

Theory and Methodology

Routing trains through railway stations: complexity issues

Leo G. Kroon*, H. Edwin Romeijn, Peter J. Zwaneveld

Erasmus University Rotterdam, Rotterdam School of Management, P.O. Box 1738, NL-3000 DR Rotterdam, The Netherlands

Received March 1995; revised November 1995

Abstract

In this paper we consider the problem of routing trains through railway stations. This problem occurs as a subproblem in the project DONS that is currently being carried out under the supervision of Railned and Netherlands Railways. The project DONS involves the determination of the required future capacity of the Dutch railway infrastructure.

In this paper we focus on the computational complexity of the problem of routing trains through railway stations. After an extensive description of the problem, we show that only a subset of the sections and routes of a railway station needs to be taken into account. Then we show that the routing problem is NP-complete as soon as each train has three routing possibilities. However, if each train has only two routing possibilities, then the problem can be solved in an amount of time that is polynomial in the number of trains. Furthermore, if the layout of the railway station is fixed, then the latter is also the case for the problem of finding an assignment of a maximum number of trains to routes that is feasible from a safety point of view. This result can be extended to the case where coupling and uncoupling of trains, certain service considerations, and a cyclic timetable have to be taken into account. © 1997 Elsevier Science B.V.

Keywords: Railway transportation; Timetabling; Complexity theory; Dynamic programming

1. Introduction

This paper is motivated by the project DONS that is carried out under the supervision of the Dutch organization Railned and Netherlands Railways.¹ It is one of the strategic tasks of Railned to assess the required future capacity of the Dutch railway infrastructure. Currently, this task is accomplished by generating a number of plausible future timetables, and by checking whether these timetables are feasible, given certain scenarios for the future Dutch railway infrastructure.

Since generating a timetable and checking its feasibility requires an enormous amount of time when carried out manually, the project DONS was initiated recently. The objective of this project is to develop a Decision Support System (DSS) that will assist the planners of Railned in their capacity planning work. Systems and models with similar aims are described by Bourachot [2] and Carey [3].

The DSS to be developed is also called DONS, and will contain at least two complementary modules. The first module, called CADANS, will assist the planners in generating cyclic hourly timetables. The second module, called STATIONS, will assist the planners in checking whether a timetable generated by CADANS is feasible with respect to the routing of trains through railway stations, and in generating operating plans for

* Corresponding author. E-mail: lkroon@fac.fbk.eur.nl.

¹ This research is sponsored by Railned and Netherlands Railways.

the routing of trains through railway stations.

The module CADANS is being developed by Schrijver and Steenbeek [23]. This module takes into account: (i) desired frequencies of lines, (ii) leg constraints, (iii) market constraints, and (iv) infra constraints. The leg constraints specify the travel times between the railway stations, the market constraints specify minimum and maximum transfer times between certain pairs of trains at a number of railway stations, and the infra constraints guarantee that there is a certain minimum buffer time between two trains passing a common section. A timetable generated by CADANS is cyclic with a cycle length of one hour. That is, such a timetable covers a time period of one hour, and after one hour the same pattern of arrivals and departures of trains is repeated.

CADANS considers the railway infrastructure only from a global point of view, and neglects the detailed layout of the railway network within the railway stations. Therefore it may happen that a timetable generated by CADANS is feasible with respect to the railway network *between* the railway stations, but turns out to be infeasible if one also considers the detailed layout of the railway network *within* the railway stations. It is the aim of the module STATIONS to assist the planners in checking whether a timetable generated by CADANS is feasible with respect to the routing of the trains through the railway stations, given the layout of the railway stations. If a routing for all trains through the railway stations does not exist, then the blocking trains should be identified. In this case, the module STATIONS is expected to provide suggestions for the modification of the arrival and departure times of these blocking trains.

The authors of this paper are involved in the development of STATIONS. For a more detailed description of the corresponding project we refer to Kroon and Zwaneveld [19]. In the current paper we describe a number of complexity issues related to the feasibility problem that has to be solved by STATIONS. This paper is complementary to the paper by Zwaneveld et al. [25], which focuses on the algorithms that are incorporated into the module STATIONS. These algorithms are based on the formulation of the problem as a Node Packing Problem, and on the application of dominance rules, valid inequalities, heuristics, and a branch & cut procedure. The current paper covers a number of theoretical aspects, since it considers the

computational complexity of several variants of the routing problem of trains through railway stations. However, some of these theoretical aspects, such as the preprocessing rules, are directly useful in practice.

The organization of this paper is as follows. Section 2 describes the feasibility problem of routing trains through railway stations in more detail. In Section 3 we introduce the notation that is used in this paper. In Section 4 we show how the size of an instance of the feasibility problem may be reduced a priori by applying specific preprocessing rules. Section 5 contains the formulation of the problem as an integer linear program, and describes the relationship between the routing problem and the Fixed Interval Scheduling Problem. Next, in Section 6, we focus on the computational complexity of a number of variants of the problem that has to be solved by STATIONS, and in Section 7 we show that these results are also valid in case of a cyclic timetable. The paper is finished in Section 8 with some conclusions and subjects for further research.

2. Problem description

The problem of routing trains through railway stations that we address in this paper can be stated as follows. Given the layout of a railway station, the arrival and departure times, as well as the arrival and departure directions of a number of trains, is it possible to route these trains through the railway station such that no pair of trains is conflicting, such that trains can be coupled or uncoupled if necessary, and such that a number of service constraints are satisfied? We will call this problem the *feasibility problem*. The characteristics of the problem that we describe in this paper pertain to the railway system in The Netherlands, but are very similar to most European railway systems.

We start out with a more detailed description of this feasibility problem: a railway station can be entered by a train from a number of *entering points*, and it can be left through a number of *leaving points*. In general, each entering point can also serve as a leaving point, and vice versa. Furthermore, each of these points corresponds to a *direction of travel*. For example, the directions of travel of the Dutch railway station Zwolle are Almelo, Amersfoort, Deventer, Kampen, and Meppel (cf. Fig. 1). The railway network

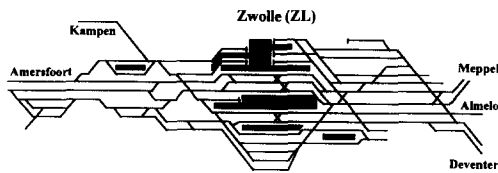


Fig. 1. Infrastructure of railway station Zwolle.

outside the entering and leaving points is not relevant for the feasibility problem.

A railway station consists of *platforms* and of a large number of *track sections*. An *inbound route* is a sequence of sections linking an entering point to a platform section next to a platform. The corresponding platform section is also part of the inbound route. Similarly, an *outbound route* is a sequence of sections linking a platform to a leaving point. A *complete route* is either a combination of an inbound and an outbound route using the same platform, or a sequence of sections connecting an entering point to a leaving point, bypassing the platforms. There will often be many different routes between a given pair of entering and leaving points, and even several different routes that use the same platform.

The *arrival time* of a train is the time at which the train stops at a platform, after traveling along an inbound route. Similarly, the *departure time* of a train is the time at which the train starts to leave the railway station along an outbound route. As was described in Section 1, the arrival and departure times of trains are generated by CADANS. Therefore we will take the arrival and departure times as given and, if at all possible, determine feasible routes for each of the trains passing through the railway station.

Clearly, the routing of one train depends on the routing of others. Most importantly, the safety rules of Netherlands Railways dictate the following route reservation procedure. As soon as a train arrives at its entering point of the railway station, it reserves an appropriate inbound route leading to a platform. Since any track section can only be reserved by one train at a time, no section of the inbound route may be reserved by another train between the moment the inbound route was reserved and the moment the section is released again. The latter happens as soon as the train leaves the section. Hence, as the train traverses the chosen inbound route, it sequentially releases each

of the sections comprising the route. An advantage of the fact that each train reserves a complete inbound route is that it guarantees that each train can travel without interruptions along the reserved inbound route to a platform. If, on the contrary, only part of an inbound route were reserved, then it might happen that, somewhere on the chosen inbound route, a train has to wait for the release of a section until it can continue its route.

A similar procedure is followed for the outbound route. That is, a complete outbound route leading from a platform towards a leaving point is reserved for each train. Also, if a train does not stop at a platform, then a complete route through the railway station, consisting of an inbound route and an outbound route, is reserved for the train.

Other constraints that have to be taken into account concern the coupling or uncoupling of trains at a platform, and a number of service considerations, such as the possibility for passengers to transfer between trains, or the desirability that all trains into the same direction depart as much as possible from the same platform.

3. Notation

We always consider one railway station at a time. The layout of the railway station consists of a set of track sections S . The set of platforms of the railway station is denoted by P . We assume that each platform $p \in P$ corresponds to exactly one track section. Therefore the set of platform sections is also denoted by $P \subset S$.

In order to keep the presentation clear, we assume that each train stops at the involved railway station. Thus the route of each train consists of a combination of an inbound route and an outbound route. The set of available routes R can be determined from the set of sections S . Each route r consists of a set of sections $S_r \subset S$.

The set of trains to be routed through the railway station is denoted by T . Train $t \in T$ has corresponding arrival and departure times A_t and D_t , usually in minutes. As was mentioned already in the introduction, a timetable generated by CADANS is cyclic with a cycle length of 60 minutes. That is, the arrival and departure times of trains should be considered *modulo 60 min-*

utes. For example, if a train arrives at time instant 59 and has a standstill of 4 minutes, then it will leave the railway station at time instant 3. Nevertheless, for the time being, we will neglect the fact that the timetable has a cyclic character. We will briefly return to this topic in Section 7.

Train t enters the railway station at its entering point E_t , and leaves the railway station from its leaving point L_t . For each train t a set $R_t \subset R$ of feasible routes is known, determined by the entering point E_t , the leaving point L_t , and possibly a number of other aspects, such as a set of allowed or preferred platforms. Furthermore, the set of sections that may be passed by train t is denoted by $S_t = \bigcup_{r \in R_t} S_r$.

Recall that the inbound route for a train is reserved at a single time instant and that the same holds for the outbound route. Thus a train is to be assigned to a complete inbound route and to a complete outbound route. Furthermore, the sections within a route are released one-by-one, as the train traverses along its route. If train t is assigned to route $r \in R_t$, then the start time of the reservation of the inbound part of route r for train t is denoted by $S(t, r, 1)$, and the start time of the reservation of the outbound part of route r for train t is denoted by $S(t, r, 2)$. Furthermore, the time instant the reservation of section s for train t and the inbound part of route r is released is denoted by $F(t, r, s, 1)$, and the time instant the reservation of section s for train t and the outbound part of route r is released is denoted by $F(t, r, s, 2)$. In these release times also some additional time may be included, in order to guarantee a certain buffer time between the crossings by two trains of a common section.

The time instants $S(t, r, i)$ and $F(t, r, s, i)$ are determined using well known formulas from the theory of dynamics, taking into account (i) the given so-called *left-to-right* coordinates of the sections, and (ii) the assumption that trains have either a constant velocity, or a constant acceleration or deceleration. However, also more sophisticated methods to determine these time instants may be implemented.

In many practical situations the time instants $S(t, r, i)$ and $F(t, r, s, i)$ are independent of the chosen route r . For example, we always have $S(t, r, 2) = D_t$, the departure time of train t . In order to keep the presentation as clear as possible, we assume throughout this paper that *all* time instants $S(t, r, i)$ and $F(t, r, s, i)$ are independent of the routes r .

Thus $S(t, i)$ denotes the start time of the reservation of part i of a route for train t , and $F(t, s, i)$ denotes the time instant the reservation of section s within part i of a route for train t is released.

The safety rules described above may be represented using a set $F_{t,t'}$ for each pair of trains $t, t' \in T$. Such a set contains the pairs of allowable route combinations (r, r') for trains t and t' . That is, $(r, r') \in F_{t,t'}$ means that the routing of train t along route r is *compatible* with the routing of train t' along route r' . In other words,

$$\forall s \in S_r \cap S_{r'} \quad \forall i, i' \in \{1, 2\}$$

$$[S(t, i), F(t, s, i)) \cap [S(t', i'), F(t', s, i')) = \emptyset. \quad (1)$$

Thus the reservation of a common section $s \in S_r \cap S_{r'}$ within part i of route r for train t does not conflict with the reservation of this section within part i' of route r' for train t' .

Now suppose we have an assignment of trains to routes. Let $r(t)$ denote the route train t has been assigned to. Then, by definition, the assignment $t \rightarrow r(t)$ is *feasible from a safety point of view* if

$$\forall t, t' \in T \quad \forall s \in S_{r(t)} \cap S_{r(t')} \quad \forall i, i' \in \{1, 2\}$$

$$[S(t, i), F(t, s, i)) \cap [S(t', i'), F(t', s, i')) = \emptyset. \quad (2)$$

It is clear that the use of the sets $F_{t,t'}$ is a very general method for modeling the safety rules. Actually, many other constraints may be modeled in the same way. For instance, consider the situation where two trains have to be *coupled* at a railway station. One of the trains is called the *leading* train, while the other train, called the *following* train, has to be coupled onto the leading train. In this case, the leading train has to be assigned to an inbound and an outbound route, while the following train has to be assigned to an inbound route only (that matches with the route of the leading train). A similar situation occurs if a train has to be *uncoupled* into two parts. It is clear that these coupling and uncoupling constraints may be modeled using the sets $F_{t,t'}$ introduced above. In this case, (r, r') is an element of $F_{t,t'}$ if and only if a necessary coupling or uncoupling procedure involving trains t and t' can be performed when this route combination is used.

Finally, service considerations towards the passengers may dictate that certain groups of trains all leave from the same platform. For instance, such a group of

trains may consist of all trains leaving into the same direction. Furthermore, one may wish to incorporate certain transfer possibilities between trains into the schedule. That is, pairs of trains need to use platforms close to each other (for example, a *cross-platform* assignment). Once again, these service constraints may be modeled by adjusting the sets $F_{t,t'}$ appropriately.

In this paper we will only take into account constraints that may be modeled using the sets $F_{t,t'}$. Moreover, we will consider the sets $F_{t,t'}$ as part of the input of our problem. The sets $F_{t,t'}$ may be determined in a preprocessing step requiring $O(|S||T|^2|R|^2)$ time.

4. Reduction of sections and routes

In this section we describe how a number of track sections and a number of routes may be eliminated from an instance of the feasibility problem, since it can be seen a priori that they do not add to the feasibility of the instance.

First, we will show in Lemma 1 that only the sections (i) containing a switch, (ii) corresponding to the entering and leaving points, (iii) corresponding to the platforms, or (iv) corresponding to a cross-over of routes are *relevant*. The set of relevant sections is denoted by S^* . Usually S^* contains significantly less elements than S . Note that, by definition of S^* , each section $s \in S \setminus S^*$ is located between two sections $s', s'' \in S^*$.

Lemma 1. *An assignment $t \rightarrow r(t)$ is feasible from a safety point of view if and only if*

$$\forall t, t' \in T \quad \forall s \in S_{r(t)} \cap S_{r(t')} \cap S^* \quad \forall i, i' \in \{1, 2\} \\ [S(t, i), F(t, s, i)) \cap [S(t', i'), F(t', s, i')) = \emptyset. \quad (3)$$

Proof. Since the *only if* part of the statement is obvious, we will only prove the *if* part. To that end, suppose condition (3) is satisfied, and choose trains $t, t' \in T$, a section $s \in (S_{r(t)} \cap S_{r(t')}) \setminus S^*$, and $i, i' \in \{1, 2\}$.

Without loss of generality $S(t', i') \geq S(t, i)$. Let s' be the next relevant section on part i of route $r(t)$ after section s . Note that section s' is also on the same (inbound or outbound) part of route $r(t')$ as section s . Then the fact $[S(t, i), F(t, s', i)) \cap [S(t', i'), F(t', s', i')) = \emptyset$, together with $S(t', i') \geq S(t, i)$, implies $S(t', i') \geq F(t, s', i)$. This result

combined with the obvious inequality $F(t, s', i) > F(t, s, i)$ implies $S(t', i') > F(t, s, i)$. It follows that $[S(t, i), F(t, s, i)) \cap [S(t', i'), F(t', s, i')) = \emptyset$. Thus condition (2) is satisfied. \square

It should be noted that the result of Lemma 1 also holds in case of a cyclic timetable. This is a result of the fact that each section $s \in S \setminus S^*$ is located between two sections $s', s'' \in S^*$. The details of this statement are left to the reader.

Based on the information contained in the sets $F_{t,t'}$ some of the allowed routes for a certain train may be excluded from further consideration, since they are *dominated* by other routes. In particular, route $r \in R_t$ may be eliminated from the set R_t if there exists another route $\tilde{r} \in R_t$ that leaves at least the same routing possibilities for all other trains. Thus

$$\{r' \mid (r, r') \in F_{t,t'}\} \subset \{\tilde{r}' \mid (\tilde{r}, \tilde{r}') \in F_{t,t'}\} \\ \text{for all trains } t' \neq t.$$

For example, in our application, route r is dominated by route \tilde{r} , if the set of relevant sections of route \tilde{r} is a subset of the set of relevant sections of route r . The latter is a consequence of the method used to determine the time instants $S(t, i)$ and $F(t, s, i)$.

5. Model formulation

In this section we describe the feasibility problem of routing trains through railway stations in terms of an integer linear program. Thereafter we describe the connection of the feasibility problem with the *Fixed Interval Scheduling Problem* (FISP). The most general variant of the feasibility problem is defined as follows.

The general feasibility problem. Given a railway station with a corresponding set of routes R . For each train t a set of feasible routes $R_t \subset R$ is given. Furthermore, for each pair of trains t and t' a set $F_{t,t'}$ of allowed pairs of routes is given. Then the question is whether there exists a feasible assignment of trains to routes.

The restricted version of the feasibility problem where all sets $F_{t,t'}$ are completely determined by the

safety aspects (1) (that is: $(r, r') \in F_{t,r'}$ if and only if condition (1) is satisfied) is called the *safety feasibility problem*.

In our application we are not satisfied with a no-answer for a no-instance of the feasibility problem: the corresponding solution should also point at the blocking trains. Therefore we choose to formulate the problem also as an optimization problem, where the objective is to maximize the number of trains that can be routed through the railway station. The corresponding versions of the routing problem are called the *general optimization problem* and the *safety optimization problem*.

5.1. Formulation as an integer program

In order to model the general feasibility problem as an integer linear program, we choose the binary decision variables $X_{t,r}$ for all $t \in T$ and $r \in R_t$. The decision variable $X_{t,r}$ assumes the value 1 if train t is assigned to route r and the value 0 otherwise. Now the objective of the feasibility problem is to find out whether a feasible solution exists to the following constraints:

$$\sum_{r \in R_t} X_{t,r} = 1 \quad \text{for all } t \in T, \quad (4)$$

$$X_{t,r} + X_{t',r'} \leq 1 \quad \text{for all } t, t' \in T; r \in R_t; r' \in R_{t'}; (r, r') \notin F_{t,t'}, \quad (5)$$

$$X_{t,r} \in \{0, 1\} \quad \text{for all } t \in T; r \in R_t. \quad (6)$$

Constraints (4) ensure that each train is assigned to exactly one route. Constraints (5) guarantee that only allowed train-route combinations are selected. These constraints may include constraints due to safety considerations, coupling and uncoupling of trains, and service aspects. Finally, constraints (6) declare the variables as binary.

The general optimization problem, where the objective is to maximize the number of trains that can be routed through the railway station, reads as follows:

$$\max \sum_{t \in T} \sum_{r \in R_t} X_{t,r}, \quad (7)$$

$$\text{subject to } \sum_{r \in R_t} X_{t,r} \leq 1 \quad \text{for all } t \in T, \quad (8)$$

(5) and (6).

Constraints (8) ensure that each train is assigned to at most one route. Obviously, a yes-instance of the feasibility problem can be recognized by the value $|T|$ for the objective function in the optimization problem.

The integer linear program has the general structure $\max \{c^T X \mid MX \leq \mathbf{1}, X \text{ binary}\}$. Here X is a binary vector of decision variables, M is a zero/one matrix and $\mathbf{1}$ is a vector of 1's. This representation shows that the problem can be interpreted as a *Node Packing Problem* (NPP) on the incidence graph of the matrix M . This allows one to deduce a number of valid inequalities that tighten the integer programming formulation, and thus make the LP-relaxation more accurate. For example, inequalities that are valid for the NPP are the *clique inequalities*, and the *(lifted) odd hole inequalities* (cf. Padberg [21]).

In principle, the routes occurring in the integer linear program are complete routes through the railway station. However, if there are many inbound routes from certain entering points to certain platforms and/or many outbound routes from certain platforms to certain leaving points, then it may be worthwhile to replace the complete routes by combinations of inbound and outbound routes. Indeed, suppose for train t there are n_1 inbound routes from the entering point E_t to each platform, and n_2 outbound routes from each platform to the leaving point L_t . Then the number of complete routes for train t from E_t to L_t visiting one of the platforms equals $n_1 \times n_2 \times |P|$, where $|P|$ denotes the number of platforms. Hence in the model the number of variables $X_{t,r}$ for train t also equals $n_1 \times n_2 \times |P|$. On the other hand, if a distinction were made between the inbound routes and the outbound routes, then $n_1 \times |P|$ variables are required to choose an inbound route for train t , and $n_2 \times |P|$ variables are required to choose an outbound route for train t . Thus the total number of variables required for train t would be $(n_1 + n_2) \times |P|$, which may be significantly less than $n_1 \times n_2 \times |P|$. As before, the sets $F_{t,t'}$ may be used to guarantee that train t is assigned to *matching* inbound and outbound routes. As a consequence, the NPP-structure of the model is retained. Splitting complete routes into inbound and outbound routes may also be useful if one wants to include shunting movements of trains into the model formulation (cf. Kroon and Zwaneveld [20]).

5.2. Generalization of FISP

If the cyclic structure of the timetable is neglected, then each train t needs to occupy a platform during the fixed time interval between A_t and D_t . As a consequence, the general feasibility problem studied in this paper is a generalization of the well known *Fixed Interval Scheduling Problem* (FISP). In FISP a number of jobs, each one having a fixed start time and a fixed finish time, have to be carried out by a number of parallel identical machines. The question is whether it is possible to find a feasible schedule for all jobs. Hashimoto and Stevens [13] and Gupta et al. [12] show that FISP can be solved in $O(J \log J)$ time. Here J denotes the number of jobs to be carried out. FISP was also studied by Gertsbakh and Stern [11] in the context of fleet planning.

In the variant of FISP described by Arkin and Silverberg [1] each job can be carried out by a subset of the machines only, and the objective is to find a schedule for a subset of the jobs of maximum total value. Arkin and Silverberg show that their variant of FISP is NP-hard. However, they also present an algorithm based on dynamic programming that solves the problem in $O(J^{M+1})$ time, where J denotes the number of jobs and M denotes the number of machines. Hence if the number of machines is fixed, then an optimal solution can be found in an amount of time that is polynomial in the number of jobs.

It follows that the variant of FISP described by Arkin and Silverberg can be used to model the assignment of trains to platforms if one neglects the complexities due to cross-overs of inbound and/or outbound routes. If one also takes into account the fact that certain assignments of trains to platforms are not allowed due to cross-overs of inbound and/or outbound routes, then one obtains a problem that seems to be significantly more difficult. Nevertheless, in the next section we will show that an approach based on dynamic programming, which is an extension of the approach of Arkin and Silverberg, can be used for solving the feasibility problem of routing trains through railway stations.

Further variants of FISP are studied by Dondeti and Emmons [4, 5], by Fischetti et al. [7–9] in the context of bus driver scheduling, and by Kolen and Kroon [14–17] in the context of aircraft maintenance. The latter papers present a comprehensive description of

the computational complexity of several variants of FISP.

6. Computational complexity

As was mentioned already, the problem of routing trains through railway stations in its most general form is a generalization of the feasibility version of the problem described by Arkin and Silverberg [1]. Since the latter problem is known to be NP-complete, the feasibility problem of routing trains through railway stations is NP-complete as well. In fact, the latter will also follow from Lemma 2, which shows that the safety feasibility problem is NP-complete as soon as each train has three routing possibilities. This result is deduced using the well known Satisfiability (SAT) problem.

Satisfiability problem. Given m clauses C_1, \dots, C_m and n boolean variables x_1, \dots, x_n . Each clause consists of a number of literals (variables, or negations of variables). A clause is satisfied if one of the literals of the clause is TRUE. That is, a variable that occurs in the clause is TRUE or a variable whose negation occurs in the clause is FALSE. Then the question is whether there exists an assignment of boolean values TRUE and FALSE to the variables such that all clauses are satisfied.

Special cases of SAT are 2-SAT and 3-SAT. In 2-SAT each clause contains exactly 2 literals, and in 3-SAT each clause contains exactly 3 literals. It is known that 2-SAT can be solved in $O(\max\{m, n\})$ time (cf. Papadimitriou and Steiglitz [22], and Tarjan [24]), whereas 3-SAT belongs to the class of NP-complete problems.

6.1. Variable layout of the railway station

In this section we will first prove Lemma 2 showing that the safety feasibility problem is NP-complete as soon as each train has three routing possibilities.

Lemma 2. *If $|R_t| = 3$ for each train t , then the safety feasibility problem is NP-complete.*

Proof. This lemma is proved by a reduction from 3-SAT. Indeed, let I be an instance of 3-SAT containing m clauses C_i in n variables, each one containing exactly three literals.

Then we construct the following instance I' of the safety feasibility problem. With each clause C_i we associate a train t . Each train corresponds to a unique entering direction, and to a unique leaving direction, and each train has to be assigned to a unique prespecified platform. Thus the number of entering directions, the number of leaving directions, and the number of platforms equals $|T|$.

For train t there are three inbound routes r_t^1 , r_t^2 , and r_t^3 , each one leading from the train's entering point towards the train's platform. Furthermore, for each train there is exactly one outbound route leading from the train's platform towards the train's leaving point. Thus each train has exactly three routing possibilities through the railway station, which are also indicated by r_t^1 , r_t^2 , and r_t^3 . Each pair of inbound routes has a cross-over or a fly-over. The outbound routes do not cross the inbound routes, nor the other outbound routes. All inbound routes have the same length and consist of one single section.

All trains have the same arrival time and the same departure time. This, together with the previous conditions, implies that a pair of routes is compatible for a pair of trains if and only if the involved inbound routes do not have a cross-over.

Route r_t^i of train t corresponds with the i th literal of clause C_i . Thus, if train t is assigned to route r_t^i , then the i th literal of clause C_i is assigned the value TRUE, and vice versa. Therefore route r_t^i has a cross-over with route $r_{t'}^j$ if the i th literal of clause C_i is the negation of the j th literal of clause $C_{i'}$. Otherwise route r_t^i has a fly-over with route $r_{t'}^j$. An example of a situation with two trains is shown in Fig. 2.

Next, suppose I is a yes-instance of 3-SAT. Then for each clause C_i at least one of the literals is assigned the

value TRUE. Select one such literal, and assign train t to the corresponding route. This obviously gives a feasible assignment of trains to routes in I' .

Conversely, suppose I' is a yes-instance of the safety feasibility problem. Then each train t is assigned to exactly one route r_t^i . Now the value TRUE is assigned to the i th literal of clause t in I . Furthermore, all unassigned variables are given an arbitrary value. Then obviously all clauses are satisfied, and since no conflicts occur in the assignment of trains to routes, the obtained solution is feasible for I .

Thus I is a yes-instance of 3-SAT if and only if I' is a yes-instance of the safety feasibility problem. Since 3-SAT is NP-complete, and it is clear that the safety feasibility problem is in NP, the safety feasibility problem is NP-complete as well. \square

It should be noted that the general feasibility problem is really more general than the safety feasibility problem. In fact, it can be proved in a similar fashion as Lemma 2 that the following result holds for the general feasibility problem.

Lemma 3. *The general feasibility problem is NP-complete, even if there are only three routes through the railway station.*

This result implies that the general feasibility problem is NP-complete even if the layout of the railway station is fixed. This is in contrast with the safety feasibility problem and the safety optimization problem which can be solved in polynomial time as soon as the layout of the railway station is fixed. The latter will be proved in the following section.

As is often the case in complexity theory, the numbers 2 and 3 describe the borderline between the *easy* problems and the *hard* problems. This is illustrated in Lemma 4 showing that the general feasibility problem can be solved in polynomial time if there are only two routes per train.

Lemma 4. *If $|R_t| \leq 2$ for each train t , then the general feasibility problem can be solved in $O(|T|^2)$ time, given the sets $F_{t,t'}$.*

Proof. Let I be an instance of the general feasibility problem. We will reduce I to an instance I' of 2-SAT. The details are as follows:

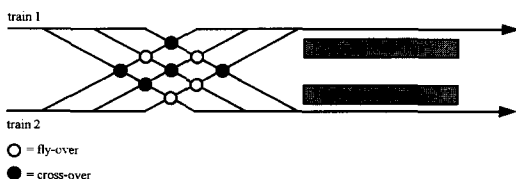


Fig. 2. An example with 2 trains.

- (1) For each train t with $|R_t| = 2$ we define variables x_t^1 and x_t^2 . For each train t with $|R_t| = 1$ we define one variable x_t^1 .
- (2) For each train t with $|R_t| = 2$ we define clauses $(x_t^1 \vee x_t^2)$ and $(\bar{x}_t^1 \vee \bar{x}_t^2)$. For each train t with $|R_t| = 1$ we define a clause $(x_t^1 \vee x_t^1)$.
- (3) If the i th route of train t is not compatible with the j th route of train t' , then we define a clause $(\bar{x}_t^i \vee \bar{x}_{t'}^j)$.

Obviously, steps 1 and 2 take $O(|T|)$ time. Step 3 takes $O(|T|^2)$ time, since for each pair of trains $t, t' \in T$ the set $F_{t,t'}$ contains at most four pairs of routes (r, r') . Since the number of variables of I' is $O(|T|)$ and the number of clauses of I' is $O(|T|^2)$, it can be verified in $O(|T|^2)$ whether I' is a yes-instance of 2-SAT. Furthermore, it is not difficult to see that I is a yes-instance of the general feasibility problem if and only if I' is a yes-instance of 2-SAT. It follows that the general feasibility problem can be solved in $O(|T|^2)$ time. \square

6.2. Fixed layout of the railway station

In this section we show that, for a station with a fixed layout, the safety optimization problem can be solved in an amount of time that is polynomial in the number of trains to be routed. This result is an extension of the result of Arkin and Silverberg [1] mentioned in Section 5. In order to prove this result, we will consider the actual railway operations in more detail. To that end, the set of inbound routes fit for train t is denoted by R_t^i , and the set of outbound routes fit for train t is denoted by R_t^o .

Furthermore, a (possible) change in the reservation of a section for a train is called an *event*. Although in the previous sections the notations $S(t, i)$ and $F(t, s, i)$ were used to denote time instants, they will also be used to denote the corresponding events, where the correct interpretation should be clear from the context. There are $2|T|$ events $S(t, i)$ corresponding to the reservation of an inbound or an outbound route for a train, and there are at most $2|T||S^*|$ events $F(t, s, i)$ corresponding to the release of the reservation of a section for a train. Thus the number of events does not exceed $2|T|(1 + |S^*|)$.

Note that each event occurs at a single time instant. However, several events may occur at the same time

instant. Thus the events can be sorted in *chronological* order, taking into account the rule that, whenever two events $F(t, s, i)$ and $S(t', i')$ occur at the same time instant, the event $F(t, s, i)$ occurs *first* and the event $S(t', i')$ occurs *second*. In this way, the reservation of a section is first released before the section can be reserved by another train. The list of events is denoted by $\{e_q \mid q = 1, \dots, Q\}$, where the index q is used to represent the described order of the events.

Theorem 5. *If the layout of the railway station is fixed, then the safety optimization problem can be solved in polynomial time.*

Proof. This theorem is proved by showing that the problem can be solved as a shortest path problem on a network with numbers of nodes and arcs that are polynomial in the number of trains if the layout of the railway station is fixed. The underlying network is based on the events e_q for $q = -1, 0, 1, \dots, Q + 1$. Here the event e_{-1} is an initial event prior to all other events, and the event e_{Q+1} is a final event after all other events.

The nodes in the network are $|S^*|$ -vectors, for $q = -1, \dots, Q$ representing feasible assignments of sections to trains between the events e_q and e_{q+1} . That is, the nodes in the network have the form $X_q = (x_{1,q}, \dots, x_{|S^*,q|})$ where $x_{s,q} = 0$ or $x_{s,q} \in T$ for $q = -1, \dots, Q$ and $s = 1, \dots, |S^*|$. Here $x_{s,q} = 0$ means that section s is free between the events e_q and e_{q+1} , and $x_{s,q} = t$ means that section s is reserved for train t between the events e_q and e_{q+1} . Note that for $q \in \{-1, Q\}$ there is only one node $X_q = (x_{1,q}, \dots, x_{|S^*,q|})$. Indeed, for $q \in \{-1, Q\}$ we have $x_{\sigma,q} = 0$ for all $\sigma \in S^*$.

If there is an arc in the network from a node $X_p = (x_{1,p}, \dots, x_{|S^*,p|})$ to a node $X_q = (x_{1,q}, \dots, x_{|S^*,q|})$, then $p = q - 1$. The latter implies that the network is acyclic. Now the network is completed by performing the following steps for $q = 1, \dots, Q$:

- Suppose $e_q = S(t, 1)$ for some $t \in T$. Then there is an arc from a node $X_{q-1} = (x_{1,q-1}, \dots, x_{|S^*,q-1|})$ to a node $X_q = (x_{1,q}, \dots, x_{|S^*,q|})$ if condition (i) or condition (ii) is satisfied:

- (i) There exists a route $r \in R_t^i$ such that $x_{\sigma,q-1} = 0$ for all $\sigma \in S_r$; furthermore, $x_{\sigma,q} = t$ for all $\sigma \in S_r$ and $x_{\sigma,q} = x_{\sigma,q-1}$ for all $\sigma \notin S_r$; if these conditions are satisfied, then the arc has length 1.

(ii) $x_{\sigma,q} = x_{\sigma,q-1}$ for all $\sigma \in S^*$; if this is satisfied, then the arc has length 0.

• Suppose $e_q = S(t, 2)$ for some $t \in T$. Then there is an arc of length 0 from a node $X_{q-1} = (x_{1,q-1}, \dots, x_{|S^*|,q-1})$ to a node $X_q = (x_{1,q}, \dots, x_{|S^*|,q})$ if condition (iii) or condition (iv) is satisfied:

(iii) There exists a route $r \in R_t^0$ and a platform section $s \in S_r \cap P$ such that $x_{s,q-1} = t$ and $x_{\sigma,q-1} = 0$ for all $\sigma \in S_r \setminus \{s\}$; furthermore, $x_{\sigma,q} = t$ for all $\sigma \in S_r$ and $x_{\sigma,q} = x_{\sigma,q-1}$ for all $\sigma \notin S_r$.

(iv) For all platform sections $s \in P$ we have $x_{s,q-1} \neq t$; furthermore $x_{\sigma,q} = x_{\sigma,q-1}$ for all $\sigma \in S^*$.

• Suppose $e_q = F(t, s, i)$ for some $t \in T, i \in \{1, 2\}$, and $s \in S_r$. Then there is an arc of length 0 from a node $X_{q-1} = (x_{1,q-1}, \dots, x_{|S^*|,q-1})$ to a node $X_q = (x_{1,q}, \dots, x_{|S^*|,q})$ if condition (v) or condition (vi) is satisfied:

(v) $x_{s,q-1} = t$, $x_{s,q} = 0$, and $x_{\sigma,q} = x_{\sigma,q-1}$ for all $\sigma \neq s$.

(vi) $x_{s,q-1} \neq t$, and $x_{\sigma,q} = x_{\sigma,q-1}$ for all $\sigma \in S^*$.

If condition (i) is satisfied, then inbound route r is reserved for train t . Indeed, condition (i) implies that all sections $s \in S_r$ are free between the events e_{q-1} and e_q . Hence all sections $s \in S_r$ are reserved for train t between the events e_q and e_{q+1} , and all other reservations remain unchanged.

If condition (iii) is satisfied, then outbound route r is reserved for train t . Indeed, condition (iii) implies that train t has a standstill at platform section $s \in S_r \cap P$ between the events e_{q-1} and e_q , and that all other sections $\sigma \in S_r \setminus \{s\}$ are free between the events e_{q-1} and e_q . Hence all sections $\sigma \in S_r$ are reserved for train t and route r between the events e_q and e_{q+1} , and all other reservations remain unchanged. Note that, if $x_{s,q-1} = t$ for some platform section $s \in P$ and there does not exist a route $r \in R_t^0$ with $s \in S_r$ such that $x_{\sigma,q-1} = 0$ for all $\sigma \in S_r \setminus \{s\}$, then there is no arc leaving node $X_{q-1} = (x_{1,q-1}, \dots, x_{|S^*|,q-1})$. This implies that in that case each path leading from node X_{-1} to this node is a deadheading path, which cannot be extended to a full path to node X_Q .

If condition (ii) or condition (iv) is satisfied, then no route is reserved for train t . Hence all reservations remain unchanged. If train t is waiting at a platform section $s \in P$, then it should be assigned to an outbound route at event $e_q = S(t, 2)$. Therefore the additional condition in (iv) that $x_{s,q-1} \neq t$ for all $s \in P$

is necessary.

If condition (v) is satisfied, then section s was reserved for train t between the events e_{q-1} and e_q . Since $e_q = F(t, s, i)$, this reservation is released, and all other reservations remain unchanged. If condition (vi) is satisfied, then section s was not reserved for train t between the events e_{q-1} and e_q . Hence all reservations remain unchanged.

We have constructed now a directed network. Obviously, only arcs corresponding to the reservation of an inbound route have a positive (unit) length. It follows that a path of total length L from node X_{-1} to node X_Q represents a feasible routing for L trains through the railway station. Thus a feasible routing for a maximum number of trains through the railway station corresponds to a path of maximum length from node X_{-1} to node X_Q . Since the network is acyclic, we can solve the longest path problem by replacing each arc length by its negative value and by finding the shortest path. We can use the idea of Edmonds and Karp [6] to carry out the shortest path computation over an equivalent network with non-negative arc lengths. Fredman and Tarjan [10] have shown that for a network $G = (V, E)$ the algorithm of Dijkstra can run in $O(|V| \log |V| + |E|)$ time.

Therefore it is relevant to count the number of nodes and arcs of the network. Obviously, the number of nodes per event e_q is $O(|T|^{|S^*|})$. Recall that the number of events Q does not exceed $2|T|(1 + |S^*|)$. Thus the number of nodes $|V|$ is $2|T|(1 + |S^*|)O(|T|^{|S^*|}) = O(|S^*||T|^{|S^*|+1})$. Furthermore, if condition (i) is satisfied for an event $e_q = S(t, 1)$, then the number of arcs leaving a node $X_{q-1} = (x_{1,q-1}, \dots, x_{|S^*|,q-1})$ does not exceed $|R_t^i|$. If condition (ii) is satisfied for such an event, then the number of such leaving arcs equals 1. Similarly, if condition (iii) is satisfied for an event $e_q = S(t, 2)$, then the number of arcs leaving a node $X_{q-1} = (x_{1,q-1}, \dots, x_{|S^*|,q-1})$ does not exceed $|R_t^0|$. If condition (iv) is satisfied for such an event, then the number of such leaving arcs equals 1. Finally, if condition (v) or (vi) is satisfied for an event $e_q = F(t, s, i)$, then the number of arcs leaving a node $X_{q-1} = (x_{1,q-1}, \dots, x_{|S^*|,q-1})$ equals 1. Thus the number of arcs $|E|$ is $|R^i||T|O(|T|^{|S^*|}) + |R^0||T|O(|T|^{|S^*|}) + 2|S^*||T|O(|T|^{|S^*|}) = O((|R^i| + |R^0| + 2|S^*|)|T|^{|S^*|+1})$.

If the layout of the railway station is fixed, then $|R^i|$, $|R^0|$, and $|S^*|$ are fixed. Hence in that case

$|V| = O(|T|^{|S^*|+1})$, and $|E| = O(|T|^{|S^*|+1})$ as well. Thus the routing problem can be solved then in $O(|T|^{|S^*|+1} \log |T|)$ time. \square

It should be noted that the result of Theorem 5 is interesting mostly from a theoretical point of view, since it provides a generalization of the result of Arkin and Silverberg [1] concerning FISP. The practical value of Theorem 5 is limited, due to the size of the described network. For example, in the medium-sized Dutch railway station Zwolle, the hourly number of trains to be routed equals about 20, and the number of relevant sections equals about 50. Therefore the $O(|T|^{|S^*|+1} \log |T|)$ amount of time is of little practical value for this railway station, let alone for larger-sized railway stations like Amsterdam or Utrecht. For a description of algorithms used in practice to solve the routing problem we refer to Zwaneveld et al. [25].

The proof of Theorem 5 is strongly based on the assumption that the events $S(t, i)$ and $F(t, s, i)$ are independent of the involved routes. However, if these events are dependent of the chosen routes, and are actually events $S(t, r, i)$ and $F(t, r, s, i)$, then the problem can still be solved in an amount of time that is polynomial in the number of trains, given the layout of the railway station. The latter can be established by considering all events $S(t, r, i)$ and $F(t, r, s, i)$ in the network, and by reserving a section for a train-route combination, instead of for a train only. Obviously, the size and complexity of the network will further increase by this modification. However, the result is still a polynomial algorithm for safely routing a maximum number of trains.

In Theorem 5 only the safety aspect of an assignment of trains to routes is taken into account, whereas other aspects, such as coupling and uncoupling of trains and service aspects, are not considered. However, coupling and uncoupling of trains may also be incorporated into the approach of Theorem 5. Indeed, suppose trains t and t' with $S(t, 1) < S(t', 1)$ have to be coupled. Then train t is first assigned to an inbound route. Next, at the moment a matching inbound route for train t' is to be selected, train t is still reserving a platform section $s \in P$, since otherwise the coupling of trains cannot be performed. Thus an appropriate inbound route $r \in R_t^i$ for train t' may be chosen, given the platform section s reserved by train t . Uncoupling

of trains may be incorporated in a similar way. Thus we have obtained the following corollary.

Corollary 6. *If the layout of the railway station is fixed, then the safety optimization problem, including coupling and uncoupling of trains, can be solved in polynomial time.*

Also some service aspects, such as preferences of certain trains for certain routes or platforms, may be handled by the dynamic programming approach. Furthermore, also the requirement that certain pairs of trains obtain a cross-platform assignment may be handled in the same way as the aspect of coupling and uncoupling of trains. Finally, different safety systems may be included. For example, also a system wherein the inbound and outbound routes are reserved on a section-by-section basis may be handled similarly.

If it is desirable that all trains into the same direction depart from the same platform, then this cannot be incorporated directly into the approach described above. Indeed, two trains bound for the same direction usually do not have overlapping standstill intervals at the railway station. Therefore at the moment the second train arrives at the railway station, the platform used by the first one cannot be recognized from the state of the system at that time, unless it is recorded somehow.

Therefore the approach of Theorem 5 is modified in the following way. Now the state variables are extended to be $(|S^*|+n)$ -vectors, $X_q = (x_{1,q}, \dots, x_{|S^*|,q}, y_{1,q}, \dots, y_{n,q})$, where n denotes the number of leaving directions of the railway station. Position l of the latter n positions of a state variable is used to record the platform used for the trains bound for leaving direction l . Such a value is set whenever the first train bound for this direction enters the railway station, and is taken into account by all consecutive assignments of trains to inbound routes. The steps in the network description become as follows.

- Suppose $e_q = S(t, 1)$ for some $t \in T$ bound for direction L_t . Then there is an arc from a node $X_{q-1} = (x_{1,q-1}, \dots, x_{|S^*|,q-1}, y_{1,q-1}, \dots, y_{n,q-1})$ to a node $X_q = (x_{1,q}, \dots, x_{|S^*|,q}, y_{1,q}, \dots, y_{n,q})$ if condition (i'_a), (i'_b), or (ii') is satisfied:

- (i'_a) there exists a route $r \in R_t^i$ with platform section $s \in S_r \cap P$ such that $x_{\sigma,q-1} = 0$ for all $\sigma \in S_r$ and $y_{l,q-1} = 0$ for $l = L_t$; furthermore, $x_{\sigma,q} = t$

for all $\sigma \in S_r$ and $x_{\sigma,q} = x_{\sigma,q-1}$ for all $\sigma \notin S_r$; also $y_{l,q} = s$ for $l = L_t$, and $y_{l,q} = y_{l,q-1}$ for all $l \neq L_t$,

- (i'_b) there exists a route $r \in R_t^i$ with platform section $s \in S_r \cap P$ such that $x_{\sigma,q-1} = 0$ for all $\sigma \in S_r$ and $y_{l,q-1} = s$ for $l = L_t$; furthermore, $x_{\sigma,q} = t$ for all $\sigma \in S_r$ and $x_{\sigma,q} = x_{\sigma,q-1}$ for all $\sigma \notin S_r$; also $y_{l,q} = y_{l,q-1}$ for $l = 1, \dots, n$.
- (ii') $x_{\sigma,q} = x_{\sigma,q-1}$ for all $\sigma \in S^*$ and $y_{l,q} = y_{l,q-1}$ for $l = 1, \dots, n$.

If condition (i'_a) is satisfied, then train t is the first train bound for leaving direction L_t that arrives at the railway station. Thus train t can be assigned to any appropriate inbound route $r \in R_t^i$. The platform section s used by train t is recorded in $y_{L_t,q}$. If condition (i'_b) is satisfied, then train t can be assigned only to an inbound route leading towards platform section s used by the previous trains bound for leaving direction L_t .

Obviously, the number of nodes and arcs of the network increases by this extension of the state space. Nevertheless, since the number of leaving directions is fixed, the network has still a number of nodes and arcs that are polynomial in the number of trains. Thus we have obtained the following corollary.

Corollary 7. *If the layout of the railway station is fixed, then the safety optimization problem, including coupling and uncoupling of trains as well as the requirement that all trains bound for the same direction should depart from the same platform, can be solved in polynomial time.*

If this is desirable, then a more refined distinction between the trains may be handled in a similar way. For example, it may be desirable that all regional trains bound for the same direction depart from the same platform, and that the same holds for the intercity trains bound for the same direction. Such a more refined distinction can be realized by extending the state space even further, in a similar way as described above.

7. Cyclic timetable

As was mentioned already in the introduction, a timetable generated by CADANS has a cyclic structure with a cycle length of 60 minutes. For simplicity, we

have neglected the cyclic structure of the timetable so far, although it obviously increases the complexity of the feasibility problem. For example, it is well known that the cyclic variant of FISP with identical parallel machines is NP-complete (cf. Kolen et al. [18], and Dondeti and Emmons [5]), in contrast with its non-cyclic variant. Nevertheless, it is not difficult to see that all results described in this paper are also valid in case of a cyclic timetable.

For example, Lemma 1 also holds in this case. The latter is a consequence of the fact that each section is located between two relevant sections. Furthermore, the model formulation in Section 5 as well as the complexity results of Lemmas 2, 3, and 4 are obviously independent of the cyclic structure of the timetable. Indeed, the sets R_t and $F_{t,t'}$ for trains t and t' are determined in a preprocessing step. In this step the cyclicity of the timetable may or may not be taken into account.

Also the result of Theorem 5 that the safety optimization problem can be solved in polynomial time if the layout of the railway station is fixed extends to the case of a cyclic timetable, although a slight modification of the network is required in order to prove this. Indeed, in this case the initial state of the system at time instant 0 (= time instant 60) need not be empty. Hence there will be several initial nodes $X_{-1} = (x_{1,-1}, \dots, x_{|S^*|,-1})$ representing the possible initial states of the system. Similarly, there will be several final nodes $X_Q = (x_{1,Q}, \dots, x_{|S^*|,Q})$ representing the possible final states of the system. Now the problem is to find the longest path in the network from some initial state $X_{-1} = (x_{1,-1}, \dots, x_{|S^*|,-1})$ to the same final state $X_Q = (x_{1,Q}, \dots, x_{|S^*|,Q})$ with $x_{\sigma,-1} = x_{\sigma,Q}$ for all $\sigma \in S^*$. Since the number of initial states is $O(|T|^{|S^*|})$, the problem can still be solved in an amount of time that is polynomial in the number of trains if the layout of the railway station is fixed.

Finally, the results of the Corollaries 6 and 7 that in the approach of Theorem 5 also coupling and uncoupling of trains as well as a number of service aspects may be taken into account can be extended directly to the case of a cyclic timetable.

8. Final remarks

In this paper we consider the problem of routing trains through railway stations. We show that only a

subset of the track sections and the routing possibilities needs to be taken into account when deciding whether a certain assignment of trains to routes is feasible from a safety point of view. Furthermore, we describe the problem as an integer linear program, which shows that the problem can be considered as a Node Packing Problem.

A reduction from 3-SAT shows that the safety feasibility problem is NP-complete as soon as each train has three routing possibilities. However, if each train has at most two routing possibilities, then the general feasibility problem can be solved in polynomial time by a reduction to 2-SAT. Furthermore, it is also shown that, if the layout of the railway station is fixed, then the safety optimization problem can be solved by a dynamic programming approach in an amount of time that is polynomial in the number of trains. In the latter approach also coupling and uncoupling of trains, a number of service aspects, and a cyclic timetable may be taken into account.

Currently we are working on the development of an algorithm for solving the problem of routing trains through railway stations. This research is described in a complementary paper by Zwaneveld et al. [25]. The developed algorithm is based on the formulation of the problem as a Node Packing Problem, and on the application of dominance rules, valid inequalities, heuristics, and a branch & cut procedure. It turns out that the algorithm is appropriate for solving the routing problem for almost all railway stations in The Netherlands. Unfortunately, the variance in the processing times is still large, and therefore the processing times are rather unpredictable. We hope to solve this problem by improving the dominance rules. This topic is a subject for further research, together with the implementation of the algorithm into the user-friendly planning system STATIONS, and the design of the interface between the modules STATIONS and CADANS.

References

- [1] Arkin, E.M., and Silverberg, E.L., "Scheduling jobs with fixed starting and finishing times", *Discrete Applied Mathematics* 18 (1987) 1–8.
- [2] Bourachot, J., "Computer-aided planning of traffic in large stations by means of the AFAIG model", *Rail International* (1986) 2–18.
- [3] Carey, M., "A model for train pathing with choice of lines platforms and routes", *Transportation Research* 28B (1994) 333–353.
- [4] Dondeti, V.R., and Emmons, H., "Fixed job scheduling with two types of processors", *Operations Research* 40-1A (1992) S76–S85.
- [5] Dondeti, V.R., and Emmons, H., "Algorithms for preemptive scheduling of different classes of processors to do jobs with fixed times", *European Journal of Operational Research* 70 (1993) 316–326.
- [6] Edmonds, J., and Karp, R.M., "Theoretical improvements in algorithmic efficiency for network flow problems", *Journal of the ACM* 19 (1972) 248–264.
- [7] Fischetti, M., Martello, S., and Toth, P., "The fixed job schedule problem with spread time constraints", *Operations Research* 35/6 (1987) 849–858.
- [8] Fischetti, M., Martello, S., and Toth, P., "The fixed job schedule problem with working time constraints", *Operations Research* 37/3 (1989) 395–403.
- [9] Fischetti, M., Martello, S., and Toth, P., "Approximation algorithms for fixed job schedule problems", *Operations Research* 40-1A (1992) S96–S108.
- [10] Fredman, M.L., and Tarjan, R.E., "Fibonacci heaps and their uses in improved network optimization algorithms", *Proceedings of the 25th Symposium on the Foundations of Computer Science* (1984).
- [11] Gertsbakh, I., and Stern, H.I., "Minimal resources for fixed and variable job schedules", *Operations Research* 26/1 (1978) 68–85.
- [12] Gupta, U.L., Lee, D.T., and Leung, J.Y.-T., "An optimal solution to the channel assignment problem", *IEEE Transactions on Computing* C-28 (1979) 807–810.
- [13] Hashimoto, A., and Stevens, J., "Wire routing by optimizing channel assignments within large apertures", *Proceedings of the 8th Design Automation Workshop* (1971) 155–169.
- [14] Kolen, A.W.J., and Kroon, L.G., "On the computational complexity of (maximum) class scheduling", *European Journal of Operational Research* 54 (1991) 23–38.
- [15] Kolen, A.W.J., and Kroon, L.G., "Licence class design: complexity and algorithms", *European Journal of Operational Research* 63 (1992) 432–444.
- [16] Kolen, A.W.J., and Kroon, L.G., "On the computational complexity of (maximum) shift class scheduling", *European Journal of Operational Research* 64 (1993) 138–151.
- [17] Kolen, A.W.J., and Kroon, L.G., "An analysis of shift class design problems", *European Journal of Operational Research* 79 (1994) 417–430.
- [18] Kolen, A.W.J., Lenstra, J.K., and Papadimitriou, C.H., "Interval scheduling problems", unpublished manuscript (1987).
- [19] Kroon, L.G., and Zwaneveld, P.J., "STATIONS: final report of phase 1", Report 201 of the Rotterdam School of Management, Rotterdam (1995).
- [20] Kroon, L.G., and Zwaneveld, P.J., "Routing trains through railway stations including shunting decisions", forthcoming in: *Operations Research Proceedings 1995*, Springer Verlag, Berlin (1995).

- [21] Padberg, M.W., "On the facial structure of set packing polyhedra", *Mathematical Programming* 5 (1973) 199–215.
- [22] Papadimitriou, C.H., and Steiglitz, K., *Combinatorial optimization: algorithms and complexity*, Prentice-Hall, Englewood Cliffs (1982) 377–378.
- [23] Schrijver, A., and Steenbeek, A., "Dienstregeling ontwikkeling voor Railned (Timetable development for Railned)", Report CADANS 1.0, C.W.I., Amsterdam (1994).
- [24] Tarjan, R., "Depth-first search and linear graph algorithms", *SIAM Journal on Computing* 1/2 (1972) 146–160.
- [25] Zwaneveld, P.J., Dauzère-Pérès, S., van Hoesel, C.P.M., Kroon, L.G., Romeijn, H.E., Salomon, M., and Ambergen, H.A., "Routing trains through railway stations: model formulation and algorithms", forthcoming in: *Transportation Science*.