) contains small routine works and also large projects (eg see Kralj and Petrović, 1988). Several papers have addressed the grouping of preventive maintenance activities in general. For instance, Van Dijkhuizen and Van Harten (1997) consider a clustering problem for frequency-constrained maintenance jobs with common and shared set-ups in an infinite-horizon setting. They only consider cyclic policies, while we also allow non-cyclic policies. Wildeman *et al* (1997) present a rolling-horizon approach to group maintenance activities on a short-term basis. In Dekker *et al* (2000), a general approach for the coordination of maintenance frequencies is presented. In both papers, there are no repetitive jobs, however, and a continuous-time approach is taken. The maintenance scheduling problem under a deterministic environment for a group of non-identical machines is studied by Hariga (1994). Here, only cyclic overhaul schedules are considered and it is assumed among others that for each machine the cycle time of the major overhauls is an integer multiple of the minor overhaul intervals. Another work relevant to maintenance scheduling is that of Sriskandarajah *et al* (1988) in which an approach for scheduling the frequency-based overhaul maintenance is

*Correspondence: G Budai, Econometric Institute, Erasmus University Rotterdam, PO Box 1738, 3000 DR, Rotterdam, The Netherlands.
E-mail: budai@few.eur.nl

given. The objective here is to satisfy the maintenance requirements of various train units as closely as possible to their due dates, since there is a cost for undertaking the maintenance tasks too early or too late.

The contribution of our paper to the existing literature on maintenance optimization is that we consider both repetitive jobs and projects in a finite horizon with a small (weekly) time step instead of infinite horizon or problems with projects only. This more closely resembles practical maintenance scheduling problems.

The remainder of the paper is organized as follows. In the next section, we place our paper in the railway world and the existing literature on maintenance scheduling of railway infrastructure. A formal problem description is given subsequently, followed by a mathematical programming formulation for PMSP and its extension. After that, we present some heuristics to solve this problem approximately. Finally, we conclude the paper with some computational experiments.

Railway maintenance planning

Preventive maintenance on railways can be subdivided into small routine works and projects. The routine (spot) maintenance activities consist of inspections and small repairs, for example, inspection of rails, switch, level crossing, overhead wire, signalling system and switch lubrication (see Esveld, 2001). These activities do not take much time to be performed and are done regularly, with frequencies varying between monthly and annual. The projects include renewal works and consist, for example, of ballast cleaning, rail grinding and tamping (see Esveld, 2001). They are only carried out once or twice every few years.

Improverail (http://www.tis.pt/proj/improverail/Downloads/D6Final.pdf accessed 10 September 2004) and Budai and Dekker (2002) show that the preventive railway maintenance works are carried out in most countries during train service. In the actual train timetable possible possession allocations are scheduled for maintenance so that it should not affect regular train services too much. Some methods are presented in Higgins (1998) and Cheung *et al* (1999). Many countries use timetabling software for finding free intervals or periods with less impact to the train operators. Carrying out maintenance works during train services might be unsafe for the maintenance crew. Therefore, in some other countries (eg The Netherlands) the maintenance works are carried out either during the night (when there are only a few or no trains), or during the day with interruption of the train service. In the first case, one can make a cyclic static schedule, which is made by Den Hertog *et al* (2001) and Van Zante-de Fokkert *et al* (2001) for the Dutch situation or a dynamic schedule with a rolling horizon, which is presented in Cheung *et al* (1999). Miwa *et al* (2001) present an optimal schedule for only one type of maintenance work, namely an annual schedule for the tie (sleeper) tamping.

Although our approach considers one rail link only, it can be extended to a network using the concept of the single track grids (STGs) presented in Van Zante-de Fokkert *et al* (2001). Solving the preventive railway maintenance scheduling problems to optimality involves large-scale mixed-integer programming problems. Several heuristics are developed to solve these problems faster, for example, Higgins (1998) applies tabu search techniques, while Grimes (1995) uses genetic algorithm and genetic programming methods.

Problem description

In this paper, our aim is to give a schedule for preventive maintenance activities in a finite horizon, such that jobs are clustered as much as possible in the same period. Combining jobs as much as possible results in cost savings, since execution of a group of activities requires only one track possession.

The PMSP can be defined as follows. Given a set of routine activities and projects, we like to schedule them such that the track possession costs and maintenance costs are minimized. Some routine works and projects may be combined to reduce the possession time, but others may exclude each other. For each routine work, the planning cycle, that is, the maximum number of time periods between two consecutive executions, is known. Furthermore, the number of time periods elapsed since the routine works have been carried out the last time is given. A list of projects that need to be performed in the planning period, duration and the earliest and latest possible starting times for each project are given. The execution costs of each routine work and project and the costs of having a track possession in the planning period are known. A list of works (routine works and/or projects) that can be combined is given. Since our planning horizon is finite and the routine activities are repetitive, an end-of-horizon valuation is needed.

At first sight, the model presented in this paper seems to be related to the machine scheduling problem and the multi-project scheduling problem in an abstract way. The similarity among these three problems is that there are jobs, with given durations and given time windows between two consecutive executions, which have to be scheduled in a certain time period. One of the differences is that in the PMSP the routine works have a repetitive character. Furthermore, the objective of the PMSP is different from the objectives of the other mentioned problems. Namely, we try to schedule the jobs together as much as possible and not necessarily as soon as possible.

Mathematical formulation

Let T be a set of discrete time periods (eg months, weeks) in which the maintenance activities need to be scheduled,

$|T|$ is the planning horizon. The model parameters are as follows:

PA set of projects
RA set of routine maintenance works
A $PA \cup RA$ set of all activities,
C $\{(m, n)|$ work m is combinable with $n, \forall m, n \in A\}$ set of combinable works,
L^a cycle length of the routine work $a \in RA$,
F^a $\lfloor \frac{|T|}{L^a} \rfloor$ frequency of the routine work $a \in RA$,
G^a number of periods elapsed since routine work $a \in RA$ was in the past (before the planning horizon starts) for the last time carried out,
LC^a $\{t \in T | 1+|T|-L^a \leqslant t \leqslant |T|\} \subseteq T$ set of time periods from the last planning cycle for routine work $a \in RA$,
b_t^a $(|T|-t)/L^a$ length of the remaining interval until the end of planning horizon divided by the length of the planning cycle for routine work $a \in RA$ and for time period $t \in LC^a$,
$T_p \subseteq T$ set of possible start points of project $p \in PA$
D_p duration of project $p \in PA$,
pc_t possession cost in period $t \in T$,
mc_a maintenance cost per time period for carrying out work $a \in A$.

The following binary decision variables are defined:

x_t^a binary variable that denotes whether activity $a \in A$ is assigned to period $t \in T$ ($x_t^a = 1$), or not ($x_t^a = 0$),
z_t^a binary variable that denotes whether activity $a \in RA$ is carried out for the last time in the planning horizon at time $t \in LC^a$ ($z_t^a = 1$), or not ($z_t^a = 0$)
m_t binary variable that denotes whether the track is used for preventive maintenance work at time $t \in T$ ($m_t = 1$), or not ($m_t = 0$),
y_t^p binary variable that denotes whether the execution of project $p \in PA$ starts at time $t \in T$ ($y_t^p = 1$), or not ($y_t^p = 0$).

The PMSP can now be formulated as follows:

$$\text{(PMSP)} \qquad \text{Min} \sum_{t \in T} pc_t m_t + \sum_{a \in A} \sum_{t \in T} mc_a x_t^a + \sum_{a \in RA} \sum_{t \in LC^a} mc_a b_t^a z_t^a \tag{1}$$

s.t.

$$\sum_{t=1}^{L^a - G^a} x_t^a \geqslant 1, \quad \forall a \in RA \tag{2}$$

$$\sum_{s=0}^{L^a - 1} x_{t+s}^a \geqslant 1, \quad \forall a \in RA, \ 1 \leqslant t \leqslant |T| - L^a + 1 \tag{3}$$

$$\sum_{t \in LC^a} z_t^a \geqslant 1, \quad \forall a \in RA \tag{4}$$

$$z_t^a \leqslant x_t^a, \quad \forall a \in RA, \quad t \in LC^a \tag{5}$$

$$x_t^m + x_t^n \leqslant 1, \quad \forall t \in T, \quad (m, n) \notin C \tag{6}$$

$$\sum_{t \in T_p} y_t^p = 1, \quad \forall p \in PA \tag{7}$$

$$x_s^p \geqslant y_t^p, \quad \forall p \in PA, \quad t \in T_p, \quad s = t, \ldots, t + D_p - 1 \tag{8}$$

$$m_t \geqslant x_t^a, \quad \forall a \in A, \quad t \in T \tag{9}$$

$$x_t^a, z_t^a, y_t^p, m_t \in \{0, 1\}, \quad \forall a \in A, \quad p \in PA, \quad t \in T \tag{10}$$

The first two terms in the objective function are the sum of possession costs and the maintenance costs. In our formulation of the PMSP we require that the interval between successive executions of an activity is bounded, but it does not always need to be of the same value. Accordingly, it is not clear beforehand how many executions will be in the planning horizon. The last term in the objective function is used to value the last interval. It is intended to eliminate the end-of-horizon effect, but it creates difficulty in modelling.

Constraints (2) guarantee that each work is carried out at least once in the possible truncated first planning cycle. Constraints (3) ensure that the works until the end of planning horizon are scheduled at most L^a time periods from each other. Constraints (4)–(5) define the length of the last interval. Basically, if in the last planning cycle there are two or more executions of the same work, then z_t^a is set to one for only one time period t that results in the shortest remaining interval until the end of planning horizon. On the same link and at the same time only combinable activities can be carried out. This is ensured by constraints (6). These combinable jobs can be either routine works or projects. Constraints (7) guarantee that each project is executed once. Furthermore, constraints (8) ensure that each project is assigned to the right number of time periods and the starting time for performing the projects is in the interval (earliest possible starting time, latest possible starting time). Furthermore, these projects are assigned to subsequent intervals. Constraints (9) ensure that time period $t \in T$ will be occupied for preventive maintenance work if and only if for that time period on this segment at least one work is planned. Finally, constraints (10) ensure that the decision variables are binary.

If we consider an individual schedule of a given routine work, then the most cost-effective way to schedule it is always at the maximum length of its planning cycle, that is, exactly at L^a periods. In this way, no extra maintenance work is done within the planning horizon, so the maintenance cost decreases but there are less opportunities to combine executions of the works in one period, so the track possession time (or track possession cost) increases.

We extend the PMSP by restricting the time periods between two consecutive executions of the same work exactly to L^a time periods and we call this problem the restricted preventive maintenance scheduling problem (RPMSP), which is formulated as follows:

$$\text{(RPMSP)} \qquad \text{Min} \sum_{t \in T} pc_t m_t + \sum_{a \in A} \sum_{t \in T} mc_a x_t^a + \sum_{a \in RA} \sum_{t \in LC^a} mc_a b_t^a x_t^a \tag{11}$$

s.t.

$$\sum_{t=1}^{L^a - G^a} x_t^a = 1, \quad \forall a \in RA \tag{12}$$

$$x_t^a = x_{t+qL^a}^a, \quad \forall a \in RA,\ 1 \leqslant t \leqslant L^a, \quad q \geqslant 1 \tag{13}$$

$$x_t^m + x_t^n \leqslant 1, \quad \forall t \in T, \quad (m,n) \notin C \tag{14}$$

$$\sum_{t \in T_p} y_t^p = 1, \quad \forall p \in PA \tag{15}$$

$$x_s^p \geqslant y_t^p, \quad \forall p \in PA, \quad t \in T_p, \quad s = t, \ldots, t + D_p - 1 \tag{16}$$

$$m_t \geqslant x_t^a, \quad \forall a \in A, \quad t \in T \tag{17}$$

$$x_t^a, y_t^p, m_t \in \{0, 1\}, \quad \forall a \in A, \quad p \in PA, \quad t \in T \tag{18}$$

The objective again minimizes the sum of possession costs, the maintenance costs and the penalty cost paid if the last execution of the routine works is carried out too early in the planning horizon compared to the end of horizon. Constraints (12) ensure that each routine maintenance work is scheduled exactly once in the first allowed planning cycle and then constraints (13) guarantee that until the end of the planning horizon the works for the other cycles will be defined as well, ensuring exactly L^a time periods between two subsequent occurrences of the same job. Constraints (14)–(18) have been already explained before.

Theorem 1 *The preventive maintenance scheduling problem is NP-hard.*

Proof It is well known that the following problem is NP-complete (see Garey and Johnson (1979)). Given an undirected graph G and a positive integer k, decide whether or not G can be coloured with k colours such that no edge is incident to vertices of the same colour. We show that it is NP-complete to decide whether the PMSP, as well as the RPMSP, has a solution of objective value 0 (ie a zero-cost solution).

Consider a graph $G=(V,E)$ and let $n=|V|$ be the number of vertices. We assume that $V=\{1, \ldots, n\}$. We construct an instance of the PMSP with $|T|=n$, with n routine works, each of them having $L^a=n$ and $G^a=0$, and without projects. The ith and jth routine works have a conflict if ij is an edge in G. We assume that the maintenance costs are zero. The first k time periods have possession cost 0, the remaining $n-k$ periods have possession costs 1. This is an instance of both the PMSP and the RPMSP. The size of our construction is clearly polynomial in the size of graph G. We claim this instance has a solution with objective value zero if and only if graph G can be coloured with k colours. Indeed, suppose that G has a k-colouring. If a vertex has colour c, assign the corresponding routine work to the cth period. Then we obtain an assignment without conflicts. This assignment has zero cost. Conversely, consider a zero-cost solution of the railway problem. Then each routine work is scheduled for any of the first k periods. Colour a vertex of G by colour c if its corresponding routine work is scheduled for the cth period. This yields a k-colouring of graph G. □

Solution approach

The PMSP and RPMSP are modelled in GAMS and they are solved afterwards with the MIP solver CPlex 7.1. Solving these optimization problems, especially PMSP, to optimality for a single link, more than 15 types of maintenance works and for more than 3–4 years, requires a large amount of time. To improve the performance of the PMSP, we added to the model the following redundant constraints:

$$z_t^a + x_s^a \leqslant 1, \quad \forall a \in RA, \quad t \in LC^a, \quad t + 1 \leqslant s \leqslant |T| \tag{19}$$

Constraints (19) guarantee that for each routine work $a \in RA$ if $d \in LC^a$ is the last time period when work a is carried out (ie $x_d^a=1$), then $z_t^a=0$ for all the time periods from the last planning cycle smaller than d. Constraints (19) improved somewhat the performance of the PMSP, but it is still impossible to get for some instances optimal solutions within 3 h. More details about performances are presented in the section on Computational results.

Since it takes too much time to find the optimal solution it might be better to settle for a non-optimal solution that has somewhat larger overall cost, but which is still quite close to the optimal objective value, and which can be found in a reasonable time. Therefore, our further purpose is to develop heuristics for solving the PMSP and RPMSP. In the literature, we could not find any algorithms that can be used for solving our problems, since the problems for which those heuristics were developed are somewhat different from our problem.

In the following section, we develop two heuristics for solving the PMSP and another two for solving the RPMSP. These heuristics are greedy in the sense that they try to combine every activity together. In the next section, each of them is presented in detail. It is worth mentioning that in each of the approximation methods we try to schedule the routine maintenance works and projects together. If two or more routine works cannot be combined then they will be scheduled for separate time periods.

Heuristics

First, we explain the two heuristics for solving the RPMSP, and thereafter the other two for the PMSP, since some steps in the first two heuristics are used in the last ones as well, namely, steps for making a preliminary schedule with restricted planning cycles.

Before we describe the heuristics, we recall the input of the problem and some notation. Given are a set of routine maintenance works RA with their planning cycle L^a and frequencies F^a, $\forall a \in RA$. In the beginning of the planning horizon T there are G^a periods elapsed since maintenance work $a \in RA$ was in the past for the last time carried out. A set of projects PA with duration D_p and a set of possible start points $T_p \subseteq T$ are given. There is also a list of works that cannot be scheduled together. $e_{a,k}$ denotes the kth execution time of work $a \in RA$, pc_t the possession cost in period $t \in T$ and mc_a cost for carrying out maintenance work $a \in A$.

The first heuristic, called Single component strategy (SCS), starts with making the best individual schedules for each of the routine works. In the literature, this heuristic is sometimes called Decomposition approach. The idea of the SCS can be found also in inventory control as a strategy for independent ordering (see Chopra and Meindl (2001)). In the SCS the time periods between two consecutive executions of the same routine work are kept constant. The projects are added later to this schedule. In this algorithm, we do not look at the possible combination of the works, but focus only on the individual plans. However, some works will be combined anyway due to the structure of the problem.

Single component strategy (SCS)

Step 0: (For $j = 1, \ldots, n$) Make an individual schedule for work j such that the sum of the possession cost, maintenance cost and penalty paid for late execution per time horizon $|T|$ is minimized. These costs are calculated for each work separately, not looking at the savings in the possession cost resulting by combining some works. There might be certain periods where work j cannot be scheduled due to the earlier choices made for works $1, \ldots, j-1$.

Step 1: Choose a project p from set PA with the earliest possible starting time. In the allowed time interval T_p, find the best time moment for performing it together, as much as possible, with already scheduled routine maintenance works.

Step 2: Calculate the overall cost of the schedule resulting from Step 0 and Step 1.

The next heuristic, called Most frequent work first (MFWF), starts with scheduling the works having the highest frequency. The time periods between two consecutive executions of the same work are again kept constant. First, we make a schedule for the most frequent work. Then the other works are scheduled such that the increase in the overall cost is minimal. After the best possible schedule is found for the routine works, the projects are also scheduled, each of them in their allowed time intervals.

Most frequent work first heuristic (MFWF)

Step 0: Order the set of routine works $RA = \{1, \ldots, n\}$ such that the planning cycles are in increasing order, that is, $L^1 \leqslant L^2 \leqslant \ldots \leqslant L^n$. Schedule routine work 1 at its maximum interval, that is, in periods 1, $1 + L^a$, $1 + 2L^a$, …, etc.

Step 1: (For $j = 2, \ldots, n$) Schedule routine work j such that the increase in the sum of possession cost, maintenance cost and penalty cost for too late execution of work j in the planning horizon is minimal. The first execution time of work j should be in the period $[1, L^j - G^j]$ and there might be certain periods where work j cannot be scheduled due to the earlier choices made for works $1, \ldots, j-1$.

Step 2: Choose a project p from set PA with the earliest possible starting time. In the allowed time interval T_p, find the best time moment for performing it together, as much as possible, with already scheduled routine maintenance works.

Step 3: Repeat Step 1 and 2 for all values where work 1 can be carried out for the first time, namely, in the period $[1, L^1 - G^1]$. The schedule resulting in the minimum cost is chosen.

We expect that the second heuristic will perform better than the first one due to optimization done in Step 1 and Step 3. Our purpose with the first heuristic is to show how much the overall costs can be decreased if some global optimization steps are also included in the algorithm.

In the last two heuristics for solving PMSP, we allow shorter intervals between two consecutive executions of the same routine work, creating more possibilities to combine work. This results in lower possession cost, but in most of the cases higher maintenance cost. Both heuristics start with making a schedule with fixed intervals, using MFWF (SCS or the IP model for the RPMSP can be used as well). After that a better/cheaper schedule is searched by modifying, that is, shortening, the planning cycles and creating more opportunities for combination of the works.

The idea of the third heuristic, which is called Opportunity-based heuristic (OBH) comes from the Opportunity-based maintenance model (see eg Dekker *et al* (1997)). In that model, preventive maintenance is carried out at opportunities that are generated by failure of a particular unit in the system. Hence, planned maintenance activities are combined with unplanned activities. In the OBH, the execution times of the most frequent work will be used as opportunities for execution times for the other routine works as well. First of all, a preliminary plan is made, where we schedule first the works having the highest frequency. After that we check whether fitting all the execution times of the other works into the schedule of the most frequent work leads to a lower cost than creating separate, own opportunities requiring more possessions.

Opportunity-based heuristic (OBH)

Step 0: Order the set of routine works $RA=\{1, \ldots, n\}$ such that the planning cycles are in increasing order, that is $L^1 \leqslant L^2 \leqslant \ldots \leqslant L^n$. Schedule these routine works together with the projects, using the MFWF heuristic. The execution times of the most frequent job give the initial values of the opportunities' list, $Opp = \{e_{1,1}, e_{1,2}, \ldots, e_{1,F^1}\}$.

Step 1: (For $j=2, \ldots, n$ and for all execution times of work j). If a given execution time of work j, $e_{j,k}$, is not yet in the Opp list, then check whether shifting $e_{j,k}$ forward to the closest earlier opportunity leads to a lower cost than the cost of a new possession. If $(mc_jS)/L^j > pc_t$, where S is the length of the time period between the current execution time and the closest earlier opportunity, then a new opportunity is created and $e_{j,k}$ is added to the Opp list. If not, then all the execution times of routine work j are shifted S time periods forward. In the latter situation sometimes forward shifts of the execution times of work j are not possible, because at that time the execution of another work, which cannot be combined with work j, has been already planned. If shifting the execution times forward is still possible, then adjust the frequencies as well, since shifting forward actual execution times might lead to more executions within the planning horizon. If in the new schedule of work j in the L^j+1 consecutive time periods there are more than two works scheduled, then the middle execution can always be deleted from the schedule, decreasing the possession and maintenance cost. If the new schedule of work j results in a lower overall cost than before, then the schedule from Step 1 is modified with this new schedule, otherwise the schedule found in Step 1 is used.

Step 2: Rescheduling of the projects is carried out according to Step 2 from the MFWF heuristic. The schedule resulting in the minimum cost is chosen.

The following heuristic (MCWF) is based again on opportunity-based maintenance. In this case, the execution times of the most costly work and half of the time intervals between two subsequent executions are used as opportunities for execution times for the other routine works as well. Basically, MCWF starts with making a preliminary plan by scheduling the most costly works first. In the preliminary plan, we shift the execution times of the other works such that in the end all works are carried out only in the listed opportunities, even if it results in a higher maintenance cost. The main difference between the last two heuristics is that in the first heuristic we shift the executions of the routine works to the closest earlier opportunity only if this is locally beneficial and in the second one we always move execution time forward, even if this action has a negative effect on the costs.

Most costly work first heuristic (MCWF)

Step 0: Order the set of routine works $RA=\{1, \ldots, n\}$ such that their maintenance costs per planning horizon $|T|$ are in decreasing order, that is, $mc_1F^1 \geqslant mc_2F^2 \geqslant \ldots \geqslant mc_nF^n$.

Step 1: Schedule these routine works together with the projects, using the MFWF heuristic (with the RA set ordered in Step 0).

Step 2: The execution times of the most costly job and the rounded-down value of the average of its two consecutive execution times give the initial values of the opportunities list, $Opp = \{\lfloor t_{1,1}/2 \rfloor, t_{1,1}, \lfloor (t_{1,1} + t_{1,2})/2 \rfloor, t_{1,2}, \ldots, t_{1,F^1}\}$.

Step 3: (For $j=2, \ldots, n$ and for all execution times of work j). If a given execution time of work j, $t_{j,k}$, is not yet in the Opp list, then $t_{j,k}$ is shifted forward to the closest earlier opportunity even if this leads to a higher overall cost. Take into account that sometimes shifting forward once the execution time of work j is not possible, because at that time the execution of another work, which cannot be combined with work j, has been planned. If this is the case and there is still one more earlier opportunity, which does not coincide with $t_{j,k-1}$, then try to shift $t_{j,k}$ one more period forward. If this leads again to conflicting execution times, then we conclude that we cannot get a feasible solution. If shifting an execution time forward is still possible, then adjust the frequencies as well, since shifting forward actual execution times might lead to more executions. Check whether in the new schedule of work j in the L^j+1 consecutive time periods there are more than two works scheduled. If yes, then delete the middle execution from the schedule, decreasing thereby the possession and maintenance cost.

Step 4: Rescheduling of the projects is carried out according to Step 2 from the MFWF. The schedule resulting in the minimum cost is chosen.

As we already mentioned, all four heuristics are greedy heuristics, some of them containing improvement steps. The complexity of these algorithms is as follows: MFWF, OBH and MCWF have complexity of $O(n(n+p)T^3)$ and SCS $O(n(n+p)T^2)$, where n is the number of routine maintenance works, p is the number of projects and $|T|$ is the planning horizon.

Computational results

Experimental set-up and implementation

The planning horizon for the generated instances is 2 years and the discrete time periods are weeks. Furthermore, we assume that each routine maintenance work has different planning cycles, and consequently, different frequencies and different maintenance costs. We assume that the track possession cost is the same for each week within the planning horizon.

To test the algorithms, we generated instances with $n=15$ and $n=25$ routine works. For each size we generated 10 instances. The generated values for the planning cycles (L^a), for the number of periods elapsed since maintenance work a was in the past for the last time carried out (G^a), for the maintenance costs (mc_a), for the number of projects which have to be performed until the end of the planning horizon (p), the possible earliest (ES_p) and latest starting times (LS_p) and the duration of the projects (D_p) are uniformly distributed random numbers, as follows: $L^a \sim U[4, 52]$, $G^a \sim U[0, L^a]$, $mc_a \sim U[1, 100]$, $p \sim U[0, 2]$, $ES_P \sim U[1, 104]$, $LS_P \sim U[1, 104]$, $D_p \sim U[1, 6]$.

We tested our models and algorithms for two scenarios. In the first scenario, we assume that each routine work can be combined with all other routine works and projects, but the projects cannot be combined with other projects. In the second scenario, we want to see the effect in the exact models and heuristics of having routine works that cannot be combined. Therefore, we assume in the second scenario that a group of two works and a group of three works cannot be combined. Thus the works within a group cannot be carried out at the same time. These works are arbitrarily chosen. Furthermore, we test both models and all four heuristics for different possession costs ($PossC=25$ and $PossC=75$).

All the tests are executed on a Pentium IV 1.60 GHz (256MB RAM) personal computer, using CPlex 7.1 for calculating the LP and the optimal integer solution. The results, after running the model for the generated instances, are shown in Tables 1 and 2. The number of variables increased by almost 100% if some routine works could not be combined.

Table 1 Problem specifications

	Scenario 1				*Scenario 2*			
	n = 25		*n = 15*		*n = 25*		*n = 15*	
	Average	*St.dev*	*Average*	*St.dev*	*Average*	*St.dev*	*Average*	*St.dev*
SFre	138.7	22.5	80.4	17.5	138.7	22.5	80.4	17.5
NV	2843.6	124.5	1768.2	97.4	5443.6	124.6	3328.2	97.5
NC	3860	203.6	2366.6	192	9866.9	3706.2	11 315.8	2605.1

SFre, the sum of works' frequencies; *NV*, the number of variables; *NC*, the number of constraints.

Table 2 Computational results for RPMSP

Method PossC		*Scenario 1 Average n = 25*	*Scenario 1 Average n = 15*	*Scenario 2 Average n = 25*	*Scenario 2 Average n = 15*
RPMSP	*VOpt*	10 030.9	5797.3	10 077.3	5829.9
25	*TCPU(s)*	358.9	90.3	186.3	274.2
	VLP	9358.2	5295.9	9358.2	5295.9
	RdOLP(%)	6.7	8.6	7.1	9.1
SCS	*ObjV*	10 645.3	6236.6	10 775.5	6249.8
25	*TCPU(s)*	0.01	0.01	0.01	0.01
	RdOH(%)	6.1	7.6	6.9	7.2
MFWF	*ObjV*	10 124.2	5848.9	10 172.7	5886.1
25	*TCPU(s)*	0.013	0.014	0.013	0.012
	RdOH(%)	0.9	0.8	0.9	0.9
RPMSP	*VOpt*	12 433.9	7667.3	12 562.6	7759.8
75	*TCPU(s)*	442.2	81.9	150.4	269.1
	VLP	10 921.7	6571.9	10 921.7	6571.9
	RdOLP(%)	12.2	14.2	13.1	15.3
SCS	*ObjV*	14 540.3	9131.6	14 560.1	9169.8
75	*TCPU(s)*	0.01	0.01	0.01	0.01
	RdOH(%)	16.9	19.1	15.9	18.1
MFWF	*ObjV*	12 669.9	7817.2	12 832.5	7897.6
75	*TCPU(s)*	0.014	0.012	0.013	0.012
	RdOH(%)	1.9	1.9	2.1	1.7

VOpt, optimal value of RPMSP; *TCPU*, CPU time; *VLP*, LP relaxation value; *RdOLP*, relative difference between the *VOpt* and *VLP*; *ObjV*, solution value of the heuristic; *RdOH*, relative difference between *VOpt* and *ObjV*.

Results for the RPMSP and PMSP

From the test results we conclude that increasing the possession cost decreases the number of possessions and increases the number of extra maintenance works. If the track possession is very cheap ($PossC < 10$) then the number of possessions and the number of activities are almost the same in both models. The PMSP results in 2 to 12% lower overall costs than the RPMSP; however, the computational time is much longer and sometimes it is even impossible to get an optimal solution. Actually, RPMSP could be solved to optimality within 8 min for all the instances, even in the scenario where some works cannot be combined. However, in the case of the PMSP only in 29% of the instances we could get an optimal solution within 3 h. Comparing Scenario 2 with 1, we can conclude that if some works cannot be combined, then the number of possessions, the overall maintenance cost and the relative difference (*RdOLP*) between the optimal value (*VOpt*) and the LP relaxation value (*VLP*) for both models increases by approximately 0.4–2.4%. An important observation that was not reported in the tables is that the gap between *VOpt* and *VLP* is much bigger for the instances where besides the routine maintenance works there are some projects as well. In our opinion, this is not a very shocking result, since from the beginning we knew that scheduling routine works and projects together gives a very complicated structure to the problem. On the other hand, combining routine works with projects means that once there is a possession for a given project, some other routine works can be carried out at the same time, saving some extra possessions.

Results for the heuristics

In Tables 2 and 3 the results for the heuristics are given. We recall that SCS and MFWF are heuristics developed for solving the RPMSP, and OBH and MCWF are for solving the PMSP.

Table 2 shows that the MFWF heuristic performs well; for 15% of the instances we get the optimal solution, especially if the possession cost is relative small ($PossC < 15$). As the

Table 3 Computational results for PMSP

		Scenario 1 Average		*Scenario 2 Average*	
Method PossC		*n = 25*	*n = 15*	*n = 25*	*n = 15*
PMSP	*VOpt*	10 036.3*	5681.4	10 086.8*	5705.2
25	*TCPU(s)*	>10 800	9455.8	>10 800	8462.1
	VLP	9124.5	5116.6	9124.5	5116.8
	RdOLP(%)	9.1	9.9	9.5	10.3
	RdOLB(%)	4.2	2.9	4.2	1.7
OBH	*ObjV*	10 085.2	5771.4	10 132.9	5816.5
25	*TCPU(s)*	0.014	0.014	0.013	0.013
	RdOH(%)	0.5	1.5	0.5	1.9
	RdHLB(%)	4.9	4.7	4.9	3.7
MCWF	*ObjV*	10 410.2	6041.3	10 283.3	6030.3
25	*TCPU(s)*	0.015	0.011	0.016	0.01
	RdOH(%)	3.7	6.3	1.9	5.7
	RdHLB(%)	8.3	9.6	6.5	7.5
PMSP	*VOpt*	11 574.7*	6720.7	11710.1*	6889.1
75	*TCPU(s)*	>10 800	7424.4	>10 800	7163.2
	VLP	10 161.2	5974.2	10161.2	5974.2
	RdOLP(%)	12.2	11.1	13.2	13.2
	RdOLB(%)	5.9	2.6	5.3	2.3
OBH	*ObjV*	12 365.9	7201.6	12 505.6	7359.1
75	*TCPU(s)*	0.014	0.012	0.014	0.014
	RdOH(%)	6.8	7.1	6.8	6.8
	RdHLB(%)	13.5	10.1	12.8	9.3
MCWF	*ObjV*	12 361.9	7524.3	12 363.1	7811.3
75	*TCPU(s)*	0.012	0.012	0.012	0.015
	RdOH(%)	6.8	11.9	5.5	13.4
	RdHLB(%)	13.5	15.0	11.5	16.1

*No optimal solution could be found within 3 h.
VOpt, optimal value of PMSP or the best value found within 3 h; *TCPU*, CPU time; *VLP*, LP relaxation value; *RdOLP*, relative difference between the *VOpt* and *VLP*; *RdOLB*, relative difference between *VOpt* and the last MIP lower bound; *ObjV*, solution value of the heuristic; *RdOH*, relative difference between *VOpt* and *ObjV*; *RdHLB*, relative difference between *ObjV* and the last MIP lower bound.

relative difference (*RdOH*) between the optimal value (*VOpt*) and the heuristic value (*ObjV*) shows, SCS performs quite poorly, and it results in a 6–19% higher overall cost than the RPMSP. On average, savings from 5 to 15% can be achieved by using MFWF *versus* SCS. The CPU times (*TCPU* in seconds) for both heuristics are very small, 1–2 hundredths of seconds.

The solution is found right away also in case of OBH and MCWF heuristics. The *RdOH* and the relative differences (*RdHLB*) between the heuristic value and the best MIP lower bound found by CPlex after running PMSP for 3 h show that OBH performs better than MCWF. Especially, if $PossC = 25$, then there is 0.5–1.9% increase in the overall costs in comparison with *VOpt* for PMSP, *versus* 1.9–6.3% increase in costs for MCWF. However, if the possession increases, then OBH does not perform as well as before, resulting in $PossC = 75$, an approximately 6% relative difference between the heuristic value and the best MIP lower bound found. The reason why this happens is that in Step 1 we decide whether it is locally worth moving forward an execution time or not, without checking what the consequences of this movement are for the whole schedule. Shifting forward an execution of a given work means that we might be forced to include another execution time to the end of planning horizon, increasing thereby the number of maintenance works, but reducing somewhat the length of the remaining interval from the last execution to the end of horizon. The increase in the overall costs is even more than 6% for MCWF if the possession cost increases. In summary, we can say that including some local improvements/optimization steps in the algorithm results in a lower overall cost (possession cost + maintenance cost + penalty paid for too early execution of the last work within the planning horizon).

Conclusions

Since rail is an important transportation mode, proper maintenance of the existing lines, repairs and replacements carried out in time are all important to ensure efficient operation. Moreover, since some failures might have a strong impact on the safety of the passengers, it is important to prevent these failures by carrying out in time, and according to some predefined schedules, preventive maintenance works. Since the infrastructure maintenance costs represent a large part of the total operating costs, there is a need for developing operations research tools that help the maintenance planners to come up with optimal maintenance plans.

In this paper, we presented a mathematical programming formulation for the PMSP. Maintenance works are assigned to different time periods (months/weeks), minimizing the track possession cost and the maintenance cost. Since the maintenance scheduling problem is a complex optimization problem and for a large set of instances it is difficult and time consuming to solve the problem to optimality, it was necessary to develop some approximation methods that still give solutions close to the optimal ones. Two heuristics, namely MFWF and SCS were developed for RPMSP and other two, OBH and MCWF for the PMSP.

From the results we can conclude that MFWF gives the best results for solving the RPMSP and OBH for solving the PMSP. Furthermore, solving the PMSP for different instances, scenarios and different possession values results in, on the one hand much lower possession cost than RPMSP, but on the other hand higher maintenance cost. If we compare the average optimal value of the RPMSP and the best solution value found within 3 h for the PMSP, then one can see that restricting the time periods between two consecutive executions of the same routine work constant until the end of horizon leads to a 1.8–12% increase in overall costs. In other words, having more freedom for choosing the execution times and increasing the possibilities to combine activities in one period results in a lower maintenance cost. However, if the number of routine works increases, then it is more and more difficult to solve the PMSP to optimality within 3 h. If the planners want to have, in very short time, a good (close to optimal) schedule, then either RPMSP or one of the heuristics can be used effectively.

As a final remark we would like to mention that the model presented in this paper is just a basic model, but it can be extended to solve many types of practical problems, since in reality there are many more constraints that a maintenance planner has to take into account.

Acknowledgements—We thank Gábor Maróti for his useful comments and the experts from Strukton Railinfra, The Netherlands, for providing us information about planning and performing railway maintenance activities.

References

Budai G and Dekker R (2002). An overview of techniques used in planning railway infrastructure maintenance and its effect on capacity. In: Geraerds WMJ and Sherwin D (eds). *Proceedings of the IFRIMmmm (Maintenance Management and Modelling Conference) 2002, Växjö, Sweden*. Växjö University, Växjö, Sweden.

Cheung BSN, Chow KP, Hui LCK and Yong AMK (1999). Railway track possession assignment using constraint satisfaction. *Eng Appl Artif Intell* **12**: 599–611.

Chopra S and Meindl P (2001). *Supply Chain Management: Strategy, Planning, and Operation*. Prentice Hall: New Jersey.

Dekker R, Van Egmond R, Frenk JBG and Wildeman RE (2000). A general approach for the coordination of maintenance frequencies. In: Ben-Daya M, Duffuaa SO and Raouf A (eds). *Maintenance, Modeling and Optimization*. Kluwer Academic Publishers, Boston/Dordrecht/London, pp 245–282.

Dekker R, Wildeman RE and Van der Duyn Schouten FA (1997). A review of multi-component maintenance models with economic dependence. *Math Methods Opns Res* **45**: 411–435.

Den Hertog D, Van Zante-de Fokkert JI, Sjamaar SA and Beusmans R (2001). *Safe track maintenance for the Dutch railways, Part I: Optimal working zone division*. Technical report, Tilburg University, The Netherlands.

Esveld C (2001). *Modern Railway Track*. MRT-Productions: Zaltbommel.

Garey MR and Johnson DS (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman: San Francisco.

Grimes CA (1995). Application of genetic techniques to the planning of railway track maintenance work. In: Zalzala AMS (ed). *First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, GALESIA, 1995, Sheffield, UK*. Conference Publication 414. IEE, University of Sheffield, pp 467–472.

Hariga M (1994). A Deterministic Maintenance-Scheduling Problem for a Group of Non-identical Machines. *Int J Opns Prod Mngt* **14**: 27–36.

Higgins A (1998). Scheduling of railway track maintenance activities and crews. *J Opl Res Soc* **49**: 1026–1033.

Kralj B and Petrović R (1988). Optimal preventive maintenance scheduling of thermal generating units in power systems—A survey of problem formulations and solution methods. *Eur J Opnl Res* **35**: 1–15.

Miwa M, Ishikawa T and Oyama T (2001). Modeling the optimal decision-making for multiple tie tamper operations. In: *Proceedings of the fifth World Congress on Railway Research*, Cologne, Germany (on CD-ROM).

Sriskandarajah C, Jardine AKS and Chan CK (1988). Maintenance scheduling of rolling stock using a genetic algorithm. *Eur J Opnl Res* **35**: 1–15.

Van Dijkhuizen G and Van Harten A (1997). Optimal clustering of frequency-constrained maintenance jobs with shared set-ups. *Eur J Opnl Res* **99**: 552–564.

Van Zante-de Fokkert JI, Den Hertog D, Van den Berg FJ and Verhoeven JHM (2001). *Safe track maintenance for the Dutch railways, Part II: Maintenance schedule*. Technical Report, Tilburg University, The Netherlands.

Wildeman RE, Dekker R and Smit ACJM (1997). A dynamic policy for grouping maintenance activities. *Eur J Opnl Res* **99**: 530–551.

Received September 2004;
accepted August 2005 after two revisions