

A multi-criteria decision support system for real-time train rescheduling

Marcella Samà¹, Carlo Meloni^{2,3},
Andrea D'Ariano¹, Francesco Corman^{4,5}

¹ Università degli Studi Roma Tre, Dipartimento di Ingegneria,
via della vasca navale 79, 00146 Rome, Italy

² Politecnico di Bari, Dipartimento di Ingegneria Elettrica e dell'Informazione,
via E. Orabona 4, 70125 Bari, Italy

³ CNR, Istituto per le Applicazioni del Calcolo "Mauro Picone", Sede di Bari,
via G. Amendola, 122I, 70126 Bari, Italy

⁴ Delft University of Technology, Transport Engineering and Logistics,
Mekelweg 2, 2628CD Delft, The Netherlands

⁵ Katholieke Universiteit Leuven, Center for Industrial Management,
Celestijnenlaan 300A, B-3001 Leuven, Belgium

Abstract

This work addresses the real-time optimization of train scheduling decisions at a complex railway network during congested traffic situations. The problem of effectively managing train operations is particularly challenging, since it is necessary to incorporate the safety regulations into the optimization model and to consider numerous performance indicators that are important to evaluate the solution quality. Moreover, solutions must be produced in a limited computation time. This paper deals with the development of a multi-criteria decision support system to help dispatchers in taking more informed decisions when dealing with real-time disturbances. As a novel idea in the literature, optimal train scheduling solutions are computed with high level precision in the modelling of the safety regulations and with consideration of state-of-the-art performance indicators to be optimized in a given computation time. To this aim, mixed-integer linear programming formulations are proposed based on the alternative graph modeling approach and solved via a commercial solver. For each problem instance, an iterative procedure is executed in order to establish an efficient-inefficient classification of the best solutions provided by the formulations via a well-established non-parametric benchmarking technique, named data envelopment analysis. Based on this classification, the procedure improves inefficient formulations with generation of additional linear constraints. Computational experiments are performed for practical-size instances from a Dutch railway network with mixed traffic and several disturbances. For each problem instance, the procedure converges after a limited number of iterations, and returns a set of efficient solutions and the relative formulations.

Keywords

Railway Traffic Control, Disturbance Management, Performance Evaluation, Mixed-Integer Linear Programming, Data Envelopment Analysis.

1 Introduction

A key problem in real-time railway traffic management is to efficiently reschedule trains during operations (Cacchiani et al. (2014)). In presence of initial delays, this problem requires to detect potential conflicts between two or more trains in each resource and to globally solve them by taking into account the propagation of consecutive delays (D'Ariano, 2009, Corman et al., 2014B). Due to the limited time available to take real-time decisions, train dispatchers usually have a limited view on the effects of conflict resolution methods and are not able to compare alternative solutions in terms of various performance indicators. In practice, the dispatching measures are often sub-optimal, and leave room for improvement. Moreover, there is no clear agreement in literature on the objective functions to be used as many should be considered, due to different stakeholders and operational aspects.

This paper deals with the development of a multi-criteria decision support system to help dispatchers in taking more informed decisions when dealing with real-time disturbance management. As a novel idea in the literature, we first compute optimal train scheduling solutions with high level precision in the modelling of the safety regulations and with consideration of state-of-the-art performance indicators to be optimized in a given computation time. To achieve this aim, the following scientific issues have to be addressed:

- Microscopic optimization models for the train rescheduling problem are required, each one taking into account multiple performance indicators, either in the objective function or in the problem constraints;
- The models are to be automatically created and solved in a short computation time, and to be assessed in terms of the various performance indicators;
- A quantitative technique is needed to automatically compare the best solutions computed for the different models, to select efficient solutions and to give suggestions for improving inefficient ones;
- A comprehensive computational analysis is required to perform the performance assessment and to provide a pool of good quality solutions.

The first issue is addressed by the development of *Mixed-Integer Linear Programming* (MILP) formulations based on the *alternative graph model* of the train rescheduling problem (D'Ariano et al., 2007-2014). The lack of a generally recognized indicator to optimize places importance on the definition of the objectives. We investigate a selection of the most used objectives in the related literature, including the minimization of the maximum (initial plus consecutive) delay (Mazzarello and Ottaviani, 2007), the maximum consecutive delay (D'Ariano et al., 2007, Pellegrini et al., 2014), the maximum consecutive delay with consideration of train priorities (Corman et al., 2011A), the cumulative consecutive delays (Pellegrini et al., 2014), the weighted sum of deviations from the arrival/departure scheduled times (Caimi et al., 2012), the sum of all completion times (Dessouky et al., 2006), the sum of delays (Meng and Zhou, 2014), a weighted sum of delays (Higgins et al., 1996) with penalties when exceeding a threshold (Törnquist and Persson, 2007), the travel time of trains as a surrogate of energy consumption (Corman et al., 2009).

All the objectives studied in this paper are *operational-centric* since they focus on the minimization of railway operations objectives with a train point of view, whereas other approaches are *passenger-centric* since they focus on the maximization of the quality of service perceived by the passengers (the latter approaches are investigated e.g. in Tomii et

al. (2005), Takeuchi et al. (2006), Caprara et al. (2006), Kanai et al. (2011), Sato et al. (2013), Binder et al. (2014), Cacchiani et al. (2014)).

The second issue is organized in the construction of the MILP formulations via the alternative graph model, the resolution of each formulation via a MIP solver, the investigation of the resulting solutions via a post-processing analysis. The latter procedure assesses the quality of each solution in terms of the various performance indicators.

The third issue is addressed via a useful benchmarking technique, named *Data Envelopment Analysis* (DEA), for assessing the relative efficiency of the different solutions (Charnes et al., 1978). It uses linear programming (LP) to determine the relative efficiencies of a set of homogeneous (comparable) units. This is a non-parametric technique, used for performance measurement and benchmarking the model solutions computed by the MIP solver (viewed as units). The analysis is based on the determination of the technical efficiency frontier as the frontier (envelope) representing the best performance and is made up of the units (formulations) in the data set which are most efficient in transforming their inputs (computational resources) into outputs (optimization results).

In DEA analysis, the performance of a formulation on a particular problem instance is calculated by comparing it to the efficiency frontier directly determined from the data. An efficiency score is thus computed for each solution based on its distance from the efficient frontier determined by the envelope of data of all solutions. This analysis is based on the BCC DEA model (Banker et al., 1984) which assumes variable returns to scale input/output relationships, and produces, for each unit, an efficiency score and additional information.

For each inefficient unit, improvement targets are individuated using the concept of composite unit. The attributes of a composite unit (which is a hypothetical efficient unit) are determined by the projection of an inefficient unit to the efficiency frontier. The attributes are formed as a combination of specific efficient units, in the proportions indicated by the results of DEA analysis. Based on an analysis of the MIP solver solutions computed for each formulation, we first classify the solutions into efficient and inefficient in terms of the given set of performance indicators. Then, for all the inefficient solutions, we propose an iterative procedure in order to improve their performance. This procedure, in a first step, generates additional constraints and then calls for the MIP solver in order to solve the modified formulations. In a second step, a further DEA analysis is devoted to establish a new efficiency ranking. The iterative procedure ends after a stopping criteria. The efficient solutions obtained for the modified formulations and the values for each performance indicator are given to the dispatchers for the final selection of the train schedule to be implemented.

The fourth issue is carried out on a busy and complex Dutch railway network, with mixed traffic and several disturbances. The train types are used to model weighted delays. In the computational experiments, we study which formulations generate inefficient solutions or can be modified, with the proper addition of specific constraints. This would quantify the interplay and cross-performance of the different objective functions, and allows for an efficient multi-criteria decision analysis.

This paper is organized as follows. Section 2 provides a description of the terms used in the paper related to the investigated problem. Section 3 presents the studied formulations with a common solution space and different objective functions. Section 4 provides the methodology proposed to evaluate the performance of the best solutions provided by the different formulations. Section 5 gives a description of the computational analysis performed on real-world Dutch railway instances. Conclusions and lines for further research are described in Section 6.

2 Problem description

In its basic form a rail network is composed of stations, links and block sections separated by signals. For safety reasons, signals, interlocking and Automatic Train Protection (ATP) systems control the train traffic by imposing a minimum safety separation between trains. To this end, railway tracks are divided into block sections, which may host at most one train at a time. A detailed description of different aspects of railway signalling systems and traffic control regulations can be found e.g. in the handbook of Hansen and Pahl (2014).

A timetable describes the movements of all trains running in the network, specifying, for each train, the planned arrival/passing times at a set of relevant points (such as stations, junctions, and the exit point of the network) along its route. Train movements can be formally described as follows. The passage of a train through a particular block section is named an *operation* i . A route of a train is then an ordered sequence of operations to be processed during the service. The timing of a route specifies the start time t_i of each operation scheduled in the route.

Each operation requires a *running time*, which depends on the speed profile followed by the train while traversing the block section. A speed profile is constrained by the rolling stock characteristics (maximum speed, acceleration and braking rates), physical infrastructure characteristics (maximum allowed speed and signalling system) and driver behaviour when approaching variable signals aspects. Safety regulations impose a minimum headway separation between the trains running in the network, which translates into a minimum *headway time* between the exit of a train from a block section and the entrance of the subsequent train into the same block section. Railway timetable design usually includes *recovery times* and *buffer times* between the train routes. The recovery time is an extra time added to the travel time of a train between two stations, which corresponds to planning train speed smaller than maximum; those times can be utilized in real-time to recover from delays by running trains at maximum speed. Instead, the buffer is an extra time inserted in the timetable between consecutive train paths to prevent or reduce the propagation of train delays in the network.

Trains have planned stops at stations. The published stopping time of each train is named *dwelt time*. A train is not allowed to depart from a platform stop before the scheduled departure time published in the timetable. Trains result in delay when arriving at the platform later than their scheduled arrival time as planned in the timetable.

Timetables are designed to satisfy all traffic regulations. However, unexpected events occur during operations; an *entrance perturbation* is a set of delayed trains at their entrance in the network, due to the propagation of delays from previous areas.

In disturbed operations, conflicting situations might be solved. A *potential conflict* arises when two or more trains claim the same block section at the same time. A decision on the train ordering has to be taken and one of the trains involved has to change running, departure, and/or passing times according to the constraints of the signalling system.

A delayed train might cause many potential conflicts, as the available infrastructure has very limited possibility for overtaking, normally only at major stations, thus many trains might be slowed down or held since the subsequent block section is occupied by another train and cannot be entered. This phenomenon is known as *delay propagation*; the related propagated delays spreads the impact of a disturbance to other trains in the network, affecting greatly traffic in time and space. In general, the main goal of train scheduling is to minimize train delays while satisfying traffic regulation constraints and guaranteeing the

compatibility with the actual position of each train. The latter information enables the computation of the *release time* of each train at the start time t_0 of traffic prediction, which is the expected time each train enters its first block section under the current traffic conditions.

Real-time railway traffic management copes with temporary infeasibility by adjusting the schedule of each train, in terms of re-timing of the scheduled operations. The newly determined schedule should optimize some performance measure, related to delay propagation. There is no general agreement in the literature and practice on the objective functions that can be considered. We briefly describe the most relevant ones:

- *Tardiness* is the difference between the estimated train arrival time and the scheduled time at a relevant point in the networks. This difference can be computed with or without *initial delays*, that are the delays caused by entrance perturbations and can only be partially recovered by exploiting available recovery and buffer times. The case with initial delays is named *total delays*, while the other case is named *consecutive delays*. The latter case represents the delays caused by the resolution of potential train conflicts via the adopted dispatching measures;
- *Priority tardiness* is the previous case in which priorities between different train categories are taken into account;
- *Punctuality* reports the percentage of trains that arrive at scheduled station stops within a given threshold. Trains arriving later are considered as delayed;
- *Schedule deviation* is computed to evaluate the difference between the off-line schedule and the schedule computed in real-time. This is adopted in order to try to limit the deviation from the off-line plan;
- *Total completion* is related to the arrival time of all trains at their destination;
- *Travel time* is a measure of the time spent by all trains in the system.

All indicators related to tardiness and punctuality measures can be computed either at all stations or at end (destination) stations only, in terms of consecutive or total delays.

3 Mathematical model

The train scheduling problem can be modelled as a job shop scheduling problem with additional constraints, in which a *job* is a sequence of operations performed by a train while respecting all operational constraints. We next represent it via a MILP based on the alternative graph model of Mascis and Pacciarelli (2002). Let $G = (N, F, A)$ be the alternative graph composed of the following sets:

- $N = \{0, 1, \dots, n, *\}$ is the set of *nodes*, where nodes 0 and $*$ represent the start and the end operations of the schedule, while the other nodes are related to the operations in the schedule. To each node $i \in N$ is associated a start time t_i of operation i . By definition, the start time of the schedule is a known value, e.g. $t_0 = 0$, and the end time of the schedule is a variable t_* .
- F is the set of *fixed directed arcs* that model the sequence of operations to be executed by trains. Let $\sigma(i)$ be the operation succeeding i , each fixed directed arc $(i, \sigma(i)) \in F$ has a length $w_{i\sigma(i)}^F$, representing the minimum running, dwell or release time of operation i before operation $\sigma(i)$ can begin, such that $t_{\sigma(i)} \geq t_i + w_{i\sigma(i)}^F$. In particular, the release times are represented by an arc $(0, i) \in F$.

- A is the set of *alternative pairs* of directed arcs that model the train sequencing decision when a potential conflict arises. To each alternative pair $((i, j), (h, k)) \in A$ belongs two arcs whose lengths are w_{ij}^A and w_{hk}^A . The alternative arc length w_{ij}^A represents a minimum headway time between the start time t_i of i and the start time t_j of j . In particular, $w_{ij}^A(w_{hk}^A)$ can be sequence-dependent.

A *selection* S is a set of alternative arcs, at most one from each pair. A solution is a *complete* selection S^c , where an arc for each alternative pair of A is selected, and it is feasible if the connected graph (N, F, S^c) has no positive length cycles. Given a feasible schedule S^c , a timing t_i for operation i is the length of a longest path from 0 to i ($l^{S^c}(0, i)$).

The alternative graph can be viewed as a *disjunctive program*. We let X be the set:

$$X = \left\{ \begin{array}{ll} \begin{array}{l} t \geq 0 \\ x \in \{0, 1\}^{|A|} \end{array} : & \begin{array}{ll} t_{\sigma(i)} - t_i \geq w_{i\sigma(i)}^F & \forall (i, \sigma(i)) \in F, \sigma(i) \neq * \\ t_j - t_i + M(1 - x_{ijhk}) \geq w_{ij}^A & \forall ((i, j), (h, k)) \in A \\ t_k - t_h + Mx_{ijhk} \geq w_{hk}^A & \end{array} \end{array} \right\} \quad (1)$$

The variables of the problem are the following: $|N|$ real variables t_i associated with the start time of each operation $i \in N$ and $|A|$ binary variables x_{ijhk} associated with each alternative pair $((i, j), (h, k)) \in A$. The variable x_{ijhk} is 1 if $(i, j) \in S$, and $x_{ijhk} = 0$ if $(h, k) \in S$. The constant M is a sufficiently large number, e.g. the sum of all arc lengths.

In the following subsections, a set of MILP formulations with different objective functions is presented and an illustrative numerical example is reported.

3.1 Formulations

The modelling of the different objective functions requires the introduction of *due date* constraints in the MILP formulation of Section 3. In order to measure the delay of a train we use *due date arcs* in the alternative graph model. In case of total delays, a due date arc $(k, *) \in F$ is used to measure the train delay with respect to the time at which the operation k is scheduled, i.e. the arrival of a train at a scheduled stop in a station or its scheduled exit from the network. When the goal is the minimization of consecutive delays (delays caused by the rescheduling decisions necessary to solve the potential train conflicts in the network), due date arcs measures the delay with respect to the time at which the operations are scheduled but without considering the initial delays.

The length d_k of a due date arc $(k, *) \in F$ is defined in the alternative graph as follows. Let k be the arriving of a train at a relevant location where the delay is measured, δ_k its scheduled (off-line) arrival time and τ_k its earliest possible arrival time. The latter time is computed as the sum of the release time of the first operation of the train plus the minimum running and dwell times in all block sections processed by the train before operation k . When we fix $d_k = \delta_k$, we measure the total delay of the train at operation k as $t_k - \delta_k$. Since the initial delay of the train at operation k is $\max\{0, \tau_k - \delta_k\}$, we measure its consecutive delay as $d_k = \max\{\tau_k, \delta_k\}$. In what follows, we model MILP formulations with different objective functions, and we study the performance indicators introduced in Section 2.

The **MAXIMUM TARDINESS - MT** (2) minimizes the maximum (consecutive or

total) delay at all relevant locations.

$$\begin{aligned}
& \min t_* \\
& s.t. \\
& t_* - t_k \geq -d_k \quad \forall (k, *) \in F \\
& \{x, t\} \in X
\end{aligned} \tag{2}$$

The **CUMULATIVE TARDINESS - CT** (3) is the minimization of the sum of (consecutive or total) delays at all relevant locations:

$$\begin{aligned}
& \min \sum_{k=1}^{|K|} z_k \\
& s.t. \\
& z_k - t_k \geq -d_k \quad \forall (k, *) \in F \\
& z_k \geq 0 \quad \forall k \in K \\
& \{x, t\} \in X
\end{aligned} \tag{3}$$

where $K \subset N$ is the set of nodes from which a due date arc starts and z_k is a real positive variable associated with the due date arc $(k, *) \in F$.

A variant of cumulative tardiness is to compute the (consecutive or total) delay of each train with respect to the due date time of its last operation only. The last operation is associated either to the exit from the network or to a scheduled stop at the ending station. The resulting formulation is named **CUMULATIVE TARDINESS END - CTE** (4):

$$\begin{aligned}
& \min \sum_{e=1}^{|E|} z_e \\
& s.t. \\
& z_e - t_e \geq -d_e \quad \forall (e, *) \in F \\
& z_e \geq 0 \quad \forall e \in E \\
& \{x, t\} \in X
\end{aligned} \tag{4}$$

where $E \subseteq K$ is the set of nodes from which the due date arc associated with the last operation of each train starts and z_e is a real positive variable associated with the due date arc $(e, *) \in F$.

We now consider a punctuality measure by counting the number of trains that are delayed (i.e. a consecutive or total delay larger than zero) with respect to the due date time of their last operation. We call this formulation **PUNCTUALITY - P0** (5):

$$\begin{aligned}
& \min \sum_{e=1}^{|E|} v_e \\
& s.t. \\
& Mv_e - t_e \geq -d_e \quad \forall (e, *) \in F \\
& v \in \{0, 1\}^{|E|} \\
& \{x, t\} \in X
\end{aligned} \tag{5}$$

where v_e is a binary variable indicating if a train is delayed ($v_e = 1$) or not ($v_e = 0$).

In some approaches different classes of trains have different priorities (see e.g. [Carey and Kwiecinski \(1995\)](#) and [Gorman \(2009\)](#)), considering the fact that the same delay could have a different cost impact depending on the train class, e.g. international passenger, local passenger or freight trains. The next two formulations take into account priorities.

The **PRIORITY CUMULATIVE TARDINESS END - PCTE** (6) is the minimization of the sum of weighted delays with respect to the due date time of the last operations only.

The weights depend on the train class:

$$\begin{aligned}
& \min \sum_{e=1}^{|E|} f_e z_e \\
& s.t. \\
& z_e - t_e \geq -d_e \quad \forall (e, *) \in F \\
& z_e \geq 0 \quad \forall e \in E \subset K \\
& \{x, t\} \in X
\end{aligned} \tag{6}$$

where f_e is the cost associated with the due date arc $(e, *) \in F$.

The **PRIORITY CUMULATIVE TARDINESS END COST - PCTEC** (7) extends the previous formulation with the introduction of a threshold to the delay of each train and a related penalty cost:

$$\begin{aligned}
& \min \sum_{e=1}^{|E|} (f_e z_e + c_e v_e) \\
& s.t. \\
& z_e - t_e \geq -d_e \quad \forall (e, *) \in F \\
& z_e \geq 0 \quad \forall e \in E \\
& Mv_e - t_e \geq -d_e - u_e \quad \forall (e, *) \in F \\
& v \in \{0, 1\}^{|E|} \\
& \{x, t\} \in X
\end{aligned} \tag{7}$$

where f_e is the weight associated with the due date arc $(e, *) \in F$, u_e the delay threshold related to operation e , c_e the related penalty cost and v_e is a boolean variable indicating if the associated train is delayed ($v_e = 1$) or not ($v_e = 0$) with respect to the threshold.

We consider three different classes of passenger trains: high speed, intercity and local. For all due dates belonging to the operations of each train class, we assign a weight and a threshold. In particular, high speed trains have $f^{HS} = 20/3600$ and $c^{HS} = 30$ minutes, intercity $f^{IC} = 10/3600$ and $c^{IC} = 2$ hours, local trains $f^L = 5/3600$ and $c^L = 2$ hours. Those thresholds correspond to the minimum time resulting in a monetary compensation for passengers.

All the previous formulations take into account train delays. Another interesting formulation is related to the assessment of positive and negative deviations. The two deviations correspond to study how the new schedule differs from the off-line schedule. Clearly, ahead trains have a different cost impact than delayed trains, so different costs are associated in objective function with the two kinds of deviation. The following formulation, named **SCHEDULE DEVIATION - SD** (8), measures the positive and negative deviations with respect to due date times and the positive deviations with respect to release times:

$$\begin{aligned}
& \min \sum_{k=1}^{|K|} (az_k - bp_k) + \sum_{r=1}^{|R|} aq_r \\
& s.t. \\
& z_k - t_k \geq -d_k \quad \forall (k, *) \in F \\
& z_k \geq 0 \quad \forall k \in K \\
& p_k - t_k \leq -d_k \quad \forall (k, *) \in F \\
& p_k \leq 0 \quad \forall k \in K \\
& t_r - q_r \leq w_{sr} \quad \forall (0, r) \in F \\
& q_r \geq 0 \quad \forall r \in R \\
& \{x, t\} \in X
\end{aligned} \tag{8}$$

where a and b are the costs associated with the kind of deviation considered (in our experiments $a = 1/180$ and $b = 1/360$), $K \subset N$ is the set of nodes from which due date arcs start,

$R \subset N$ is the set of nodes from which release arcs start, z_k (p_k) is a real positive (negative) variables associated with each due date $(k, *) \in F$, and q_r is a real variable associated with each release arc $(0, r) \in F$.

The **TOTAL COMPLETION - TC** (9) models the minimization of the sum of the arrival times of all trains at their exit from the network. This objective function can be viewed as the maximization of throughput. This formulation requires to fix $d_k = 0 \forall k \in K$.

$$\begin{aligned} \min & \sum_{e=1}^{|E|} z_e \\ \text{s.t.} & z_e - t_e \geq 0 \quad \forall (e, *) \in F \\ & z_e \geq 0 \quad \forall e \in E \\ & \{x, t\} \in X \end{aligned} \quad (9)$$

We also consider the minimization of the travel time of all trains in the network as a surrogate for the minimization of the energy consumption. This formulation is named **TRAVEL TIME - TT** (10):

$$\begin{aligned} \min & \sum_{g=1}^{|G|} (t_l^g - t_f^g) \\ \text{s.t.} & \\ & \{x, t\} \in X \end{aligned} \quad (10)$$

where G is the set of trains in the network, t_f^g and t_l^g and the start times of the first and the last operations (f and l) in the schedule for a train g , and $(t_l^g - t_f^g)$ is the travel time spent by train g in the network.

We are interested in minimizing the above-presented objective functions, both in case of consecutive and total delays. However, the distinction between the two cases only stands for formulations (2) and (5), for which four distinct cases must be considered: MT^C and MT^T for (2); $P0^C$ and $P0^T$ for (5). Regarding the formulations (3), (4), (6), (7), (8), (9) and (10), a distinction between the consecutive and total delay minimization is not necessary, since the only difference in the objective function value would be a constant factor (i.e. the unavoidable delay).

3.2 A numerical example

This section illustrates an example of traffic situation on a network of six block sections (numbered 1–6), three of which (3, 4, 5) are traversed in both directions, and a station (Q) with a single stop platform.

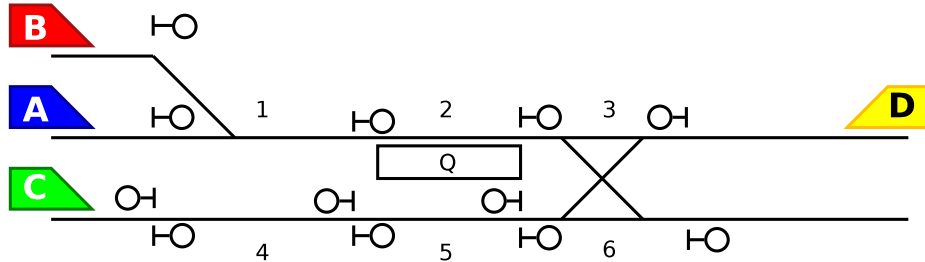


Figure 1: Example with four trains running in a small network

In Figure 1, we have 4 trains (named A, B, C, D): trains A and B follow the same route (even if they enter the area from different entrance points) traversing resources 1 and 2, stopping at station Q , and traversing resource 3; train C follows the route traversing resources 4, 5 and 3; train D follows the route traversing resources 3, 5 and 4.

Table 1 presents the numerical data of the example. Row 1 specifies which train the value refers to, Row 2 the release time at which each train enters the network, Rows 3-7 the minimum running time on the various block sections, Row 8 the minimum dwell time in station Q , Row 9 the minimum departure time from station Q , Rows 10-11 the due date times of arrival at station Q for the total and consecutive delay minimization cases, Rows 12-13 the due date times of exiting the network for the total and consecutive delay minimization cases. All trains must keep a minimum headway time of 1 unit on all resources.

Table 1: Numerical Example : Data

| Train | | A | B | C | D |
|---------------------|-----------------|-----|-----|-----|----|
| Release Time | Entrance | 60 | 45 | 45 | 60 |
| Running Time | Block Section 1 | 10 | 10 | | |
| Running Time | Block Section 2 | 10 | 10 | | |
| Running Time | Block Section 3 | 15 | 15 | 15 | 5 |
| Running Time | Block Section 4 | | | 25 | 15 |
| Running Time | Block Section 5 | | | 25 | 15 |
| Dwell Time | Station Q | 20 | 20 | | |
| Release Time | Station Q | 70 | 110 | | |
| Total Due Date Time | Station Q | 40 | 65 | | |
| Cons. Due Date Time | Station Q | 80 | 65 | | |
| Total Due Date Time | Exit | 105 | 125 | 110 | 45 |
| Cons. Due Date Time | Exit | 115 | 125 | 110 | 95 |

For sake of simplicity, the illustrative example only considers four formulations of Section 3: MT^C , MT^T , $P0^T$, PTE with $w_c = 10$ and $w_a = w_b = w_d = 1$. Table 2 reports on the four optimal solutions obtained by CPLEX for each formulation. Row 1 indicates which objective function has been minimized, Row 2 the total computation time (in seconds), Rows 3-4 if the solution is optimal (1) or not (0) and the relative gap of optimality (in percentage), Rows 5-8 the value for each performance indicator (in bold the one optimized in the corresponding formulation), Row 9 the train ordering in each solution.

Table 2: Numerical Example : Results

| Formulation | MT^C | MT^T | $P0^T$ | PCTE |
|----------------|-----------|-----------|----------|------------|
| Comp. Time (s) | 0.01 | 0.01 | 0.01 | 0.01 |
| Num. Opt. Sol. | 1 | 1 | 1 | 1 |
| Opt. Gap % | 0 | 0 | 0 | 0 |
| MT^C | 51 | 52 | 67 | 63 |
| MT^T | 101 | 52 | 117 | 101 |
| $P0^T$ | 3 | 4 | 3 | 3 |
| PCTE | 106 | 541 | 110 | 106 |
| Trains Orders | C-D-B-A | D-A-B-C | C-D-A-B | C-B-D-A |

Each formulation gives a different optimal solution and no solution outperforms the others in terms of all performance indicators. The four solutions should thus be given to the dispatcher. But how a subset of solutions can be selected? And, for each problem instance, how to combine the formulations in order to obtain better quality solutions (if any exists)?

The following section proposes a new methodology for the systematic evaluation of performance of the different formulations and for the enhancement of their performance.

4 Performance evaluation methodology

The approach proposed in this work focuses on identifying relatively efficient formulations among a set of available formulations for the real-time train rescheduling problem. A new decision support system (DSS) is developed for this purpose. In addition, the proposed DSS enables the possibility to enhance the available formulations on the basis of an efficiency assessment conducted applying an iterative procedure based on DEA. We next describe the DSS and the developed procedure.

4.1 Decision support system

Each formulation is considered as a *decision-making unit* (DMU) with multiple inputs and outputs. Then, DEA is used to evaluate the relative efficiency of the different optimization formulations. The DEA-based approach differs from methods based on statistical tests in that multiple inputs and outputs are simultaneously considered in an integrated framework for the evaluation of unit efficiency. In addition, the proposed approach gives a support in order to improve the configurations showing inefficiency. In fact, besides the identification of relatively efficient and inefficient DMUs (i.e. formulations with respect to a given instance) and the efficiency ranking of the formulations under study, DEA helps to identify also the sources and the level of inefficiency for each of the considered inputs and outputs. This translates into an iterative procedure for the generation of improved formulations via the addition of specific constraints on a number of inefficient performance indicators.

Figure 2 describes the DSS scheme, which it is applied to a set of formulations F_1, \dots, F_q . Each formulation F_i is characterized by a set of inputs I_i (representing the use of resources) and a set of outputs O_i (representing the achieved results) used by the DEA Module to elaborate the efficiency assessment. In the DEA methodology, the measurement of relative efficiency in the presence of multiple inputs and outputs is addressed by assigning weights so that the overall relative efficiency score is actually a ratio of the weighted sum of the outputs to the weighted sum of the inputs. The result of this analysis includes: a) a relative efficiency measure for each DMUs; b) the individuation of a set of inefficient DMUs; c) the determination of a *virtual composite* efficient DMU for each inefficient DMU; d) indications on how to drive improvements for the inefficient DMUs.

These results are the inputs for another module of the proposed DSS, namely the **Formulation Enhancement Module** which suggests to modify the set of constraints of the evaluated formulations in order to improve their performance. The set of DMUs is updated with the new enhanced formulations and proposed to the DEA Module for a new efficiency analysis. At end of this iterative process, a final set of enhanced formulations F_{e1}, \dots, F_{eq} and the correspondent solutions S_{e1}, \dots, S_{eq} are returned by the DSS. A detailed description of the DEA literature, of the DEA model adopted for the DEA Module and of the phases of the overall process of Figure 2 is the argument of the following sections.

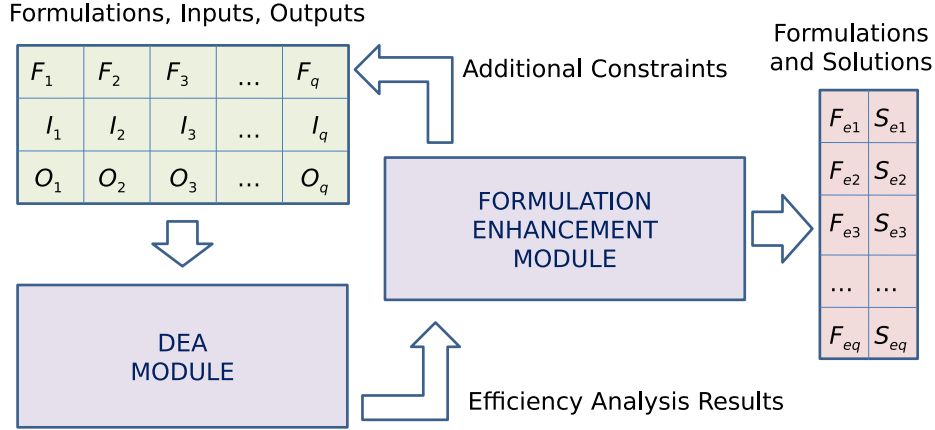


Figure 2: The general scheme of the DSS

4.2 DEA evaluation

Setting the best formulations in use should be considered as an integral part of an advanced optimization system and plays a crucial role in the success of a modern planning and scheduling system. For the sake of simplicity and without loss of generality, we consider a formulation of high quality if it effectively and efficiently solves the problem of interest. To help to decide efficient configurations of optimization approaches, the literature recently proposed to use the well-known DEA method, which is based on LP and has been widely adopted in the non-parametric performance evaluation of several systems and organizations. Those previous studies applied to measure empirically the production efficiency of a set of DMUs with multiple inputs and outputs. Also, the feature of handling the multiple inputs and outputs of DMUs makes DEA an attractive alternative for evaluating the algorithmic performance of different heuristics or exact algorithms as well as those of different algorithmic configurations, because researchers and practitioners may desire to compare several inputs and outputs at the same time.

Although other techniques (typically based on design of experiments (DOEs), analysis of variance (ANOVA), and paired statistical tests) help to individuate best configurations to arrive at conclusions that have a statistical meaning, conducting the analysis becomes very difficult when the numbers of distinct parameter values and evaluation criteria are large. Furthermore, parametric methods (e.g. using DOEs and ANOVA) require to check some specific statistical hypotheses. On the other hand, when the differences of the configurations in evaluation criteria are small, the results might not be statistically significant. Moreover, given selected inputs and outputs, DEA analysis requires to manage a set of LP problems which can be easily solved by a standard software.

This paper considers each formulation as a DMU and applies DEA to assess the relative efficiency of a set of formulations tackling the same optimization problem under investigation. The proposed DEA approach differs from classical empirical analysis methods in that

multiple inputs and outputs of the unit configurations under evaluation are simultaneously taken into account in an integrated framework rather than being separately and sequentially compared. Different DEA models have been proposed in the literature (see e.g. Charnes et al. (1994), Cooper et al. (2007)). Based on the assumption of *constant returns-to-scale* (CRS), Charnes et al. (1978) developed a DEA model using LP, named the CCR (*Charnes Copper Rhodes*) model. That model is based on the choice of input and output weight for each DMU. An optimal solution to the CCR model assigns the input and output weights which maximize the efficiency of the evaluated DMU, relative to the other DMUs. The model is solved in turn once for each DMU, to obtain the relative efficiencies of all the DMUs under evaluation. To relax the strict CRS assumption, Banker et al. (1984) proposed a generalization of the CCR model named the BCC model, which allows *variable returns-to-scale* (VRS) on production frontiers. Both CCR and BCC models can be adapted to measure the relative efficiency of different unit configurations.

Lu and Yu (2012) adopted the CCR model of DEA to evaluate relative efficiencies of a set of algorithms for a variant of the vehicle routing problem. Dellino et al. (2009) use the same approach in the configuration of a metamodel-assisted engineering design optimization. The CCR model is often criticized due to the assumption of CRS, which might not always hold to the inputs to and/or outputs from an algorithm for solving optimization problems. For instance, more computational time for executing an algorithm does not guarantee to obtain a better objective value. In the evaluation of algorithmic efficiency, Lu and Wu (2012) applied the BCC model which allows VRS on production frontiers. Their work shows how the relative efficiency of each configuration can be determined using DEA, helping to easily determine the best configuration or the set of efficient configurations of an algorithm. In addition, it compares the evaluation results of the CCR and BCC models for determining efficient algorithmic configurations. In general, BCC seems to be more adequate for the decision-maker who needs to analyze the performance of optimization approaches, identifying efficient unit configurations, ranking them effectively, and using the analysis results to give indications on how to improve inefficient cases.

4.3 The adopted DEA model

The CCR model (by Charnes et al. (1978)) is considered as the origin of many following ideas and models in DEA literature. The mathematical formulations and summaries adopted in this paper are based on Cooper et al. (2007). We suppose that there are q DMUs representing q different optimization problem formulations: $DMU_1, DMU_2, \dots, DMU_q$. Suppose there are m inputs and s outputs for each one of them. For DMU_j the inputs and outputs are represented by $(w_{1j}, w_{2j}, \dots, w_{mj})$ and $(y_{1j}, y_{2j}, \dots, y_{sj})$, respectively. For each DMU_o , DEA basically tries to maximize the ratio θ_o representing the efficiency, according to the following fractional model:

$$\begin{aligned} \max \theta_o &= \frac{(u_1 y_{1o} + \dots + u_s y_{so})}{(v_1 w_{1o} + \dots + v_m w_{mo})} \\ s.t. \quad &\frac{(u_1 y_{1o} + \dots + u_s y_{so})}{(v_1 w_{1o} + \dots + v_m w_{mo})} \leq 1 \\ &v_1, v_2, \dots, v_m \geq 0; u_1, u_2, \dots, u_s \geq 0 \end{aligned} \tag{11}$$

For each DMU involved in the performance evaluation process, DEA provides an efficiency score between 0 and 1. This efficiency score for a DMU is determined by computing the

ratio of cumulative weighted outputs to cumulative weighted inputs for it. DEA enables variable weights, which are calculated in such a way that the efficiency score for the DMU is maximized. For a DEA analysis with q different DMUs, q different optimization problems are solved to compute the efficiency scores of each of the DMUs. The basic (fractional) formulation (11) can be easily transformed into a LP model, while the input and output data can be arranged in matrix notation W and Y , respectively. Based on these input and output matrices, the DEA model can be rewritten in the following *envelopment form* which is expressed with a variable θ and a non-negative vector of $\mathbf{h} = (h_1, \dots, h_q)^T$:

$$\begin{aligned}
& \min \theta \\
& s.t. \\
& \theta \mathbf{w}_o - W\mathbf{h} \geq \mathbf{0} \\
& Y\mathbf{h} \geq \mathbf{y}_o \\
& \mathbf{h} \geq \mathbf{0}
\end{aligned} \tag{12}$$

In model (12), the objective is to guarantee at least the output levels expressed by \mathbf{y}_o of DMU_o in all dimensions while reducing the input vector \mathbf{w}_o proportionally as much as possible. An efficient DMU_o has a θ_o equal to 1, while for an inefficient DMU_o , its *reference set* RS_o is defined as follows: $RS_o = \{j | h_j^* > 0\} (j = 1, \dots, q)$.

The rationale behind DEA technique is that if a given DMU is capable of producing a level of outputs using a certain amount of inputs, then other DMUs should also be able to do the same if they were to operate efficiently. The DMUs can then be combined to form a target DMU with composite inputs and composite outputs. Such an unit does not necessarily exist and it is typically indicated as virtual composite DMU (DMU_{VC}). The heart of the analysis lies in finding the *best virtual unit* for each real DMU. If DMU_{VC} is better than the original DMU under study by either making more output with the same input or making the same output with less input then the original DMU is inefficient. Even more importantly, considering the envelopment form, at the optimum, the constraints give indication about both the input excesses and the output shortfalls, in such a way an inefficient DMU can be improved correspondingly. The reference set allows to determine the DMU_{VC} obtained as a linear combination of the elements in RS_o using \mathbf{h}^* as coefficients.

The presented CCR models deal with trying to reduce input variables while attaining at least the provided output levels. There is another type of model aiming to maximize output levels while spending no more than existing resources or inputs. This kind of models is referred as output-oriented. A CCR output-oriented model is formulated (in its envelopment form) as follow: In DEA, θ^* indicates the input reduction rate whereas η^* represents the output enlargement rate (Cooper et al., 2007). It is clear that the less the η^* value, the more efficient the DMU, or vice versa. In order to obtain an efficiency score of between 0 and 1 and to relate the efficiency scores with the input oriented model, $1/\eta^*$ is used to express the efficiency score of the DMU in an output oriented model.

The model orientation does not change the set of efficient DMUs; however the possible improvements and the composite virtual DMUs are calculated with different \mathbf{h} values. The CRS assumption of CCR models typically is oversimplified for several real world problems and specifically for optimization approaches. In fact, in the practice of algorithmic design it is difficult to obtain an increase/decrease in outputs proportional to an augmentation/reduction in input variables. To this aim, the BCC model firstly proposed by Banker et al. (1984) allows the consideration of more useful VRS. The BCC model differs from the presented CCR model for a convexity condition in its constraints. Moreover, in order to an-

alyze the performance of different formulations, the output-orientation appears to be more adequate. We adopt the following envelopment form, where \mathbf{e} indicates the unit vector:

$$\begin{aligned}
& \max \eta \\
& s.t. \\
& \mathbf{w}_0 - W\mathbf{h} \geq \mathbf{0} \\
& \eta \mathbf{y}_0 - Y\mathbf{h} \leq \mathbf{0} \\
& \mathbf{e}\mathbf{h} = 1 \\
& \mathbf{h} \geq \mathbf{0}
\end{aligned} \tag{13}$$

In order to prepare data for the DEA analysis proposed in this paper, the DSS scheme contains a *pre-processing phase* based on Sarkis (2007), including scaling and normalization data and a correlation analysis.

4.4 Formulation enhancement procedure

For the train rescheduling problem under study, each considered formulation F_i is associated to a DMU in the DEA analysis. The set of inputs I_i associated to a F_i is limited to the computation time, while the set of outputs O_i includes: a) the value of the performance index related to the best solution S_i attained by F_i ; b) the gap of optimality showed by F_i ; c) the optimality of the solution S_i (through a binary index); d) the $q-1$ evaluations of S_i with the performance indicators (i.e. objective functions) of formulations F_j with $j \neq i$. The DEA analysis gives for each DMU (i.e. each formulation F_i) an assessment of its relative efficiency represented by η (or θ). If F_i is not fully efficient, the method individuates a reference set RS_i of efficient DMUs which can be used to obtain a virtual composite unit as a benchmark for F_i . This DMU_{VC} is determined as a linear combination of the units in RS_i , being \mathbf{h}^* the vector of coefficients. In more detail, this linear combination involves the outputs of the reference being $\mathbf{y}_{VC} = \mathbf{h}^{*T} \mathbf{Y}$.

For each inefficient F_i a formulation enhancement procedure is executed. This procedure compares each output of F_i (i.e. $\mathbf{y}_i = (y_{1i}, y_{2i}, \dots, y_{si})$) with the correspondent output of the DMU_{VC} and individuates output improvements to drive F_i towards efficiency. The output improvements refer to the evaluation of S_i with the different performance indicators and are used to modify F_i by introducing constraints on the inefficient performance indicators. Each additional constraint requires to F_i an improvement with respect to a specific performance index (i.e. to increase the correspondent output). These constraints are determined by all the relations in $\mathbf{y}_i \leq \mathbf{y}_{VC}$ violated in the evaluations of the solution S_i (i.e. the train order obtained by F_i) with respect to the performance indicators of F_j with $j \neq i$. The procedure is executed iteratively by updating at each step the set of formulations via the improvements suggested by the DEA analysis. The stopping criteria is reached either when all DMUs are fully efficient or when inefficient DMUs are not improvable.

4.5 Numerical example

We use the example of Section 3.2 to better explain the DSS work. We evaluate the four formulations with the DEA Module, considering inputs and outputs according to Section 4.4. As a result, the solutions found by $P0^T$ and $PCTE$ are shown to be inefficient and the DMU_{VC} is represented by MT^C in both cases (i.e. the reference set contains MT^C only).

Comparing the values of each performance indicator, the solution resulting from MT^T is efficient, since MT^T is by far the most difficult indicator to take into account indirectly. The Formulation Enhancement Module suggests the following additional constraints: PCTE is required to add a constraint regarding MT^C (≤ 51); and $P0^T$ three constraints regarding MT^C (≤ 51), MT^T (≤ 101) and PCTE (≤ 106). The solutions returned by the enhanced PCTE and $P0^T$ formulations are the same found by MT^C . All four formulations are now efficient and the final solutions presented by the DSS to the dispatcher are the trains orders of MT^C and MT^T solutions.

5 Computational experiments

This section presents the computational results on the formulations and DEA techniques introduced in this paper. The tests have been performed in a laboratory environment on practical-size instances of a Dutch railway network. The optimal MILP solutions are computed via the solver IBM ILOG CPLEX MIP 12.0. The experiments are executed on a workstation Power Mac with processor Intel Xeon E5 quad-core (3.7 GHz), 12 GB of RAM.

5.1 Test case description

The infrastructure considered is a large part of the railway network in the east of the Netherlands. As shown in Figure 3, the network is composed of major stations with complex interlocking systems and dense traffic, including Utrecht Central, Geldermalsen, Nijmegen, Arnhem and Den Bosch. All minor stations are also considered in our model.

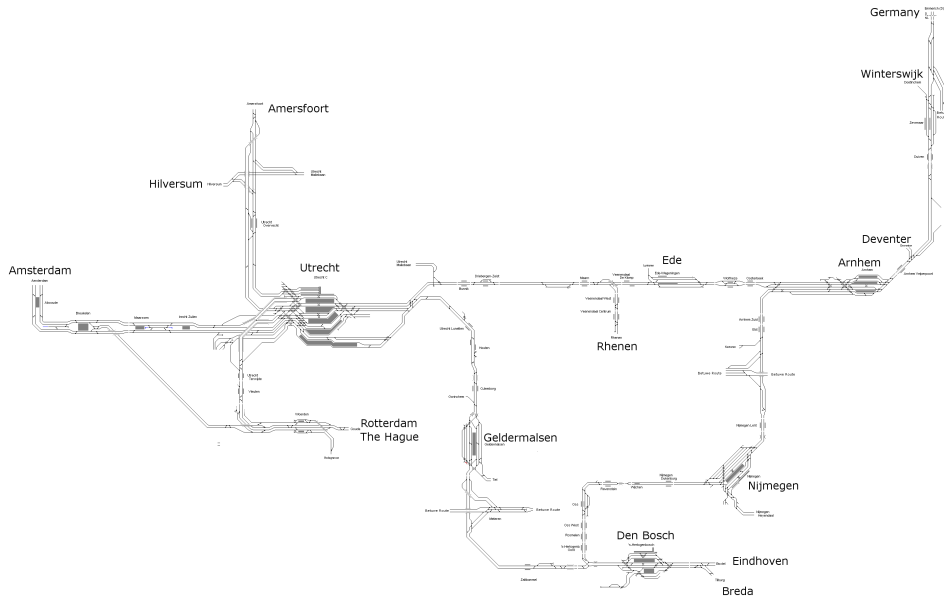


Figure 3: The considered Dutch railway network

The two main traffic directions are served by the line between Utrecht and Arnhem

(towards Germany) and the line between Utrecht and Den Bosch (from Amsterdam towards Eindhoven and the southern part of the country). The network comprises a combination of single/double-tracks of different length, with a maximum distance between area borders of around 300 km. In total, there are more than 1000 block sections and 200 stopping locations.

For the computational experiments, we use the 2008 timetable that is cyclic with a cycle length of one hour, though most intercity and local trains services operate up to 4 times per hour per direction. There are 154 trains running each hour: 80 are local trains, 70 are intercity trains, and 4 are high speed trains.

The various train lines are reported in Figure 4, each thick line (solid or dotted) represents a train line with 4 trains running per hour per direction, each thin line represents a train line with 2 trains running per hour per direction (i.e. with a half-hour frequency).

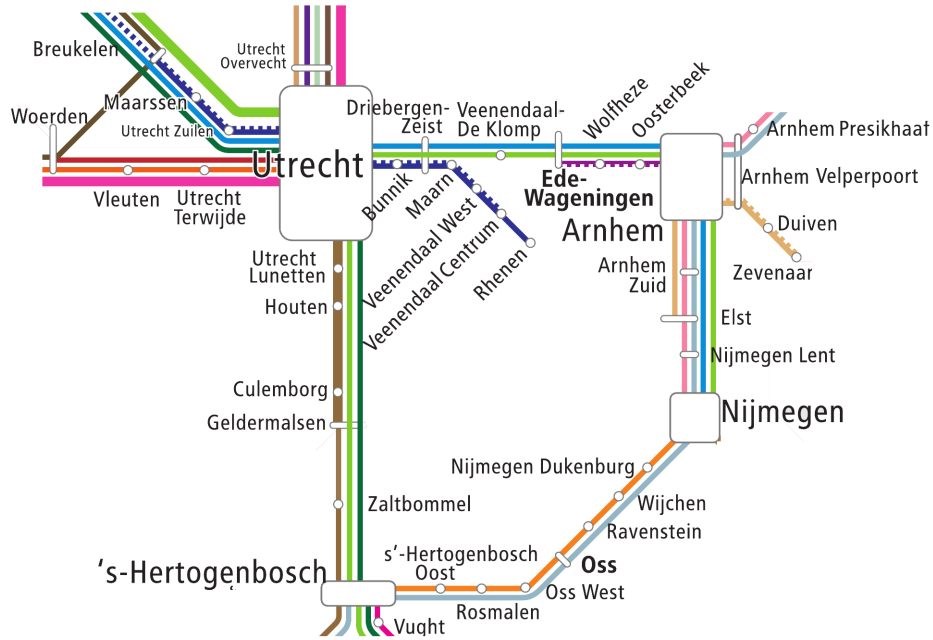


Figure 4: Train traffic flows on the network (source: treinreiziger.nl)

We generated timetable perturbations that cause initial delays in the network. Random variations of the entrance times are applied to all trains running in the network. The real-life stochasticity of train operations is modeled by Weibull distributions, as reported in [Corman et al. \(2011B\)](#). We consider two sets of delay configurations, as used in [Corman et al. \(2014A\)](#): 5 “normal” entrance perturbations that correspond to the fitted parameters, and 5 “increased” entrance perturbations that are obtained by using the scale and shift parameter according to the same fitting, and doubling the shape parameter of the Weibull distributions. The latter perturbations result in an increased variability of train delays.

For each perturbation, we deal with two types of instances: 30-minute traffic predictions (99 trains), that are considered *practical-size instances*, and 60-minute traffic predictions (154 trains) that are *large-size instances*, generated to study the computational limits of our approach. For the large-size instances, we use CPLEX with one-hour computation time.

5.2 Performance evaluation

Tables 3 and 4 report on the first and last iterations of the DEA framework for the practical-size and large-size instances, respectively. In each table, we performed 110 runs of CPLEX per iteration: 10 delay configurations multiplied by 11 formulations. Specifically, each row of the tables presents average results on all the delay configurations. The results of the first/last iterations are provided in both the tables with a 14-row information: Row 1 gives the average computation time (in seconds), Row 2 the number of problems that were solved to optimality by CPLEX, Row 3 the average optimality gap (in percentage), Rows 4-14 the average value of each performance indicator. For each column (i.e. for each formulation), the average value of the optimized performance indicator is highlighted in bold, while the improvements of the last iteration versus the first iteration are highlighted in italic.

Table 3: Computational results for practical-size instances

| Formulation | MT ^C | CT | CTE | P0 ^C | PCTE | PCTEC | SD | TC | TT | MT ^T | P0 ^T |
|-----------------|-----------------|-------------|------------|-----------------|------------|---------------|------------|---------------|--------------|-----------------|-----------------|
| First Iteration | | | | | | | | | | | |
| Comp. Time (s) | 3.3 | 11.9 | 8.9 | 5.3 | 9.1 | 7.6 | 9.1 | 7.3 | 14.0 | 1.8 | 3.5 |
| Num. Opt. Sol. | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| Opt. Gap % | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MT ^C | 220 | 290 | 245 | 530 | 278 | 284 | 287 | 264 | 264 | 414 | 604 |
| CT | 4485 | 2486 | 3032 | 5519 | 3668 | 3696 | 2568 | 3244 | 3237 | 7537 | 9300 |
| CTE | 1559 | 1116 | 954 | 1735 | 1058 | 1058 | 1207 | 1032 | 1032 | 2481 | 2917 |
| P0 ^C | 16.9 | 14.5 | 14.5 | 11.7 | 14.6 | 14.6 | 14.7 | 15.0 | 15.0 | 18.3 | 16.5 |
| PCTE | 3.4 | 2.8 | 2.3 | 3.7 | 2.2 | 2.2 | 3.1 | 2.5 | 2.5 | 5.0 | 6.2 |
| PCTEC | 3.4 | 2.8 | 2.3 | 3.7 | 2.2 | 2.2 | 3.1 | 2.5 | 2.5 | 5.0 | 6.2 |
| SD | 192 | 171 | 179 | 203 | 185 | 185 | 170 | 181 | 181 | 222 | 242 |
| TC | 513825 | 513449 | 513320 | 514166 | 513440 | 513440 | 513606 | 513247 | 513247 | 514716 | 515084 |
| TT | 91422 | 91046 | 90917 | 91763 | 91037 | 91037 | 91203 | 90844 | 90844 | 92313 | 92681 |
| MT ^T | 881 | 917 | 881 | 1011 | 893 | 893 | 917 | 881 | 881 | 872 | 1033 |
| P0 ^T | 70 | 70 | 70 | 69 | 70 | 70 | 70 | 70 | 70 | 71 | 66 |
| Last Iteration | | | | | | | | | | | |
| Comp. Time (s) | 3.4 | 11.9 | 8.9 | 5.3 | 9.1 | 8.3 | 9.1 | 7.3 | 14.3 | 2.9 | 3.5 |
| Num. Opt. Sol. | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| Opt. Gap % | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MT ^C | 220 | 290 | 245 | 530 | 278 | 275 | 287 | 264 | 263 | <i>313</i> | 604 |
| CT | <i>4381</i> | 2486 | 3032 | 5519 | 3668 | <i>3629</i> | 2568 | 3244 | <i>3229</i> | <i>6125</i> | 9300 |
| CTE | <i>1512</i> | 1116 | 954 | 1735 | 1058 | 1058 | 1207 | 1032 | 1032 | <i>2051</i> | 2917 |
| P0 ^C | <i>16.5</i> | 14.5 | 14.5 | 11.7 | 14.6 | 14.6 | 14.7 | 15.0 | 15.0 | <i>16.5</i> | 16.5 |
| PCTE | 3.4 | 2.8 | 2.3 | 3.7 | 2.2 | 2.2 | 3.2 | 2.5 | 2.5 | <i>4.2</i> | 6.2 |
| PCTEC | 3.4 | 2.8 | 2.3 | 3.7 | 2.2 | 2.2 | 3.2 | 2.5 | 2.5 | <i>4.2</i> | 6.2 |
| SD | <i>191</i> | 171 | 179 | 203 | 185 | <i>184</i> | 170 | 181 | 181 | <i>208</i> | 242 |
| TC | <i>513780</i> | 513449 | 513320 | 514166 | 513440 | <i>513438</i> | 513606 | 513247 | 513247 | <i>514350</i> | 515084 |
| TT | <i>91377</i> | 91046 | 90917 | 91763 | 91037 | <i>91035</i> | 91203 | 90844 | 90844 | <i>91947</i> | 92681 |
| MT ^T | 881 | 917 | 881 | 1011 | 893 | 893 | 917 | 881 | 881 | 872 | 1033 |
| P0 ^T | 70 | 70 | 70 | 69 | 70 | 70 | 70 | 70 | 70 | 70 | 66 |

For the practical-size instances of Table 3, CPLEX takes on average up to 15 seconds to compute the optimal solution for each formulation and the DSS requires at most 3 iterations in order to converge, i.e. no more constraints can be added to improve efficiency. Overall, the DSS is therefore very quick to compute efficient solutions. Regarding the large-size instances of Table 4, CPLEX is not always able to compute an optimal solution within 1 hour of computation, except for MT^C, MT^T and P0^T. As for the other set of instances, the DSS requires at most 3 iterations in order to converge.

Table 4: Computational results for large-size instances

| Form. | MT ^C | CT | CTE | P0 ^C | PCTE | PCTEC | SD | TC | TT | MT ^T | P0 ^T |
|-----------------|-----------------|-------------|-------------|-----------------|------------|------------|------------|---------------|---------------|-----------------|-----------------|
| First Iteration | | | | | | | | | | | |
| Comp. Time (s) | 914.4 | 1022.6 | 1182.3 | 411.8 | 924.1 | 900.6 | 1376.9 | 874.8 | 1160.9 | 85.0 | 365.5 |
| Num. Opt. Sol. | 10 | 8 | 7 | 9 | 8 | 7 | 7 | 8 | 7 | 10 | 10 |
| Opt. Gap % | 0 | 10.3 | 18.9 | 0.4 | 11.7 | 11.1 | 11.4 | 0 | 0.3 | 0 | 0 |
| MT ^C | 313 | 389 | 418 | 840 | 499 | 497 | 441 | 424 | 427 | 595 | 733 |
| CT | 14789 | 7760 | 10079 | 16556 | 12495 | 12085 | 8368 | 9971 | 10143 | 21393 | 24697 |
| CTE | 5178 | 3081 | 2619 | 5484 | 2981 | 2906 | 3406 | 2673 | 2699 | 6121 | 7115 |
| P0 ^C | 37.0 | 28.6 | 28.1 | 22.6 | 29.3 | 28.9 | 29.4 | 28.4 | 28.9 | 37.8 | 33.6 |
| PCTE | 12.2 | 8.6 | 6.8 | 14.5 | 6.1 | 6.0 | 9.5 | 6.9 | 6.8 | 13.2 | 15.6 |
| PCTEC | 12.2 | 8.6 | 6.8 | 14.5 | 6.1 | 6.0 | 9.5 | 6.9 | 6.8 | 13.2 | 15.6 |
| SD | 437 | 370 | 399 | 458 | 423 | 419 | 371 | 397 | 399 | 505 | 539 |
| TC | 965034 | 962888 | 962433 | 965175 | 962840 | 962813 | 963308 | 962327 | 962384 | 965918 | 966716 |
| TT | 186553 | 184407 | 183952 | 186695 | 184360 | 184332 | 184827 | 183846 | 183903 | 187437 | 188235 |
| MT ^T | 1017 | 1034 | 1028 | 1349 | 1057 | 1057 | 1088 | 1039 | 1039 | 1004 | 1221 |
| P0 ^T | 112 | 109 | 109 | 107 | 109 | 110 | 110 | 109 | 109 | 112 | 104 |
| Last Iteration | | | | | | | | | | | |
| Comp. Time (s) | 920.5 | 1238.8 | 1243.9 | 416.4 | 1644.1 | 930.3 | 1850.2 | 1025.8 | 918.7 | 121.6 | 365.5 |
| Num. Opt. Sol. | 10 | 8 | 5 | 8 | 8 | 7 | 6 | 7 | 7 | 10 | 10 |
| Opt. Gap % | 0 | 12.2 | 20.4 | 2.7 | 11.2 | 11.8 | 7.9 | 0.0 | 0.2 | 0 | 0 |
| MT ^C | 313 | 383 | 403 | 734 | 498 | 491 | 435 | 425 | 415 | 553 | 733 |
| CT | 14638 | 7894 | 10154 | 15465 | 12047 | 12120 | 7635 | 10012 | 9023 | 18680 | 24697 |
| CTE | 5075 | 3067 | 2650 | 4974 | 2945 | 2913 | 2997 | 2668 | 2433 | 5495 | 7115 |
| P0 ^C | 36.3 | 29.0 | 28.0 | 23.1 | 29.3 | 28.9 | 28.7 | 28.5 | 28.0 | 34.8 | 33.6 |
| PCTE | 12.1 | 8.4 | 6.8 | 13.4 | 6.1 | 6.0 | 8.4 | 6.8 | 6.5 | 12.3 | 15.6 |
| PCTEC | 12.1 | 8.4 | 6.8 | 13.4 | 6.1 | 6.0 | 8.4 | 6.8 | 6.5 | 12.3 | 15.6 |
| SD | 437 | 371 | 400 | 449 | 418 | 419 | 360 | 398 | 382 | 479 | 539 |
| TC | 964915 | 962910 | 962455 | 964680 | 962811 | 962804 | 962624 | 962336 | 961883 | 965274 | 966716 |
| TT | 186434 | 184429 | 183974 | 186199 | 184330 | 184323 | 184453 | 183855 | 183712 | 186794 | 188235 |
| MT ^T | 1017 | 1022 | 1022 | 1249 | 1051 | 1051 | 1060 | 1034 | 1011 | 1004 | 1221 |
| P0 ^T | 112 | 110 | 109 | 107 | 109 | 110 | 110 | 109 | 109 | 112 | 104 |

When looking at the number of efficient solutions for each instance, the DEA Module delivers, on average, 8 different efficient solutions. The Formulation Enhancement Module is able to improve several performance indicators, but some other indicators decrease their performance, thus yielding to trade-off solutions. In some particular cases, the addition of constraints deteriorates the performance of some formulation in terms of its objective function, because the set of feasible solutions decreases for the enhanced formulation and the optimal solution of the original formulation is discarded by the solver.

Table 5 reports average information on the DEA evaluation performed by the DSS. The evaluation is given both for practical-size and large-size instances. Each row presents the average results obtained on the 10 delay configurations. Row 1 gives the cumulative number of instances for which each formulation results to be efficient at the first, second and third iterations; Row 2 counts how many times a DMU is in some reference set RS_i , i.e. how many times it has been used in a DMU_{VC} (at first, second and third iterations); Row 3 returns the average number of constraints added to each formulation at each iteration.

From Table 5, we have the following observations. The efficiency found at the first iteration is often improved at the next iterations (4 formulations are improved for the practical-size instances and 5 formulations are improved for the large-size instances). This is a confirmation that the additional constraints are very useful to improve the formulations and their solution quality in terms of multiple performance indicators.

Table 5: Cumulative DEA evaluation

| Formulation | MT ^C | CT | CTE | P0 ^C | PCTE | PCTEC | SD | TC | TT | MT ^T | P0 ^T |
|--------------------------|-----------------|-------|-------------|-----------------|---------|---------|-----------|---------|-----------|-----------------|-----------------|
| Practical-Size Instances | | | | | | | | | | | |
| Efficient cases | 9 10 10 | 6 6 6 | 8 8 8 | 10 10 10 | 9 9 9 | 4 4 5 | 9 9 9 | 8 8 8 | 2 3 3 | 6 9 9 | 10 10 10 |
| $\in RS_i$ | 0 0 0 | 1 1 1 | 4 4 6 | 1 0 0 | 11 9 5 | 3 1 1 | 6 3 3 | 6 5 5 | 0 0 0 | 0 0 2 | 1 0 0 |
| Added Constr. | 8 0 0 | 0 0 0 | 0 0 0 | 0 0 0 | 0 0 0 | 3 2 0 | 0 0 0 | 0 0 0 | 2 0 0 | 9 0 0 | 0 0 0 |
| Large-Size Instances | | | | | | | | | | | |
| Efficient cases | 8 9 9 | 8 8 8 | 6 6 6 | 8 8 8 | 7 8 8 | 4 6 6 | 7 7 7 | 6 7 7 | 4 4 4 | 6 9 9 | 10 10 10 |
| $\in RS_i$ | 2 0 0 | 4 3 4 | 4 2 2 | 1 1 1 | 5 4 4 | 9 5 5 | 6 6 6 | 8 8 8 | 1 1 1 | 8 8 8 | 2 0 0 |
| Added Constr. | 7 0 0 | 6 0 0 | 3.3 3.5 3.5 | 9 9 0 | 2.7 7 7 | 3.3 3 3 | 4.3 8 6.5 | 2.3 4 4 | 2.3 3.3 3 | 9.5 9 9 | 0 0 0 |

Regarding the specific formulations, P0^T is already efficient for all the instances, since the other formulations have a poor performance for this performance indicator. TT has the smallest number of efficient solutions, since competitive solutions are found by the other formulations in a shorter computation time. Also, MT^T has a high number of inefficient solutions, however these are often improved by the iterative procedure. This is due to the nature of the formulation: only a small number of trains have effects in this objective function and different optimal solutions exist. Furthermore, MT^T is quite often improved and presents the largest number of average constraints added during the iterative process. The improvement of MT^T is also visible when looking at the specific performance indicators in Tables 3 and 4. The other formulations are improved for similar reasons.

6 Conclusions and future research

This paper presents a multi-criteria decision support system to help dispatchers in taking more informed decisions when dealing with real-time traffic disturbances. We are particularly interested in the computation of near-optimal solutions in terms of a set of performance indicators. This is achieved by the development of MILP formulations based on the alternative graph model, each one taking into account multiple performance indicators, either in the objective function or in the problem constraints. An iterative DEA-based procedure is proposed to establish an efficient-inefficient classification of the formulations and to improve inefficient formulations. For each tested instance, the procedure converges after a limited number of iterations and returns a set of efficient formulations and their best solutions. The final selection of the train schedule to be implemented is left to the dispatchers.

Experiments are performed for practical-size and large-size instances from a Dutch railway network with mixed traffic and various sources of disturbance. We study which formulations are efficient or can be modified to be efficient, with the proper addition of selected linear constraints. The proposed multi-criteria decision support system is shown to be able to improve the inefficient formulations, and to deliver of a pool of improved formulations and their solutions in a short computation time for practical-size instances.

The insights of the proposed approach regard the investigation of different objective functions and constraints and the identification of the most representative formulations for the computation of good trade-off solutions. On-going research is dedicated to a more comprehensive evaluation of alternative formulations and DEA classifications. The long term contribution to the practice of dispatching is the possibility to automatically generate a combination of objective functions and constraints such that the most efficient solutions can be quickly identified, classified and delivered to the dispatchers.

References

- Banker, R., A. Charnes, W.W. Cooper (1984) Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management Science* **30** 1078–1092.
- Binder, S., Chen, J., Bierlaire, M. (2014) Generation and evaluation of passenger-oriented railway disposition timetables, Proceedings of the 14th Swiss Transport Research Conference, Ascona, Switzerland.
- Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L., Wagenaar, J. (2014) An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B* **63** (1) 15–37.
- Caimi, G., Fuchsberger, M., Laumanns, M., Lüthi M. (2012) A model predictive control approach for discrete-time rescheduling in complex central railway station areas. *Computers and Operations Research* **39** (11) 2578–2593.
- Caprara, A., Kroon, L.G., Monaci, M., Peeters, M., Toth, P. (2006) Passenger Railway Optimization. In: C. Barnhart & G. Laporte (Eds.), *Handbooks in Operations Research and Management Science*, vol. 14, pp. 129–187, Elsevier, Amsterdam, The Netherlands.
- Carey, M., Kwiecinski, A. (1995) Properties of expected costs and performance measures in stochastic models of scheduled transport. *European Journal of Operational Research* **83** (1) 182–199.
- Charnes, A., Cooper, W.W., Lewin, A.Y., Seiford, L.M. (1994) *Data Envelopment Analysis: Theory, Methodology, and Application*. Kluwer Academic Publishers, Dordrecht, The Netherlands.
- Charnes, A., Cooper, W.W., Rhodes, E. (1978) Measuring the efficiency of decision making units. *European Journal of Operational Research* **2** (6) 429–444.
- Cooper, W.W., Seiford, L.M., Tone, K. (2007) *Data Envelopment Analysis: A Comprehensive Text with Models, Applications, References and DEA-Solver Software*, 2nd edn., Springer, New York, USA.
- Corman, F., D’Ariano, A., Pacciarelli, D., Pranzo, M. (2009) Evaluation of green wave policy in real-time railway traffic management. *Transportation Research Part C* **17** (6) 607–616.
- Corman, F., D’Ariano, A., Hansen, I.A., Pacciarelli, D. (2011A) Optimal multi-class rescheduling of railway traffic. *Journal of Rail Transport Planning & Management* **1** (1) 14–24.
- Corman, F., D’Ariano, A., Pranzo, M., Hansen, I.A. (2011B) Effectiveness of dynamic reordering and rerouting of trains in a complicated and densely occupied station area. *Transportation Planning and Technology* **34** (4) 341–362.
- Corman, F., D’Ariano, A., Pacciarelli, D., Pranzo, M. (2014A) Dispatching and coordination in multi-area railway traffic management. *Computers and Operations Research* **44** (1) 146–160.
- Corman, F., D’Ariano, A., Hansen, I.A. (2014B) Evaluating disturbance robustness of railway schedules, *Journal of Intelligent Transport Systems: Technology, Planning, and Operations* **18** (1) 106–120.
- D’Ariano, A., Pacciarelli, D., Pranzo, M. (2007) A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research* **183** (2) 643–657.
- D’Ariano, A. (2009) Innovative Decision Support System for Railway Traffic Control. *IEEE Intelligent Transportation Systems Magazine* **1** (4) 8–16.

- [D'Ariano, A., Samà, M., D'Ariano, P., Pacciarelli, D. \(2014\) Evaluating the applicability of advanced techniques for practical real-time train scheduling. *Transportation Research Procedia* **3** 279–288.](#)
- [Dellino G., Lino P., Meloni C., Rizzo A. \(2009\) Kriging metamodel management in the design optimization of a CNG injection system. *Mathematics and Computers in Simulation* **79** \(8\) 2345–2360.](#)
- [Dessouky, M., Lu, Q., Zhao, J., Leachman, R. \(2006\) An exact solution procedure to determine the optimal dispatching times for complex rail networks. *IIE Transactions* **38** \(2\) 141–152.](#)
- [Gorman, M.F. \(2009\) Statistical estimation of railroad congestion delay. *Transportation Research Part E* **45** \(3\) 446–456.](#)
- [Hansen I.A., Pachl, J. \(2014\) Railway timetabling & operations: analysis, modelling, optimisation, simulation, performance evaluation. Eurailpress, Hamburg, Germany.](#)
- [Higgins, A., Kozan, E., Ferreira, L. \(1996\) Optimal scheduling of trains on a single line track. *Transportation Research Part B* **30** \(2\) 147–158.](#)
- [Kanai, S., Shiina, K., Harada, S., Tomii, N. \(2011\) An optimal delay management algorithm from passengers viewpoints considering the whole railway network. *Journal of Rail Transport Planning & Management* **1** \(1\) 25–37.](#)
- [Lu C.-C. J., Wu Y.-C. J. \(2014\) Evaluation of Heuristics Using Data Envelopment Analysis. *International Journal of Information Technology and Decision Making* **13** \(4\) 795–810.](#)
- [Lu, C.-C., Yu, V.F. \(2012\) Data envelopment analysis for evaluating the efficiency of genetic algorithms on solving the vehicle routing problem with soft time windows. *Computers & Industrial Engineering* **63** \(2\) 520–529.](#)
- [Mascis, A., Pacciarelli, D. \(2002\) Job shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research* **143** \(3\) 498–517.](#)
- [Mazzarello, M., Ottaviani, E. \(2007\) A traffic management system for real-time traffic optimisation in railways. *Transportation Research Part B* **41** \(2\) 246–274.](#)
- [Meng, L., Zhou, X. \(2011\) Robust single-track train dispatching model under a dynamic and stochastic environment: A scenario-based rolling horizon solution approach. *Transportation Research Part B* **45** \(7\) 1080–1102.](#)
- [Pellegrini, P., Marlière, G., Rodriguez, J. \(2014\) Optimal train routing and scheduling for managing traffic perturbations in complex junctions. *Transportation Research Part B* **59** \(1\) 58–80.](#)
- [Sarkis, J. \(2007\) Preparing Your Data for DEA, In: Joe Zhu, Wade D. Cook \(Eds.\) *Modeling Data Irregularities and Structural Complexities in Data Envelopment Analysis*, 305–320.](#)
- [Sato, K., Tamura, K. Tomii, N. \(2013\). A MIP-based timetable rescheduling formulation and algorithm minimizing further inconvenience to passengers. *Journal of Rail Transport Planning Management* **3** \(3\) 38–53.](#)
- [Takeuchi, Y. Chikara, H. Tomii N. \(2006\) Robustness Indices based on passengers' utilities. *Proceedings of the 7th World Conference on Railway Research*, Montréal, Canada.](#)
- [Tomii, N., Tashiro, Y., Tanabe, N., Hirai, C., Muraki, K. \(2005\) Train rescheduling algorithm which minimizes passengers' dissatisfaction. In: Ali, M., Esposito, F. \(Eds.\), *Innovations in Applied Artificial Intelligence, Lecture Notes in Artificial Intelligence*, vol. 3533, pp. 829–838. Springer, Berlin, Germany.](#)
- [Törnquist, J., Persson, J.A. \(2007\) N-tracked railway traffic re-scheduling during disturbances. *Transportation Research Part B* **41** \(3\) 342–362.](#)