Centrum voor Wiskunde en Informatica

REPORTRAPPORT

PNA

Probability, Networks and Algorithms

Probability, Networks and Algorithms

PNA

Maintenance Routing for Train Units: the Transition Model

G. Maróti, L.G. Kroon

REPORT PNA-E0415 AUGUST 2004

CWI is the National Research Institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organization for Scientific Research (NWO).
CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3711

Maintenance Routing for Train Units: the Transition Model

ABSTRACT

Train units need regular preventive maintenance. Given the train units that require maintenance in the forthcoming 1-3 days, the rolling stock schedule must be adjusted so that these urgent units reach the maintenance facility in time. In an earlier paper Maroti and Kroon propose a model that requires a large amount of input data. In this paper we describe a less involved multicommodity flow type model for this maintenance routing problem. We study the complexity of the problem. It turns out that the feasibility problem for a single urgent train unit is polynomially solvable but the optimisation version is NP-hard. Finally, we report our computational experiments on practical instances of NS Reizigers, the main Dutch operator of passenger trains.

2000 Mathematics Subject Classification: 90B06; 90B10; 90C35
Keywords and Phrases: Railway transportation; Maintenance routing

Maintenance Routing for Train Units: the Transition Model

Gábor Maróti

CWI, Amsterdam and
NS Reizigers, Utrecht
P.O.Box 94079, 1090 GB Amsterdam, The Netherlands
E-mail: G.Maroti@cwi.nl

Leo Kroon

NS Reizigers, Utrecht and
Erasmus University, Rotterdam
P.O.Box 1738, 3000 DR Rotterdam, The Netherlands
E-mail: L.Kroon@fbk.eur.nl

Abstract

Train units need regular preventive maintenance. Given the train units that require maintenance in the forthcoming 1–3 days, the rolling stock schedule must be adjusted so that these urgent units reach the maintenance facility in time. Maróti and Kroon [11] propose a model that requires a large amount of input data. In this paper we describe a less involved multicommodity flow type model for this maintenance routing problem. We study the complexity of the problem. It turns out that the feasibility problem for a single urgent train unit is polynomially solvable but the optimisation version is NP-hard. Finally, we report our computational experiments on practical instances of *NS Reizigers*, the main Dutch operator of passenger trains.

AMS Subject classification: 90B06, 90B10, 90C35
Keywords: Railway transportation, maintenance routing.

1 Introduction

The Dutch passenger railway operator *NS Reizigers* runs a number of train units. The units need regular preventive maintenance checks, say, after every 30,000 km. Units that travelled that distance must go to the maintenance facility. For this, the rolling stock schedule has to be modified so that the units, that need maintenance, arrive in time at a maintenance facility while still being deployed for timetable services. As yet NS Reizigers does not use a decision support system for maintenance routing. It is the goal of this research to provide mathematical programming models and solution methods that can serve as basis for such a decision support system.

Maróti and Kroon [11] describe an integer programming model for maintenance routing. Despite promising computational results, the model also has a certain drawback. The issue is that it requires many details about the shunting possibilities; details that are quite difficult to obtain in practice. Therefore we present in this paper a model that requires less input data. Although it is a bit further from reality, we believe that it is detailed enough to produce useful results.

We describe the maintenance routing problem in Section 2. Our new integer programming model is presented in Section 3. In Section 4 we prove some complexity results. Our computational results are in Section 5.

2 The maintenance routing problem

2.1 Maintenance planning in practice

The Logistics Department of NS Reizigers determines the rolling stock schedule. We call this the *regular plan*. It contains the rolling stock movements between different stations but does not involve movements inside stations. These are described by shunting plans created by local shunting crew. A smallest indivisible movement in the regular plan to be carried out by a single train unit is called a *task*. So, a movement carried out by a train with more than one unit corresponds to several tasks. Two tasks form a *regular transition* if a train unit carries them out consecutively in the regular plan.

Some of the tasks in the regular plan are *maintenance tasks*. If a train unit carries out such a task, it undergoes a maintenance check. (Maintenance checks can only take place at specialised stations.) The regular plan assigns maintenance tasks arbitrarily, irrespective of which unit needs to be maintained and when. The actual maintenance routing is handled shortly before execution. NS Reizigers uses a fixed planning horizon for maintenance routing, usually 3 days. Input is the regular plan and a list of *urgent* train units: those that need a maintenance check within the planning horizon. They may have different urgencies, expressed by a deadline (for instance saying that the unit should be maintained within 2 days). The part of the regular plan that lies within the planning horizon must be adjusted so that each urgent unit is routed to a maintenance facility while carrying out timetable services. This maintenance planning process is carried out daily, with a rolling horizon.

The maintenance planners work as follows. They consider a (small) number of train units that are at some moment at the same station and replace their regular transitions by new transitions. The purpose of these changes is to lead the urgent units in time to a maintenance task. Figure 1 shows a small example. Thick lines represent tasks, dashed lines regular transitions, and arrows new transitions. The new duty of the urgent unit contains a maintenance task.

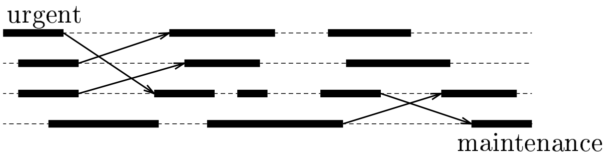

Figure 1

Changes may require shunting at the stations. The maintenance planners do not have direct access to shunting options. Therefore they check with the local shunting crew if the preferred changes can be implemented in practice. If not, maintenance planners must come up with other changes and check them again. Communication between the maintenance planners and the local shunting crew takes time, so the maintenance routing planners cannot test too many possibilities. Therefore they are usually satisfied with the first acceptable solution found.

2.2 Previous work

Several related papers can be found in the literature. Lingaya et al. [10] describe a model for operational management of locomotive-hauled railway cars. They seek for a maximum expected profit schedule that satisfies various constraints, among them also maintenance requirements.

Some other papers focus on aircraft maintenance routing. For example, Barnhart et al. [2], Clarke et al. [5], Feo and Bard [8], Gopalan and Talluri [9] and Talluri [13] deal with this problem. Anderegg et al. [1] describe models for railway applications that are similar to the aircraft routing models.

These models cannot be applied directly to the maintenance routing at NS Reizigers. The models consider maintenance routing as a part of the medium- and long-term vehicle scheduling problem, and specify the tasks of the individual vehicles during the entire planning period. The models above create the vehicle schedule from scratch, without taking shunting issues into account. Therefore in our railway application there is no guarantee that the output of these models can be carried out in practice.

Furthermore, disturbances and delays make it unlikely that the rolling stock schedule can be carried out exactly as planned in a period longer than a couple of days. On the other hand, the Dutch railway network contains a large number of frequently operated relatively short train lines. This provides many exchanging possibilities for the train units, therefore a couple of days is mostly enough to route an urgent train unit to a maintenance facility. This explains why the maintenance planners consider planning periods of length 1–3 days, and only a small number of train units to be routed.

2.3 Our approach

The bottleneck in maintenance routing is shunting: planners should make sure that the desired new transitions can in fact be carried out. A decision support system for maintenance routing should be able to judge the difficulty of the considered solutions. However, generating shunting plans is difficult, even for just a single middle-sized station. Thus we cannot instantly check if it is possible to carry out the shunting operations involved with a change. What we can do is estimate shunting costs.

We know that the regular plan allows a shunting plan. Therefore if we apply only a small number of changes, the modified plan will be close to the regular plan and we may hope that the local shunting crew will be able to carry out the new plan.

Maróti and Kroon [11] propose the "Scenario Model" for the maintenance routing problem. It is based on scenarios; a scenario is a group of transitions for which the whole shunting process can be carried out. The Scenario Model requires many details on shunting, more than is likely to be available in practice. Therefore we describe here another model, the "Transition Model". It is a variant of the multi-depot vehicle routing problem, studied in detail e.g. in Desrosiers et al. [7] and Toth and Vigo [14]. Our results and methods are related to those in Carraresi and Gallo [4] and in Bertossi et al. [3].

The Transition Model incorporates no other information on shunting possibilities than a description of which new transitions are allowed; the sole criterion for that is whether there is enough time to carry out the shunting operations involved with the particular transition without causing delays. We then assign a cost value to every allowed new transition and define the cost of a solution as a simple function of the costs of the new transitions in the solution. As the Transition Model is less involved than the Scenario Model, it may produce less acceptable solutions. However, its input data are a lot easier to obtain and in the setting of an interactive decision support system the cost function can be tuned based on rejections or approvals of local shunting crew. Recall that whatever method for maintenance planning is used, each solution needs approval of the shunting crew before it can be implemented anyway.

3 The Transition Model

3.1 The input: maintenance routing graphs

We formalise the input for our model. A *timetable* is a set V of *tasks* together with four integer valued functions $\mathsf{Dep\_station}(), \mathsf{Arr\_station}(), \mathsf{Dep\_time}()$ and $\mathsf{Arr\_time}()$ on V such that $\mathsf{Dep\_time}(v) \leqslant \mathsf{Arr\_time}(v)$ for each $v \in V$. A *regular duty* is a sequence $(v_1, \ldots, v_l)$ of tasks with $\mathsf{Arr\_station}(v_i) = \mathsf{Dep\_station}(v_{i+1})$ and $\mathsf{Arr\_time}(v_i) < \mathsf{Dep\_time}(v_{i+1})$ for each $i = 1, \ldots, l-1$. A *regular plan* is a partition of V into regular duties. Each pair of consequent tasks in such a regular plan is a *regular transition*. In a railway application, the regular duties are the sequences of tasks assigned to a particular train unit. A *candidate transition* is a pair of tasks (v, w) such that it is not a regular transition, and such that $\mathsf{Arr\_station}(v) = \mathsf{Dep\_station}(w)$ and $\mathsf{Dep\_time}(w) \geqslant \mathsf{Arr\_time}(v) + \mathsf{Buffer\_time}$ where $\mathsf{Buffer\_time}$ is a given number.

All graphs in this paper are directed. A *source* in a graph is a node with no in-going arcs and a *sink* is a node with no out-going arcs. The set of tasks V of a timetable and the set of regular and candidate transitions form an acyclic graph. (A graph is acyclic if it does not contain directed cycles.) Any graph that can be obtained in this way is called a *maintenance routing graph* (or *MR-graph*).

An *urgency* in an MR-graph $G = (V, A)$ is a function M on a subset U of the set of sources in G with $M(u) \subseteq V$ for each $u \in U$; members of U are *urgent nodes*. In a maintenance routing application a node is urgent if it is the earliest task of one of the train units that must reach a maintenance facility in the planning period; for $u \in U$, $M(u)$ is the set of maintenance tasks that can be assigned to u in order to satisfy the maintenance requirement of the corresponding train unit.

In real world maintenance routing, we may allow the buffer time to vary from arc to arc, but here an MR-graph always has a uniform buffer time. It also happens in real world maintenance routing that some practical instances can only be solved when allowing non-scheduled train movements without passengers. If desired we can model such movements by transitions between tasks that arrive and depart at different stations. However, such more general transitions are not part of our definition of MR-graphs.

3.2 The model

From the timetable and Buffer_time we construct the corresponding MR-graph.

The sequences of tasks that can be assigned to a single train unit over a whole planning period are precisely the directed paths in the corresponding MR-graph that connect a source to a sink. A *path cover* in an acyclic graph $G = (V, A)$ is a set of arcs that forms a collection of node disjoint paths such that the paths cover V and that each path connects a source of G with a sink of G; the regular transitions in a maintenance routing graph form a path cover. So, the maintenance routing problem amounts to finding a path cover such that the path in the path cover that starts at urgent node u contains a node in $M(u)$.

In order to evaluate shunting efforts, we calculate for each transition a cost value estimating how difficult it is to use it for a train unit (see Section 5.1 for a short discussion on how we calculate these estimates.) We measure the overall shunting difficulty of a solution by the sum of the individual transition costs.

With these data, the maintenance routing problem becomes the following integer multicommodity flow model; we define it for general acyclic graphs for later reference.

Transition Model: *Given an acyclic graph $G = (V, A)$, a set U of sources and an urgency M on U, as well as a cost function $c \colon A \to \mathbb{R}_+$, find a path cover Z minimising*

$$\sum_{a \in Z} c(a)$$

such that for each $u \in U$, the path starting at u contains a member of $M(u)$.

We formulate this as a binary linear program. The variables are:

$$z\colon A \to \{0,1\}, \text{ and for each } u \in U,\, x_u\colon A \to \{0,1\}. \tag{1}$$

The functions x_u are network flows indicating the paths of the urgent units, while z is the characteristic function of a path cover.

The constraints are:

1. flow conservation for z at every node v except for the sources and sinks; each node has throughput 1 in z (δ^{in} and δ^{out} denote the set of entering and leaving arcs, respectively):

$$z\left(\delta^{\text{in}}(v)\right) = z\left(\delta^{\text{out}}(v)\right) = 1; \tag{2}$$

2. flow conservation for each flow x_u at every node v except if $v = u$ or $v \in M(u)$:

$$x_u\left(\delta^{\text{in}}(v)\right) = x_u\left(\delta^{\text{out}}(v)\right); \tag{3}$$

3. the flows x_u must form a subsystem of the flow z:

$$\text{for every arc } a\text{: } \sum_{u \in U} x_u(a) \leqslant z(a); \tag{4}$$

4. every node $u \in U$ has out-flow 1 in x_u; every source v' has out-flow 1 in z:

$$x_u\left(\delta^{\text{out}}(u)\right) = 1; \qquad z\left(\delta^{\text{out}}(v')\right) = 1; \tag{5}$$

5. a maintenance task v has neither any out-going flow in any of the flows x_u, nor any in-going flow in a flow it is not allowed for:

$$\sum_{u \in U} x_u(\delta^{\text{out}}(v)) = 0; \qquad \sum_{u \in U\colon\, v \notin M(u)} x_u\left(\delta^{\text{in}}(v)\right) = 0. \tag{6}$$

The objective function is:

$$\text{minimise} \sum_{a \in A} c(a) \cdot z(a). \tag{7}$$

Reducing the problem size

An arc in an acyclic graph G is said to be *covered* if it is in at least one path cover of G. Many transitions in a maintenance routing problem will have a very long time span; such transitions will typically not be covered in the corresponding MR-graph, so they make the model larger than necessary. We call an acyclic graph *reduced* if each arc is covered. Finding a path cover and determining which arcs are covered are network flow problems, so we can reduce a graph in polynomial time. Actually when a path cover is given, as in the case in MR-graphs, one can determine the covered arcs in linear time. In spite of the fact that this is standard network flow theory (for details see e.g. Schrijver [12]), we explain it as we need the construction later. Given an acyclic graph $G = (V, A)$ with a path cover R, construct an auxiliary graph $\widehat{G} = (\widehat{V}, \widehat{A})$ as follows: for each $v \in V$ there are two distinct nodes v_{out} and v_{in} in $\widehat{V}$. For each candidate transition $vw \in A$, there is an arc $v_{\text{out}}w_{\text{in}}$ in $\widehat{A}$, and for each regular transition $vw \in A$, there is an arc $w_{\text{in}}v_{\text{out}}$ in $\widehat{A}$. A small example is given in Figure 2. Now an arc in G is covered if and only if it is either in R or the corresponding arc in $\widehat{G}$ is in a directed circuit. Therefore identifying the covered arcs amounts to finding the strong components of $\widehat{G}$ which can be done in linear time, indeed.

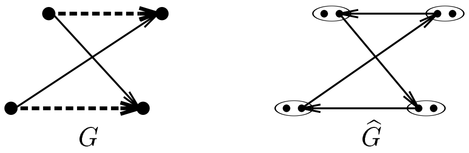

Figure 2: The auxiliary graph. (The dashed arcs are regular.)

4 Complexity results

In this section we derive complexity results for the Transition Model. We prove that the feasibility problem is NP-complete even when there is only a single urgent node. When the number of sources is fixed, an optimal solution to the Transition Model can be found in polynomial time with dynamic programming irrespective the number of urgent nodes. For MR-graphs with one urgent node, the feasibility version is polynomially solvable, but the optimization problem remains NP-hard. The following table summarizes these results of this section.

Complexity of the Transition Model	Feasibility	Optimising
Arbitrary acyclic graphs, $\|U\| = 1$	NP-hard	NP-hard
Arbitrary acyclic graphs, fixed number of sources	P	P
MR-graphs, $\|U\| = 1$	P	NP-hard

4.1 NP-hardness in arbitrary acyclic graphs

Theorem 1. *It is NP-complete to decide if an acyclic graph has a path cover such that one of its paths connects a given source with a given sink.*

Proof. We reduce the satisfiability problem to our problem. Let $x_1, \ldots, x_n$ be Boolean variables, let $\varepsilon \in \{\pm 1\}$ and denote

$$x_j^\varepsilon = \begin{cases} x_j & \text{if } \varepsilon = +1, \\ \neg x_j & \text{if } \varepsilon = -1. \end{cases}$$

Consider the disjunctive normal form

$$\varphi = \bigwedge_{i=1}^{k} \left(x_{i_1}^{\varepsilon_{i_1}} \vee x_{i_2}^{\varepsilon_{i_2}} \vee \cdots \vee x_{i_{m(i)}}^{\varepsilon_{i_{m(i)}}} \right)$$

It is well-known that it is NP-complete to decide whether or not we can assign values to the variables that make φ true (Cook [6]). We start with an empty graph. We insert two nodes a_j, b_j for each variable x_j and draw an arc $a_j b_j$. For each $i = 0, \ldots, k$, we create three nodes s_i, c_i, d_i, and insert the arcs $c_i s_i$, $s_i d_i$ and $c_i d_i$. Moreover, we insert the arc $c_k d_0$. Let $s = c_0$ and $t = d_k$.

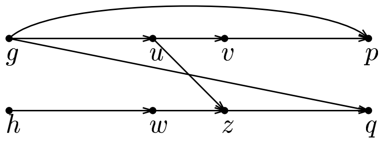

Figure 3

For the occurrence of a variable x_j (either as x_j or as $\neg x_j$) in clause i, we create a "box" as shown in Figure 3. We create 8 new nodes u, v, w, z, g, h, p, q and draw the arcs $uv, uz, wz, gu, hw, vp, zq, gp$, and gq. We insert the arc $s_{i-1}u$. Moreover, we draw the arc vs_i if the non-negated x_j occurs in clause i, and we draw the arc zs_i if $\neg x_j$ occurs in clause i.

We order the occurrences of x_j by increasing clause index. We draw an arc from a_j to the v-node of the first x_j-box. We insert an arc from the w-node of any x_j-box to the v-node of the next x_j-box. Finally, we join the w-node of the last x_j-box to b_j. Then we obtain a picture as in Figure 4. Note that the arcs a_jb_j, c_is_i, s_id_i, gu, uv, vp, hw, wz, zq form a path cover.

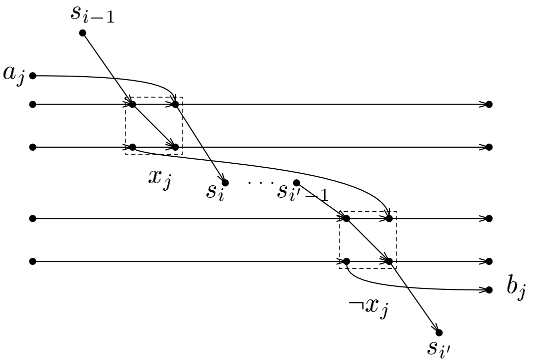

Figure 4

We claim that a path cover connecting s with t exists in this graph if and only if φ is satisfiable.

Suppose there exists such a path cover. Observe that for any pair of occurrences of x_j, the paths use in both boxes either the uv-arc or the uz-arc. This follows easily for two consecutive occurrences (i.e. for those that are joint by a wv-arc) using the fact that the v-nodes and the z-nodes have exactly two entering arcs while the u-nodes and the w-nodes have exactly two leaving arcs. Set x_j true if the uv-arcs are used in the x_j-boxes, otherwise set it false. Observe that every $s-t$ path in the graph contains all the nodes s_i. Then the $s-t$-path proves that every clause is satisfied.

Conversely, suppose that the variables have an assignment satisfying φ. For any clause i, label any of the literals that makes it true.

If x_j has value true, select the arc a_jb_j, as well as the uv, wz, hw and zq arcs of all the x_j-boxes. If an occurrence of x_j is labelled for clause i, select the $s_{i-1}u$, vs_i and gp-arcs of this box, otherwise select the gu and vp-arcs.

If x_j has value false, select the uz, hw and vp-arcs of the x_j-boxes, the wv-arcs between the x_j-boxes as well as the arcs connecting the nodes a_j and b_j to the x_j-boxes. If the occurrence is labelled for clause i, select the arcs $s_{i-1}u$, zs_i and gq in this x_j-box, otherwise select the arcs gu and zq.

Finally, select the arcs c_id_i for $i = 1, \ldots, k-1$ and the arc c_kd_0. Then the selected arcs form a path cover connecting s with t. $\square$

4.2 Bounded number of train units

In the maintenance routing application, the number of sources equals the number of train units of the given type. We can consider it as a constant, while the length of the planning period (therefore the number of nodes and arcs in the graph) is the varying parameter. Then the Transition Model can be solved in polynomial time.

Theorem 2. *For any acyclic graph $G = (V, A)$, the optimisation problem (1) – (7) can be solved in $O\left((|V||A|)^{O(d)}\right)$ time where d is the number of sources.*

Proof. Let $n = |V|$, $m = |A|$, and let $k = |U|$. We assume that the sources are $S = \{1, \ldots, d\}$ and that $U = \{1, \ldots, k\}$. The optimisation problem can be reduced to a shortest path problem in an acyclic graph $G' = (V', A')$ with at most $(2mn)^d$ nodes and at most $(2mn)^{2d}$ arcs. The sketch of the construction is as follows.

We colour the nodes of G red. By subdividing arcs, we can transform G into a graph where the node set is partitioned into disjunct subsets $V_0, \ldots, V_\ell$ such that each arc a connects V_i to V_{i+1} for some index i. We also assume that the arcs between V_0 and V_1 form a matching of size d. The enlarged graph has at most mn nodes and at most mn arcs. We colour the newly created nodes black. Then a path cover in the original graph corresponds to a system of paths in the enlarged graph covering all the red nodes.

We create a member of V' for every $(d+k)$-tuple $v' = (a_1, \ldots, a_d, \varepsilon_1, \ldots, \varepsilon_k)$ where (i) the arcs a_i lie between the subsets V_{t-1} and V_t for some index t, (ii) the arcs a_i cover all the red nodes in $V_{t-1} \cup V_t$, and (iii) $\varepsilon_j \in \{0, 1\}$ for each $j = 1, \ldots, k$. The arcs a_i indicate the routes of the train units through the network, while ε_j encodes whether urgent unit j has been maintained so far.

Let s' be the node in V' with arcs a_i between V_0 and V_1 and with $\varepsilon_1 = \cdots = \varepsilon_k = 0$. Let T' contain all nodes $v' \in V'$ such that the arcs a_i lie in v' between the subsets $V_{\ell-1}$ and V_ℓ, and such that $\varepsilon_j = 1$ for each j.

Let $v' = (a_1, \ldots, a_d, \varepsilon_1, \ldots, \varepsilon_k)$ and $w' = (b_1, \ldots, b_d, \zeta_1, \ldots, \zeta_k)$ nodes of G'. Let a pair (v', w') be an arc in G' if

- for every i, the head of the arc a_i is the tail of the arc b_i;
- the paths (a_i, b_i) for $i = 1, \ldots, d$ are node disjoint;

- for every $j = 1, \ldots, k$, $\zeta_j = 1$ if and only if $\varepsilon_j = 1$ or if the head of the arc b_j is an allowed maintenance node for urgent unit j.

We define the cost of the arc (v', w') as $\sum_{i=1}^{d} c(b_i)$.

Then a system of d disjoint paths in G covering all the red nodes and satisfying all the maintenance requirements corresponds to an $s' - T'$-path in G' with the same cost value, and vice versa. □

4.3 MR-graphs with one urgent unit

We have seen in Section 4.1 that the feasibility problem with a single urgent node is NP-complete in artificially constructed graphs. It is, however, easy to solve in MR-graphs.

Theorem 3. *Let $G = (V, A)$ be an MR-graph and let A_0 be the set of covered arcs. Moreover, let $s, t \in V$ and suppose that P is an $s - t$-path in $G_0 = (V, A_0)$ with inclusionwise minimal node set. Then there exists a path cover containing P.*

Proof. Let R denote the set of regular transitions. Consider the auxiliary graph $\widehat{G}$ of G as defined in Section 3.2. Let W be the set of nodes v_{out} with v in P and $v \neq t$, and let W_N be the set of nodes $v_{\text{out}} \in W$ with $vw \in P \setminus R$.

We show that no pair of different nodes $v_{\text{out}} \in W_N$ and $v'_{\text{out}} \in W$ lie in the same strong component of $\widehat{G}$.

Suppose the contrary. Let vw and $v'w'$ be arcs in path P. We may assume that v' lies on P between v and t, the other case being analogous. Let K be the strong component of $\widehat{G}$ that contains v_{out} and v'_{out}. Then $w_{\text{in}} \in K$ since $v_{\text{out}}w_{\text{in}}$ belongs to a directed circuit in $\widehat{G}$. Moreover, if $v'w'$ is a regular transition, then every $v_{\text{out}} - v'_{\text{out}}$-path contains the node w'_{in} since $w'_{\text{in}}v'_{\text{out}}$ is the only arc that enters v'_{out}. If, however, $v'w'$ is a candidate transition, then $v'_{\text{out}}w'_{\text{in}}$ is contained in a circuit of $\widehat{G}$. In both cases we have $w'_{\text{in}} \in K$. In particular, $\widehat{G}$ contains a $w'_{\text{in}} - v_{\text{out}}$-path $\widehat{P}$.

For every arc $x_{\text{out}}y_{\text{in}} \in \widehat{A}$, we have $\mathsf{Arr\_station}(x) = \mathsf{Dep\_station}(y)$, and for every arc $x_{\text{in}}y_{\text{out}} \in \widehat{A}$, we have $\mathsf{Dep\_station}(x) = \mathsf{Arr\_station}(y)$. Therefore $\mathsf{Arr\_station}(v) = \mathsf{Dep\_station}(w')$. Moreover,

$$\mathsf{Arr\_time}(v) \leqslant \mathsf{Dep\_time}(w) - \mathsf{Buffer\_time} \leqslant \mathsf{Dep\_time}(w') - \mathsf{Buffer\_time}.$$

Thus vw' is either a regular transition (and thus it is covered) or a candidate transition. In the latter case, $\widehat{P} \cup \{v_{\text{out}}w'_{\text{in}}\}$ is a circuit in $\widehat{G}$. Therefore vw' is covered. This contradicts the minimality of the path P. Therefore v_{out} and v'_{out} cannot lie in the same strong component of $\widehat{G}$.

For every arc $vw \in P \setminus R$, fix a directed circuit $C_{v,w}$ in $\widehat{G}$ containing the arc $v_{\text{out}}w_{\text{in}}$. Let C be the set of those edges in G that correspond in $\widehat{G}$ to an edge in $\bigcup\{C_{v,w} : vw \in P \setminus R\}$. Then the circuits $C_{v,w}$ are node disjoint, therefore $(R \setminus C) \cup (C \setminus R)$ is a path cover of G. Moreover, no arc in $P \cap R$ belongs to C, therefore $(R \setminus C) \cup (C \setminus R)$ contains P. □

This implies

Corollary 4. *The feasibility problem of the Transition Model in an MR-graph with one urgent node (train unit) can be solved in polynomial time.*

The NP-hardness of the feasibility problem in arbitrary acyclic graphs is partly caused by sparsity of acyclic graphs that are not MR-graphs. The proof above is based on the property of MR-graphs that the existence of a certain pair of arcs vw and $v'w'$ implies the existence of the arc vw'.

The algorithm for the feasibility problem does not extend to the optimisation version of the single urgent unit case. Reason is that the cost of a path serving the urgent unit does not reflect the cost of a corresponding solution, as also non-urgent units will use new transitions, and these do contribute to the cost of the solution as well. Actually, finding an optimal solution to the Transition Model with one urgent node is also NP-hard in MR-graphs.

Theorem 5. *It is NP-hard to find an optimal solution to the transition model on MR-graphs with one urgent train unit, one maintenance task and* $\{0,1\}$*-valued cost function.*

Proof. First we show that each acyclic directed graph G with a path cover R can be extended to an MR-graph G'. Indeed, there is only one station, and the arcs in R are the regular transitions. We define departure and arrival times as follows. Consider a topological order of G, that is, an order on $V = \{v_1, \dots, v_n\}$ such that $v_iv_j \in A$ implies $i < j$. Let $\mathsf{Dep\_time}(v_i) = 2i - 1$, $\mathsf{Arr\_time}(v_i) = 2i$, and let $\mathsf{Buffer\_time} = 0$. Then every arc $vw \in A$ is a regular or candidate transition. We insert an arc for all the other pairs (v, w) that form a candidate transition according to the definition in Section 3.1. The obtained graph G' is an MR-graph.

We apply this to the graph G constructed in the proof of Theorem 1. We set the cost value 0 for the original arcs and we set the cost value 1 for the newly inserted arcs. Then deciding whether or not the optimisation problem in the extended graph G' has optimum value zero amounts to solving the feasibility problem in G. □

5 Computational results

5.1 The cost values

The success of the model in practical applications highly depends on the choice of the cost function. We assume that the regular plan has a corresponding shunting plan that can be carried out in practice. The cost values measure how much the regular plan has to be modified in order to carry out the given transitions. We determined these transition costs from the timetable data, by taking the following factors into account:

- the positions of the units in the train compositions;
- the time difference between the tasks;
- information about the infrastructure like physical distance between the arrival and departure tracks;
- the station where the transition takes place; stations with heavy traffic provide less changing possibilities;
- the time that the transition is carried out; changes during rush hours are hard, while changes at night are typically easy.

5.2 Experiments

We used the train units of type 'Sprinter' as test example. NS Reizigers runs 47 train units of this type. The dimensions of the problem are given in Table 1. We can see that the reduction step described in Section 3 results in decreasing the number of arcs dramatically. Without this reduction, the binary linear programs could not be used at all. There are two maintenance tasks on every day. For simplicity, we set Buffer_time uniformly to 10 minutes, based on discussions with the planners.

We considered planning horizons of 2 up to 5 days. We generated 1000 instances per planning horizon. We selected urgent train units randomly: each with a fixed maintenance day, 2 urgent units per day in the planning period.

Planning period	2 days	3 days	4 days	5 days
# of tasks	1,196	1,762	2,326	2,709
# of transitions	98,035	198,614	332,031	508,853
# of covered transitions	11,394	17,067	22,830	28,570

Table 1

Planning period	2 days	3 days	4 days	5 days
# urgent units	4	6	8	10
# variables in IP	28,453	51,193	79,786	114,096
# constraints in IP	15,337	23,816	32,947	42,672
Average solving time	4.31 s	8.92 s	19.69 s	43.03 s
Max. solving time	6 s	12 s	138 s	486 s

Table 2

Since the maintenance tasks start at the same time at the same location on every day, the integer program requires only one flow x_u for every possible deadline.

For our computations we used the modelling software ILOG OPL Studio 3.7 and the integer program solver ILOG CPLEX 9.0 on a PC with an Intel P4 3.0 GHz processor and 512 MB internal memory. Our results are summarised in Table 2.

Despite the large sizes of the integer linear programs, optimal solutions could be found in reasonably short time. It indicates that, although one can create difficult instances, the network structure of the problem allows a tight linear programming description for our test instances. Then, after a small number of branchings, CPLEX was able to construct an optimal integer solution.

In order to compare the Transition Model to the more involved Scenario Model that is described in [11], we derived the transition costs from the Scenario Model and modified slightly the objective function. Then the Transition Model becomes a relaxation of the Scenario Model. When solving the same instances in both models, the difference between the objective values turned out to be small: in all cases less than the cost of just one considerably expensive arc. That is, the solutions to both models hardly differ, although the Transition Model takes much less explicit details about shunting into account.

The practical value of the solutions, however, cannot be evaluated easily. A small number of examples has been discussed with the maintenance planners. They found the solutions satisfactory, although this does not mean that the local shunting crew would accept all these solutions. In this respect, the solutions generated by the model are comparable to the solutions generated by the maintenance planners themselves.

6 Summary

In this paper we described the Transition Model for routing passenger train units towards a maintenance facility. We analysed the computational complexity of this maintenance routing problem. In our computational tests, optimal solutions could be found in reasonably short running times by the commercial MIP solver CPLEX.

The Transition Model takes shunting into account via the arc cost values, resulting in an intuitive model which is much less involved than the Scenario Model described by Maróti and Kroon [11]. An advantage of the Transition Model is the fact that the required data can be collected more easily than in the case of the Scenario Model. A disadvantage may be that the probability of obtaining solutions that are feasible in the model but infeasible in practice is somewhat higher. Our computational experience showed that the solutions obtained from the two models do not differ much. Based on discussions with the maintenance planners in practice, we believe that the Transition Model is an appropriate basis for a decision support system for maintenance planning.

Acknowledgement

This research was supported by the Human Potential Programme of the European Union under contract no. HPRN-CT-1999-00104 (AMORE).

We would like to thank Bert Gerards and Lex Schrijver for their valuable comments on this paper.

References

[1] L. Anderegg, S. Eidenbenz, M. Gantenbein, Ch. Stamm, D.S. Taylor, B. Weber, P. Widmayer, Train routing algorithms: Concepts, design choices, and practical considerations, *Proceedings of ALENEX'03*, `www.siam.org/meetings/alenex03/Abstracts/LAnderegg3.pdf`.

[2] C. Barnhart, N.L. Boland, L.W. Clarke, E. Johnson, G.L. Nemhauser, R.G. Shenoi, Flight string models for aircraft fleeting and routing, *Transportation Science* 32 No. 3 (1998), pp. 208–220.

[3] A.A. Bertossi, P. Carraresi, G. Gallo, On some matching problems arising in vehicle scheduling models, *Networks* 17 (1987), pp. 271–281.

[4] P. Carraresi, G. Gallo, Network models for vehicle and crew scheduling, *European Journal of Operational Research* 16 (1984), pp. 139–151.

[5] L. Clarke, E. Johnson, G. Nemhauser, Z. Zhu, The aircraft rotation problem, *Annals of Operations Research* 69 (1997), pp. 33–46.

[6] S.A. Cook, The complexity of theorem-proving procedures, in: *Conference Record of Third Annual ACM Symposium on Theory of Computing* The Association for Computing Machinery, New York, 1971, pp 151–158.

[7] J. Desrosiers, Y. Dumas, M.M. Solomon, F. Soumis, Time constrained routing and scheduling, In: *Handbooks in OR and MS*, vol. 8, pp. 35–139, Elsevier, 1995.

[8] T.A. Feo, J.F. Bard, Flight scheduling and maintenance base planning, *Management Science* 35 No. 12 (1989), pp. 1514–1532.

[9] R. Gopalan, K.T. Talluri, The aircraft maintenance routing problem, *Operations Research* 46 No. 2 (1998), pp. 260–271.

[10] N. Lingaya, J.F. Cordeau, G. Desaulniers, J. Desrosiers, F. Soumis, Operational car assignment at VIA Rail Canada, *Transportation Research Part B* 36 (2002) pp. 755–778.

[11] G. Maróti, L.G. Kroon, Maintenance routing for train units: the Scenario Model, CWI Research Report, 2004

[12] A. Schrijver, Combinatorial Optimization: Polyhedra and efficiency, Springer Verlag, Berlin, Heidelberg, 2003.

[13] K.T. Talluri, The four-day aircraft maintenance routing problem, *Transportation Science* 32 No. 1 (1998), pp. 43–53.

[14] P. Toth, D. Vigo, The vehicle routing problem, *SIAM Monographs on Discrete Mathematics and Applications*, 2002.