



Considering passenger and operator inconvenience in the scheduling of large railway projects

Kidd, Martin Philip; Lusby, Richard Martin ; Larsen, Jesper

Publication date:
2018

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Kidd, M. P., Lusby, R. M., & Larsen, J. (2018). *Considering passenger and operator inconvenience in the scheduling of large railway projects*.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Considering passenger and operator inconvenience in the scheduling of large railway projects

Martin Philip Kidd, Richard Martin Lusby & Jesper Larsen

Department of Management Engineering

Technical University of Denmark

June 22, 2017

Abstract

The continued development and renewal of railway infrastructure and technology is necessary to enable railway operators to provide high quality services subject to ever increasing demand. However, the execution of large infrastructure projects causes disturbances in the network due to the occupation of infrastructure over extended periods of time. In this paper we propose a multiobjective project scheduling optimization model for railway infrastructure projects that takes inconvenience caused to users of the infrastructure into account. We illustrate how the model can be used in an interactive way by planners based on their preferences, and we show that Pareto optimal solutions can be found in reasonable time using instances with realistic features. The result is a decision support model to aid infrastructure project planners in ensuring that passenger and operator inconvenience are also taken into account.

Keywords: Project scheduling, railway infrastructure, passenger perspective, multi-objective optimization, integer programming

1 Introduction

In order for railways to remain an attractive and energy efficient alternative for the transportation of passengers and freight, infrastructure managers have to ensure that enough capacity is provided through the railway infrastructure. Significant increases in demand over time often require significant changes to the current network, such as the replacement of outdated technology or the construction of new lines, which result in large construction and renewal projects to be managed by the infrastructure manager. The current projects for the Danish railways, for example, “reflect the vision to double the freight and passenger traffic by year 2030” (Banedanmark, 2017a), and include the upgrade of the signalling systems, the electrification of a large part of the Danish railway network, and the construction of new high speed railway lines to increase rail connectivity between Scandinavia and the rest of Europe. These are large, long-term projects with durations of 1–5 years and budgets of 1–2 billion Euros (Banedanmark, 2017b).

Even though the ultimate aim of railway infrastructure projects is to increase capacity, capacity is necessarily *decreased* during their execution, since the occupation of parts of

infrastructure by construction work renders it unusable to train traffic. Infrastructure works can therefore be highly disruptive to the normal traffic flow. Moreover, disturbances caused by elements such as noise, vibrations or dust can be an inconvenience to people living or working in areas adjacent to tracks where infrastructure work is being executed. Disturbances caused by the execution of railway infrastructure projects may therefore contribute to a negative reputation of the train operators, the infrastructure manager, and the mode of transport itself — exactly the opposite of the ultimate intention behind these projects. Thus, great care has to be taken by infrastructure managers to include inconvenience caused as a factor in the strategic planning of infrastructure projects.

In this paper we propose different measures of the suitability of a long-term strategic plan from the perspective of the infrastructure users (i.e. passengers and operators), and we include these measures as objectives in a mixed-integer programming model for the strategic planning of railway infrastructure projects. The measures are based on current practice at Banedanmark, as well as on similar measures that have been taken into account before in the (tactical or operational) planning of routine maintenance of railway infrastructure, where the problem is equally relevant. Moreover, as railway infrastructure projects have significant financial implications, we consider the smoothing of the cost profile over the long-term horizon as an additional objective. The problem we study can therefore be seen as a classical resource levelling problem (Rieck and Zimmermann, 2015) with additional constraints and objectives relating to the proposed measures.

As far as we are aware, an optimization model has not been proposed before for the scheduling of large railway infrastructure projects from a strategic point of view. When it comes to the tactical or operational scheduling of railway infrastructure maintenance, on the other hand, various optimization models can be found in the literature (see Lidén (2015) for an overview). In terms of taking inconvenience caused into account, the major difference between strategic scheduling and tactical or operational scheduling is that in the latter cases train timetables can usually be assumed to be available. The availability of timetables allows for a more realistic measure of inconvenience caused, and this has been taken into account in several papers from the literature. In some models (Higgins, 1998; Lake et al., 2002; Van Zante-De Fokkert et al., 2007) disturbances are avoided altogether by scheduling activities in “train-free slots” in the timetables, while other models (Higgins, 1998; Andrade and Teixeira, 2011; Albrecht et al., 2013) consider minimizing delays caused to train traffic as a result of interference by maintenance activities. Even though the timetable might not yet be available, it might be in the process of being constructed. In this case disturbances can be avoided or minimized by scheduling trains and maintenance activities simultaneously, as is considered in Albrecht et al. (2013), Forsgren et al. (2013) and Lidén and Joborn (2017).

On a strategic level a planning horizon of several years or decades can be considered, where we assume that timetables are not yet being constructed for periods so far in the future. For this reason we restrict our focus to measures that can be used independently of whether or not timetables are available. The contributions of this paper are summarized as follows:

1. We survey different ways in which inconvenience caused by maintenance works have been taken into account in scheduling models, and the ways in which it is taken into account at Banedanmark in strategic planning, and we group these into three general principles.
2. We show how these principles can be modelled by mixed integer programming (MIP) techniques and include them in a standard MIP formulation for classical resource

levelling.

3. Given that the problem contains multiple possibly conflicting objectives, we illustrate via an example how the MIP model can be used in an interactive approach in cases where the preferences of the planners are not clear.
4. Through a computational study on randomly generated instances with realistic features we show that the MIP model can be solved fast enough to facilitate an interactive approach.

2 Problem definition

In this paper we assume that a project portfolio has already been created, together with estimates of project durations and costs, and that strategic decisions need to be made as to when to execute these projects in the long term. The measures of inconvenience caused we consider in this paper is presented below as three formal *principles* to be followed in the scheduling of the infrastructure works in the project portfolio. Roughly stated, the three principles are aimed at maximizing potential traffic throughput, minimizing disruption durations, and maximizing durations of breaks inbetween disruptions, respectively.

Principle 1 *Maximize potential flow through a critical set of infrastructure resources by minimizing the number of resources occupied simultaneously at any point in time.*

A critical set could, for example, be a railway corridor that represents a crucial link in a network. Given a partition of such a corridor into sections of track, this principle states that bottlenecks or capacity reductions (such as train speed restrictions) should be imposed only on very few sections simultaneously along the corridor. [Nemani et al. \(2010\)](#), for example, consider *service corridors*, each of which is associated with a number of *subdivisions*. In their scheduling model a constraint is imposed that at most one subdivision of each service corridor may be occupied by maintenance work during any week, in order to ensure that the movement of high volumes of freight through the corridor is not too severely affected. Similar constraints related to subdivision of the infrastructure are also considered in a number of other papers, see for example [Boğ et al. \(2010\)](#), [Peng et al. \(2011\)](#), [Peng and Ouyang \(2012\)](#) and [Peng and Ouyang \(2014\)](#). A critical set could also represent a set of parallel track sections at a junction or in a railway yard, where the occupation of fewer sections simultaneously allows more possibilities for the rerouting of trains ([Peng et al., 2011](#); [Jenema, 2011](#)). There is also some similarity to the work of [Boland et al. \(2013, 2014\)](#), where maintenance activities are scheduled in such a way that the total flow of trains through the network is maximized, given that some links are occupied by maintenance tasks.

At Banedanmark the principle is imposed on high traffic regional lines that are subdivided into line sections, and where any two infrastructure projects on the same regional line, but on different line sections, should preferably not be scheduled in the same year.

Principle 2 *Minimize the duration (or cost) of an occupation of an infrastructure resource by scheduling activities simultaneously.*

This principle relates to the occupation of a single resource, e.g. a track section. Scheduling activities simultaneously will allow for fewer time periods during which capacity is restricted by infrastructure activities and therefore fewer disruptions to train operations. The assumption here is that the severity of the disruption caused, such as the

closure of a track section, depends significantly more on the duration of the occupation of the resource than on the amount of work done during the occupation. This principle is not uncommon in practice. For example, one of the principles outlined by RailNet-Europe (RNE) for planned temporary capacity restrictions on the European Rail Freight Corridors (RFCs) is indeed that “[i]n case of total closure the aim should be to plan the maximum amount of works simultaneously if technically possible” (RailNetEurope, 2015). At Banedanmark, a set of *coordination principles* are imposed that aim to synchronize infrastructure projects that are to be executed on the same route section in the network.

Lidén (2015) groups previous work that considers this principle under *possession and work coordination*. In Budai et al. (2006), for example, the durations of possessions are minimized (among other objectives) by synchronizing maintenance and renewal activities, and similarly Caetano and Teixeira (2013) minimizes the track unavailability during the maintenance period. More generally speaking, costs on the duration of the resource occupation can also be imposed. Jenema (2011), for example, introduces a time-dependent cost on the duration of so-called “Train-Free-Periods,” and possession costs are taken into account by Zhao et al. (2009) and Pouryousef et al. (2010). As discussed by Zhao et al. (2009), possession costs are sometimes also meant as compensation to train operators for disrupting their services. Finally, a somewhat different approach is taken by Peng et al. (2011) and Peng and Ouyang (2012), who consider the option of “splitting” projects, i.e. assigning twice the amount of resources (maintenance teams in their case) to halve the infrastructure occupation time.

Principle 3 *Maximize the durations of breaks between consecutive non-overlapping occupations of an infrastructure resource.*

Due to conflicts among activities, it will not always be possible to schedule all activities that occupy a specific resource simultaneously. Often more than one occupation of a resource is necessarily required. A difficulty that might arise is that, even though activities are synchronized in such a way that, for example, two non-overlapping occupations have minimal durations, it might still be perceived by users of the system to be a single occupation if they are scheduled close enough in time. This motivates the inclusion of a break inbetween two occupations that is long enough to avoid them being perceived as one occupation. A principle imposed by Banedanmark in the strategic planning of infrastructure projects is therefore to avoid returning to a part of infrastructure too soon (i.e. within 5–10 years) after a significant amount of work has been completed there.

3 Mixed integer programming formulation

In this section we present a formulation of a mixed integer program (MIP) with four objectives, one for each principle introduced in the previous section, as well as a resource levelling objective. Rieck et al. (2012) compares two formulations for classical resource levelling, and as the basis of the MIP proposed in this paper we use the one found to be dominating in their computational studies. Moreover, we will use *scalarization techniques* (Ehrgott, 2006) (also known as *criterion space search* (Boland et al., 2015)), which are methods that transform the problem into a single objective optimization problem which is then solved by a black-box MIP solver. Different settings of the parameters of the scalarization typically result in different solutions on the Pareto front.

Determining a single scalarization to be used to generate schedules is likely not a practically feasible approach, as it is difficult to make any assumptions about the preferences

of the planners a priori. It would therefore be more appropriate to use *interactive methods* (Miettinen et al., 2016), where the parameters of the scalarization are adjusted by the planners each time after reviewing a generated schedule, until a satisfactory schedule is produced. Preference modelling and the implementation of interactive methods are outside the scope of this paper, but we do illustrate the use of scalarizations of our model using a small example. For this example we make some preference assumptions and we develop a schedule step by step as the different objectives are introduced. The example will be introduced in §3.1.

3.1 Project scheduling constraints

We assume that a set of activities A (e.g. infrastructure projects) has to be scheduled. The scheduling horizon comprises a set of time periods T , where $|T|$ gives the length of the horizon. We define a *schedule* as a vector S that contains an integer variable S_a for each activity $a \in A$, where S_a represents the start time period to be assigned to a . Each activity $a \in A$ is further associated with a fixed processing time of a certain number of periods p_a , and the feasible time periods in which activity a can start is given by $T_a \subseteq T$. In what follows we define $[S]_a = [S_a, S_a + p_a - 1]$, and an activity a is said to be *active* during time period t in schedule S if $t \in [S]_a$. We introduce a number of *big-M* constraints in this section, and towards this end we define $M_{ab} = \max T_a - \min T_b$, and we note that $-M_{ba} \leq S_a - S_b \leq M_{ab}$ is satisfied by any feasible schedule.

Among some activities coordination may be required, e.g. because they form part of a larger common project, and we denote by $P \subseteq A \times A$ the set of all pairs of activities that need to be coordinated. By that we mean that $\underline{\delta}_{ab} \leq S_a - S_b \leq \bar{\delta}_{ab}$ should be satisfied for some predefined parameters $\underline{\delta}_{ab}, \bar{\delta}_{ab}$ for all $(a, b) \in P$. Such constraints are known as *generalized precedence constraints* in project scheduling (Elmaghraby and Kamburowski, 1992). On the other hand, pairs of activities may be in conflict due to technical aspects of their execution, and we denote by $Q \subseteq A \times A$ the set of all activities that are in conflict. In this case the constraint $[S]_a \cap [S]_b = \emptyset$ should be satisfied for all $(a, b) \in Q$.

Constraints (1) and (2) ensure that activities are coordinated while Constraints (3) and (4) ensure, by means of a binary variable ξ_{ab} , that conflicts between activities are avoided.

$$S_a - S_b \leq \bar{\delta}_{ab} \quad (a, b) \in P \quad (1)$$

$$S_a - S_b \geq \underline{\delta}_{ab} \quad (a, b) \in P \quad (2)$$

$$S_a - S_b \geq p_b - (M_{ba} + p_b)\xi_{ab} \quad (a, b) \in Q \quad (3)$$

$$S_b - S_a \geq p_a - (M_{ab} + p_a)(1 - \xi_{ab}) \quad (a, b) \in Q \quad (4)$$

$$\xi_{ab} \in \{0, 1\} \quad (a, b) \in Q \quad (5)$$

The activities also share a set of resources R (e.g. manpower, equipment, budgets), where each resource $k \in R$ has a limited availability u_k . Each activity a consumes an amount r_{ak} of resource k during each time period a is active, and the total consumption of resource k during time period t is given by

$$\rho_k(S, t) = \sum_{a \in A: t \in [S]_a} r_{ak}$$

for schedule S . The step function $\rho_k(S, \cdot)$ is known as the *resource profile* of resource k for schedule S . In general two different approaches to dealing with shared resources can be

considered. The first and most common is *resource constrained project scheduling*, where a constraint $\rho_k(S, t) \leq u_k$ is imposed for all $k \in R$ and $t \in T$. The second approach is *time constrained project scheduling*, where $u_k = \infty$ is assumed and instead a function related to the resource profile is to be minimized. In *resource levelling*, the classical objective function (to be minimized) is

$$\sum_{k \in R} \lambda_k \sum_{t \in T} \rho_k(S, t)^2,$$

which is equivalent to minimizing a weighted sum of the variances of the $|R|$ resource profiles. This could model, for example, a requirement for a smooth cost profile (as considered in this paper) or for a balanced workload over the scheduling horizon (as considered in [Van Zante-De Fokkert et al. \(2007\)](#), for example).

In this paper we consider resource levelling of a single resource k , and we therefore simplify notation by defining $r_a = r_{ak}$. Let x_a^t be a binary variable that indicates whether or not activity $a \in A$ starts in time period $t \in T_a$, and let r_t be a continuous variable that denotes the total consumption of the resource in time period $t \in T$. The total resource consumption of the resource during each time period is calculated as a function of the schedule vector S by Constraints (6)–(10). Note that Constraint (7) also ensures that each activity is only assigned one of its feasible start time periods.

$$\sum_{t \in T_a} t x_a^t = S_a \quad a \in A \quad (6)$$

$$\sum_{t \in T_a} x_a^t = 1 \quad a \in A \quad (7)$$

$$\sum_{a \in A} r_a \sum_{\substack{t' \in T_a; \\ t-p_a+1 \leq t' \leq t}} x_a^{t'} \leq r_t \quad t \in T \quad (8)$$

$$x_a^t \in \{0, 1\} \quad a \in A, t \in T_a \quad (9)$$

$$r_t \geq 0 \quad t \in T \quad (10)$$

The objective is to minimize

$$z_{\text{RL}} = \sum_{t \in T} r_t^2, \quad (11)$$

which can be modelled by linear constraints as described in [Rieck et al. \(2012\)](#).

As promised, we illustrate the use of our model by means of a small example. The example is shown in [Figure 1](#), and it consists of 17 activities of durations between 1 and 2 periods, 3 distinct parts of infrastructure (sections), and a planning horizon of 15 periods. The light-gray activities are to be coordinated in that they all have to start and end within 6 consecutive time periods, and conflicts exist between light-gray and dark-gray activities if they are on the same section. A single resource is considered, where the consumption of the resource by each activity is given by the number given inside the block representing the activity. The schedule shown in [Figure 1](#) minimizes (11) subject to (1)–(9); in particular, in each period 8 units of the resource are consumed, except for periods 0, 10 and 14, during which 7 units of the resource are consumed.

In the next section we discuss the first principle. Recalling this principle, and assuming that the three sections in the example either form a busy corridor or parallel lines in a junction or railway yard, we can see that the schedule shown in [Figure 1](#) scores badly on this principle. This can be concluded since in all but periods 0, 8 and 14 two out of the three sections are occupied simultaneously, while in period 12 all three sections

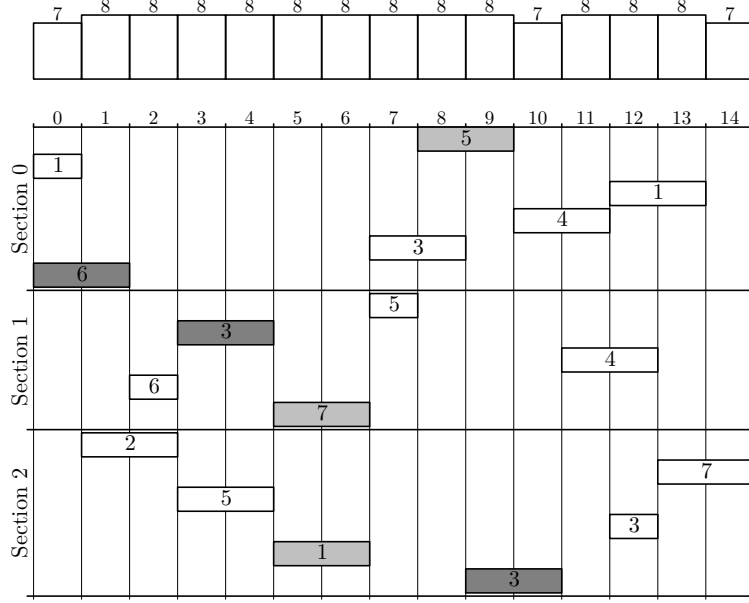


Figure 1: Example consisting of 17 activities, 3 sections and 15 time periods. Light-gray activities all have to start and end within 6 consecutive time periods, while light-gray and dark-gray activities on the same section may not overlap. The resource consumption of each activity is given inside the block representing the activity, and the resource profile is shown above the schedule.

are occupied simultaneously. In the next section we incorporate the first principle in the model in order to remedy this situation.

3.2 Principle 1

We assume that the activities are associated with a set of *infrastructure resources* I (e.g. track sections), where each activity $a \in A$ occupies an infrastructure resource $i_a \in I$ during its execution. For any infrastructure resource $i \in I$, let $A_i = \{a \in A : i = i_a\}$ denote the set of activities associated with i , and let $\mathcal{A}_i(S, t) = \{a \in A_i : t \in [S]_a\}$ denote the set of activities associated with i that are active during time period t in schedule S . We say that an infrastructure resource $i \in I$ is *occupied* during time period t in schedule S if $\mathcal{A}_i(S, t) \neq \emptyset$; otherwise we say that the infrastructure resource is *free* during that period. Let y_i^t be a binary variable that indicates whether or not infrastructure resource i is occupied during time period t . Constraints (12)–(13) link these variables to the existing model.

$$\sum_{\substack{t' \in T_a: \\ t-p_a+1 \leq t' \leq t}} x_a^{t'} \leq y_i^t \quad i \in I, a \in A_i, t \in T \quad (12)$$

$$y_i^t \in \{0, 1\} \quad i \in I, t \in T \quad (13)$$

For the first principle we define a collection of *critical sets* \mathcal{C} (e.g. corridors, junctions), where each critical set $C \in \mathcal{C}$ is a subset of infrastructure resources. We introduce an integer variable σ_C^t denoting the number of infrastructure resources in $C \in \mathcal{C}$ in addition to one that are occupied during time period t , which is calculated by Constraint (14).

$$\sum_{i \in C} y_i^t \leq \sigma_C^t + 1 \quad C \in \mathcal{C}, t \in T \quad (14)$$

The objective in this case is to keep σ_C^t small for all values of t and C , and so we minimize

$$z_{P1} = \sum_{C \in \mathcal{C}} \lambda_C \sum_{t \in T} (\sigma_C^t)^2, \quad (15)$$

where $\lambda_C \geq 0$ is an input parameter that represents the relative importance of critical set C . We use the same linearization technique as for (11).

At this point we have two objectives, namely to minimize z_{RL} and z_{P1} , respectively. Minimizing z_{P1} subject to (1)–(15) for our example gives a solution with an optimal objective function value of 0 — that is, no two sections are ever occupied simultaneously in this schedule. Figure 2 (left) shows a schedule that minimizes z_{RL} subject to (1)–(15) and the additional constraint $z_{P1} = 0$. In other words, among all solutions that are optimal with respect to the first principle, this schedule optimizes the resource profile. This is known as *lexicographical optimization*, and imposes a strict priority of one objective over another.

More generally one can minimize z_{RL} subject to (1)–(15) and the additional constraint $z_{P1} \leq \epsilon$ for some $\epsilon \geq 0$, which is known as an ϵ -*constraint method*. By solving the MIP four times for $\epsilon = 1, 2, 3, 4$ (note that for this example z_{P1} will necessarily be integer) we find that $\epsilon = 4$ is the smallest value such that the optimal resource profile is attained, as shown in Figure 2 (right). This is therefore also a schedule that is an optimal solution to the lexicographical optimization problem where resource levelling has strict priority over the first principle. In-between the two solutions shown in Figure 2, that is for $\epsilon = 1, 2, 3$, alternative *nondominated* schedules can be generated with different tradeoffs between the two objectives. The final decision of which of these schedules to accept depends on the preferences of the planners.

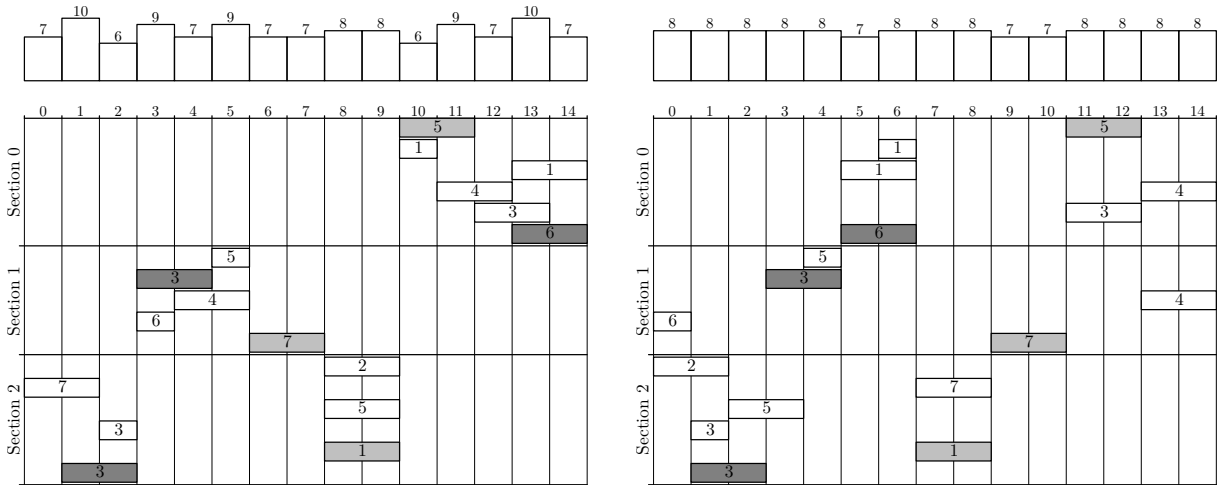


Figure 2: Two lexicographically optimal schedules, where in one case (left) Principle 1 has higher priority than resource levelling, while in the other (right) the reverse is true.

3.3 Principles 2 & 3

An *occupation window* on a section is defined as a maximal set of consecutive time periods during which the section is occupied, while a *break* contains all time periods between one occupation and the next. For example, in the schedule shown in Figure 2 (left) there are two occupation windows on Section 2, one with a duration of three time periods and one

with a duration of two time periods. Moreover, there is one break starting in time period 3 and ending in time period 7.

In order to enforce the second principle, we define Θ as the duration of the longest occupation window in the schedule and define the second objective to be to minimize

$$z_{P2} = \Theta. \quad (16)$$

For the third principle, we define Δ as the shortest break in the schedule and define the third objective to be to maximize

$$z_{P3} = \Delta. \quad (17)$$

Before showing how these objectives can be linked to the scheduling variables, we first consider the example. Assume that the planners agree that a value $z_{P1} \leq 2$ for the first principle is acceptable, i.e. during at most two time periods across the scheduling horizon may at most two out of three sections be occupied. Minimizing z_{P2} we find a schedule where no occupation window is longer than two time periods, which attains the trivial dual bound $\max_{a \in A} p_a = 2$. Figure 3 shows a schedule that minimizes z_{RL} subject to the additional ϵ -constraints $z_{P1} \leq 2$ and $z_{P2} \leq 2$.

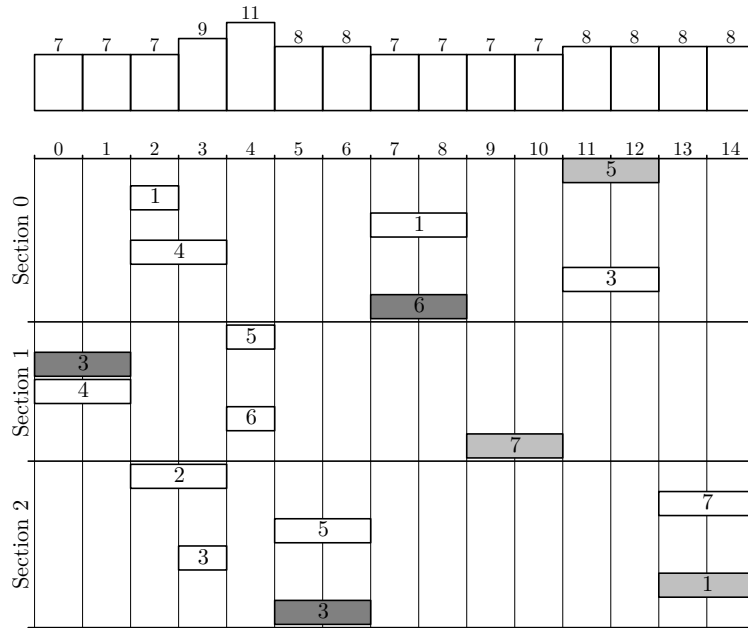


Figure 3: A schedule that minimizes z_{RL} subject to $z_{P1} \leq 2$ and $z_{P2} \leq 2$.

Note that even though the occupation windows are as short as they can be for this example, some of them are fairly close to each other in time. For example, the first and second occupation windows of Section 2 are only one time period apart. We therefore next maximize z_{P3} , and find $z_{P3} = 8$ to be optimal. Figure 4 (left) shows a schedule that minimizes z_{RL} subject to the additional ϵ -constraints $z_{P1} \leq 2$, $z_{P2} \leq 2$ and $z_{P3} \geq 8$. As can be seen this results in a resource profile that is not very well balanced. Assume that the planners agree to relax $z_{P3} \geq 8$ to $z_{P3} \geq 3$, i.e. that breaks of duration 3 would be acceptable, but not less. Figure 4 (right) shows the resulting schedule after this relaxation, which has a more balanced resource profile. Moreover, this schedule now takes into account all objectives at acceptable levels to the planners, and can therefore be seen as a candidate solution.

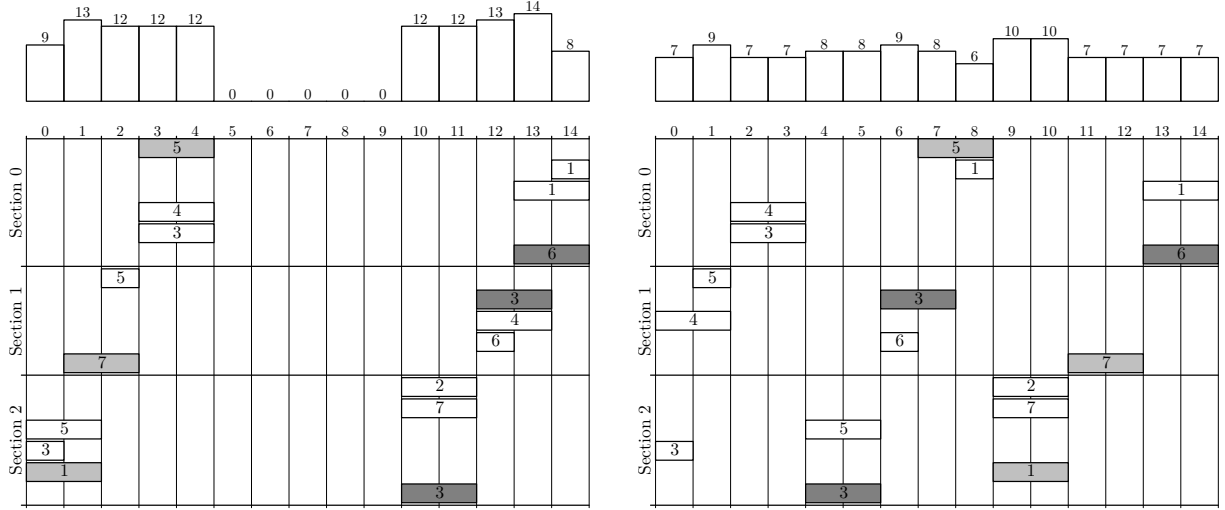


Figure 4: A schedule (left) that minimizes z_{RL} subject to $z_{\text{P}1} \leq 2$, $z_{\text{P}2} \leq 2$ and $z_{\text{P}3} \geq 8$, and a schedule (right) that minimizes z_{RL} subject to $z_{\text{P}1} \leq 2$, $z_{\text{P}2} \leq 2$ and $z_{\text{P}3} \geq 3$.

Note that this example is only for illustrative purposes, and that in general such an interactive approach will be case dependent, and in particular it will depend on the preferences of the planners. The MIP model introduced above, together with the many possibilities that scalarization offers (note that theoretically *any* nondominated schedule can be generated by some scalarization (Ehrgott, 2006)), results in an optimization framework that is able to facilitate such an interactive approach. An important question that remains unanswered is the computational performance on the MIP model introduced, as an interactive approach is most effective when the computation times are not too large. This question is answered in §4.

It remains to be discussed how the variables Θ and Δ can be linked to the rest of the model. In what follows we present two different ways of linking Θ to the scheduling variables, which we respectively refer to as the *discrete model* and the *disjunctive model*. In the discrete model the variables Θ and Δ are defined in terms of the binary time variables x_a^t and $O(|T|^2)$ additional binary variables are introduced for each infrastructure resource. In the disjunctive model, on the other hand, the variables Θ and Δ are defined in terms of the integer time variables S_a (using big- M constraints) and $O(|A_i|^2)$ additional binary variables are introduced for each infrastructure resource $i \in I$.

3.3.1 Discrete model

We introduce a binary variable $\alpha_{tt'}^i$ that indicates whether or not the time interval $[t, t']$ lies inside an occupation window on resource i , or equivalently whether or not infrastructure resource i is occupied during each period in $[t, t']$. We will denote the length of the interval $[t, t']$ by $d_{tt'} = t' - t + 1$. Constraints (18) and (19) use this variable to link the binary time variables (through the binary occupation variables) with the duration of the longest occupation window.

$$\sum_{s=t}^{t'} y_i^s \leq d_{tt'} - 1 + \alpha_{tt'}^i \quad i \in I, t, t' \in T, t \leq t' \quad (18)$$

$$\alpha_{tt'}^i d_{tt'} \leq \Theta \quad i \in I, t, t' \in T, t \leq t' \quad (19)$$

Constraint (18) ensures that if the infrastructure resource is occupied during each period in $[t, t']$, then $\alpha_{tt'}^i$ is forced to assume a value of 1, and (19) ensures that Θ is equal to the length of the longest continuous occupation of the infrastructure resource.

Moreover, we introduce a binary variable $\beta_{tt'}^i$ that indicates whether or not the interval $[t, t']$ corresponds to a break on resource i . In other words, $\beta_{tt'}^i = 1$ if resource i is free during all time periods in $[t, t']$ but occupied during time periods $t - 1$ and $t' + 1$ (where we assume that $y_i^{-1} = 0 = y_i^{|T|}$). Constraints (20) and (21) use this variable to link the binary time variables with the duration of the shortest break.

$$y_i^{t-1} + y_i^{t'+1} - \sum_{s=t}^{t'} y_i^s \leq 1 + \beta_{tt'}^i \quad i \in I, t, t' \in T, t \leq t' \quad (20)$$

$$|T|(1 - \beta_{tt'}^i) + \beta_{tt'}^i d_{tt'} \geq \Delta \quad i \in I, t, t' \in T, t \leq t' \quad (21)$$

Note that the intervals $[\min T, t]$ and $[t, \max T]$ cannot be breaks for any $t \in T$ according to this definition. Moreover, if there are no breaks on the section (i.e. if there is only one occupation window) then, since it is technically undefined, Δ is set equal to the trivial upper bound $|T|$.

3.3.2 Disjunctive model

For each infrastructure resource $i \in I$ we introduce a binary variable ψ_{ab} that indicates whether or not two activities $a, b \in A_i$, are in the same occupation window, and a binary variable ϕ_{ab} that indicates the order in which the two activities are scheduled. In order to determine whether or not two activities are in the same occupation window it is useful to consider, for each infrastructure resource i and a schedule S , a graph $G_i(S)$ with vertex set A_i for which an edge (a, b) exists if either $S_a \leq S_b \leq S_a + p_a$ or $S_b \leq S_a \leq S_b + p_b$. Each connected component of this graph therefore corresponds to an occupation window, and two activities are in the same occupation window if there exists a path between the two corresponding vertices in the graph. Equivalently, two activities are in the same occupation window if they are adjacent in the transitive closure of $G_i(S)$.

The scheduled order of two activities is determined by Constraints (22) and (23), and whether or not they are adjacent in $G_i(S)$ or connected by a path through transitivity is determined by Constraints (24)–(26). Note that no specific order between the elements a and b is assumed, and therefore we add the substitutions $\psi_{ab} = \psi_{ba}$ and $\phi_{ab} = \phi_{ba}$ and assume all duplicate constraints are removed from the model.

$$S_a - S_b \leq M_{ab}(1 - \phi_{ab}) \quad i \in I, a, b \in A_i \quad (22)$$

$$S_b - S_a \leq M_{ba}\phi_{ab} \quad i \in I, a, b \in A_i \quad (23)$$

$$S_b - S_a \geq (p_a + 1)(1 - \psi_{ab}) - (M_{ab} + p_a + 1)(1 - \phi_{ab}) \quad i \in I, a, b \in A_i \quad (24)$$

$$S_a - S_b \geq (p_b + 1)(1 - \psi_{ab}) - (M_{ba} + p_b + 1)\phi_{ab} \quad i \in I, a, b \in A_i \quad (25)$$

$$\psi_{ac} + \psi_{bc} \leq 1 + \psi_{ab} \quad i \in I, a, b, c \in A_i \quad (26)$$

Note that if a and b are scheduled such that $S_a < S_b$, then $\phi_{ab} = 1$ and $S_b - S_a \geq (p_a + 1)(1 - \psi_{ab})$ is imposed by (24). This forces $\psi_{ab} = 1$ if $S_b \leq S_a + p_a$ holds, and otherwise imposes no constraint on ψ_{ab} (in order to allow it to still attain 1 through transitivity). The same idea holds for (25), whereas (26) imposes transitivity.

In order to impose the value of Θ with respect to a schedule, we compare activities pairwise in order to check whether they are in the same occupation window. If they are,

the timespan from the start of the earlier one to the finish of the later one must be at most the duration of the longest occupation window. This is determined by Constraints (27) and (28). We also have to account for the fact that an occupation window can span a single activity. For this it is sufficient that the duration of the longest occupation window must be at least the duration of the longest activity, as imposed by (29).

$$S_a + p_a - S_b \leq \Theta + (M_{ab} + p_a)(1 - \psi_{ab}) \quad i \in I, a, b \in A_i \quad (27)$$

$$S_b + p_b - S_a \leq \Theta + (M_{ba} + p_b)(1 - \psi_{ab}) \quad i \in I, a, b \in A_i \quad (28)$$

$$\max_{a \in A_i} p_a \leq \Theta \quad i \in I \quad (29)$$

In order to impose the value of Δ with respect to a schedule, we again compare activities pairwise in order to check whether they are in the same occupation window. If they are not, the timespan from the end of the earlier one to the start of the later one must be at least the length of the shortest break. This is determined by Constraints (30) and (31). We also have to account for the fact that a section may have only one occupation window and no breaks, in which case Δ is equal to the upper bound $|T|$ as imposed by (32).

$$S_b - (S_a + p_a) \geq \Delta - (|T| + M_{ab} + p_a)(1 - \phi_{ab} + \psi_{ab}) \quad i \in I, a, b \in A_i \quad (30)$$

$$S_a - (S_b + p_b) \geq \Delta - (|T| + M_{ba} + p_b)(\phi_{ab} + \psi_{ab}) \quad i \in I, a, b \in A_i \quad (31)$$

$$|T| \geq \Delta \quad i \in I \quad (32)$$

Note that a further implication of Constraints (24)–(26) is that no restriction is imposed on ψ_{ab} if a and b are not in the same occupation window in some schedule. It is thus feasible to have $\psi_{ab} = 1$ even though a and b are not in the same occupation window. This is not a problem in case (16) is used as the objective function, since there is an indirect cost associated with setting $\psi_{ab} = 1$ in this case. If (17) is used as an objective, on the other hand, then having $\psi_{ab} = 1$ even though a and b are not in the same occupation window avoids the break in-between the occupation windows of a and b , respectively.

This is, however, a very rare occurrence in our case, which is due to the fact that when maximizing the minimum break duration we follow a lexicographical approach where a tight upper bound on the durations of occupation windows is imposed at the same time. Since such an infeasibility only rarely occurs, we check feasibility in a postprocessing phase. If an infeasibility is detected, a feasibility cut is included in the model and it is solved again.

Assume that in a solution it is found that $\psi_{ab} = 1$ while a and b are not in the same occupation window. Assume further w.l.o.g. that $S_b - (S_a + p_a) = \bar{d}$ in this solution. To avoid this infeasibility, and infeasibilities similar to it, we add the Constraints (33).

$$\psi_{ab} + x_a^t + x_b^{t'} \leq 2 + \sum_{s=\min(t+p_a, t'+p_b)+1}^{\max(t, t')-1} y_i^s \quad (t, t') \in T_a \times T_b : \max(t, t') - \min(t + p_a, t' + p_b) = \bar{d} \quad (33)$$

Note that this is in total $2(|T| - p_a - p_b - \bar{d})$ additional constraints. We found in our computational study that of the 700 instances we considered, for only 3 of them such feasibility cuts were needed. In these three cases one, two and four iterations were needed, respectively, to resolve the infeasibilities.

3.4 Possible extensions of the model

We conclude this section with a short note on the flexibility of the modelling approaches with respect to possible problem extensions.

Note that the way in which the discrete model was formulated for Principle 3 can also be used in the formulation of Principle 2. In this case we enforce that $\alpha_{tt'}^i$ indicates whether or not $[t, t']$ is exactly an occupation window, instead of merely a subset of one as previously defined. In other words, $\alpha_{tt'}^i = 1$ if resource i is occupied during all time periods in $[t, t']$ but free during time periods $t - 1$ and $t' + 1$ (where we again assume that $y_i^{-1} = 0 = y_i^{|T|}$). This can be ensured by using Constraint (34) instead of Constraint (18).

$$\sum_{s=t}^{t'} y_i^s - y_i^{t-1} - y_i^{t'+1} \leq d_{tt'} - 1 + \alpha_{tt'}^i \quad i \in I, t, t' \in T, t \leq t' \quad (34)$$

However, this results in additional symmetry, since if $[t, t']$ is not an occupation window the variable $\alpha_{tt'}^i$ is free to take either value, whereas if $[t, t']$ is a subset of an occupation window $\alpha_{tt'}^i$ is forced take a value of 1 when using (18). Moreover, this is a weaker formulation as (18) implies (34).

Nonetheless, modelling occupation windows in this way allows us to model an arbitrary function $f_i(t, d)$ that maps to a cost of occupying resource i for d time periods starting in time period t . In this case the objective is to minimize

$$\sum_{i \in I} \lambda_i \sum_{t, t' \in T, t \leq t'} f_i(t, d_{tt'}) \alpha_{tt'}^i.$$

This function may assume any form, as long as its value can be calculated for all i , t and d . For example, it could represent compensation paid to operators, or a quantitative measure (found through simulation for example) of the severity of the disruption to train traffic given that a part of infrastructure is occupied for a certain period in time.

The same can also be done for Principle 3, where a function $g_i(t, d)$ can be used to model the incentive of allowing a break in infrastructure works for d time periods on a specific resource i starting in time period t . In this case the objective is to maximize

$$\sum_{i \in I} \lambda_i \sum_{t, t' \in T, t \leq t'} g_i(t, d_{tt'}) \beta_{tt'}^i.$$

4 Computational study

In our computational study we address two main questions. Firstly, what can be expected in terms of required computation time when solving realistic instances of the problem to (Pareto) optimality using the models proposed in this paper? Secondly, how do the performances of the discrete and disjunctive models compare, and when is it appropriate to use which one? We consider seven instance classes, each of which consists of 100 randomly generated instances available online at <http://www.ms.man.dtu.dk/research/instances>. The first class, **Base**, is used to answer the first question, while the remaining six classes is used to answer the second question. In the following three sections we discuss the features of these classes, the scalarizations of the models used, and the computational results, respectively.

$ T $	$ A $	$ I $	$ P $	$ Q $	p_a	$\forall C \in \mathcal{C}$ $ C $	$\forall a \in A$ r_a	$\forall (a, b) \in P$ $\underline{\delta}_{ab}$	$\bar{\delta}_{ab}$
20	50	6	25	12	$[1, 3]$	$[2, 4]$	$[1, 10]$	$-p_a$	p_b

Table 1: Features of the class **Base**.

4.1 Instance features and classes

The features of the class **Base** are given in Table 1. Except for the resource consumption, all features are based on what we observed in the current project portfolio of Banedanmark when looking at the problem from a regional point of view. Since we consider planning at a strategic level, the railway infrastructure, the time dimension and the project portfolio is defined at a highly aggregated level of detail. For the Banedanmark case, for example, the railway infrastructure is partitioned into *line sections* of length roughly between 10 and 100 kilometres. This results in only a limited number of line sections within certain designated regions of the country. Project durations can be aggregated into integer multiples of semesters or years, and the planning horizon can be between 10 and 20 years. Since projects are for the most part independent, the number of coordinated and conflicting activities is relatively small. In all instances each activity is coordinated with one other activity on average, while on average half of the activities are in conflict with one other activity.

Critical sets, which in the Banedanmark case correspond to regional corridors, contain between two and four infrastructure resources. To generate critical sets, we sequentially sample integers uniformly from the range $[2, 4]$ until the cumulative sum of these integers exceeds $|I|$, at which point the last number generated is discarded. The infrastructure resources are then randomly inserted into these positions in the critical sets. For the resource consumption of each activity $a \in A$ we generate r_a uniformly in the standard integer range $r_a \in [1, 10] \cap \mathbb{Z}$ that is used for instances in the project scheduling problem library PSPLIB (Kolisch and Sprecher, 1997). For coordinated activities, generalized precedence constraints are generated in such a way that the two activities are forced to overlap. Finally, we consider complete flexibility within the horizon as to when projects can be scheduled, i.e. $T_a = \{t \in T : t + p_a - 1 \in T\}$.

We define the density of an instance to be $\frac{|A|-2|I|}{|A|-2}$, which assumes a value of 0 if there is only one activity associated with each infrastructure resource and a value of 1 if all activities are associated with a single infrastructure resource. Recall that the number of additional variables and constraints per infrastructure resource $i \in I$ for modelling Principles 2 & 3 is $O(|T|^2)$ for the discrete model and $O(|A_i|^2)$ for the disjunctive model. The expectation is therefore that the discrete model outperforms the disjunctive model on instances with high density and few time periods, while the reverse is true for instances with low density and many time periods. In order to explore this hypothesis, a distinction is made in the remaining instance classes between sparse and dense instances, and instances with a planning horizon of either 10, 15 or 20 time periods. The features of these classes that are different from those of the class **Base** are given in Table 2.

4.2 Scalarizations

We consider two scalarization techniques applied to our model, one based on lexicographical optimization and one using ϵ -constraints, and we denote them by **LEX** and **EPS**($z_{P1}, z_{P2}, \bar{z}_{P3}$), respectively. Both scalarizations eventually minimizes z_{RL} subject to

Class	$ T $	$ A $	$ I $	$ P $	$ Q $	p_a
Sparse10	10	20	7	10	5	[1, 2]
Sparse15	15	20	7	10	5	[1, 2]
Sparse20	20	20	7	10	5	[1, 2]
Dense10	10	20	4	10	5	[1, 2]
Dense15	15	20	4	10	5	[1, 2]
Dense20	20	20	4	10	5	[1, 2]

Table 2: Features of the instance classes used to compare the discrete model with the disjunctive model. The remaining instances features not shown here have the same values as in Table 1.

$z_{P1} \leq \underline{z}_{P1}$, $z_{P2} \leq \underline{z}_{P2}$, $z_{P3} \geq \bar{z}_{P3}$, and all other necessary constraints (depending on whether the discrete or disjunctive model is used). However, the scalarizations differ on how the bounds for the principles are defined. In the case of **LEX**, the bounds are defined by solving the three scalarizations

$$\begin{aligned} \underline{z}_{P1} &= \min z_{P1}, \\ \underline{z}_{P2} &= \min z_{P2} : z_{P1} \leq \underline{z}_{P1}, \text{ and} \\ \bar{z}_{P3} &= \max z_{P3} : z_{P1} \leq \underline{z}_{P1}, z_{P2} \leq \underline{z}_{P2}, \end{aligned}$$

whereas in the case of **EPS**($\underline{z}_{P1}, \underline{z}_{P2}, \bar{z}_{P3}$) they are predefined.

The case of **LEX** corresponds to a case where the planners might initially be uncertain as to what bounds for the principles are realistic. The specific lexicographical ordering was chosen based on the following four observations obtained through initial experimentation:

- Optimizing z_{P3} with no upper bound on z_{P2} typically results in single occupations with long durations in order to avoid the necessity of having any breaks at all, leaving no flexibility for reducing the value of z_{P2} .
- Optimizing z_{P2} and z_{P3} (in that order) with no upper bound on z_{P1} tends to cluster activities at the start and end of the planning horizon, leaving no flexibility for reducing the value of z_{P1} .
- Even if $z_{P1} = 0$ (its best case), it was found to not be exceedingly prohibitive to finding good solutions for z_{P2} and z_{P3} (see Figure 4 (right), for example).
- Even if $z_{P2} = \max_{a \in A} p_a$ (its best case), it was found to not be exceedingly prohibitive to finding good solutions for z_{P3} .

Finally, z_{RL} is considered to be the least important objective, as the focus of this paper is on the three principles.

The case of **EPS**($\underline{z}_{P1}, \underline{z}_{P2}, \bar{z}_{P3}$), on the other hand, corresponds to a case where the planners have an idea of what are realistic bounds for the principles and would like to refine the solution by specifying specific bounds. In our study we consider **EPS**(0, 3, 7) for the class **Base**, as this corresponds to current practice at Banedanmark. In particular, the first two principles are imposed as efficiently as possible. For the remaining classes we use **EPS**(0, 2, 3), **EPS**(0, 2, 5) and **EPS**(0, 2, 7). Once again the first two principles are imposed as efficiently as possible, while we study a wider range of minimum break lengths for these instances.

4.3 Results

Scalarizations of the models were solved using CPLEX 12.6 with default settings on a computer with two Intel[®] Xeon[®] X5550 processors. In order to compare the perfor-

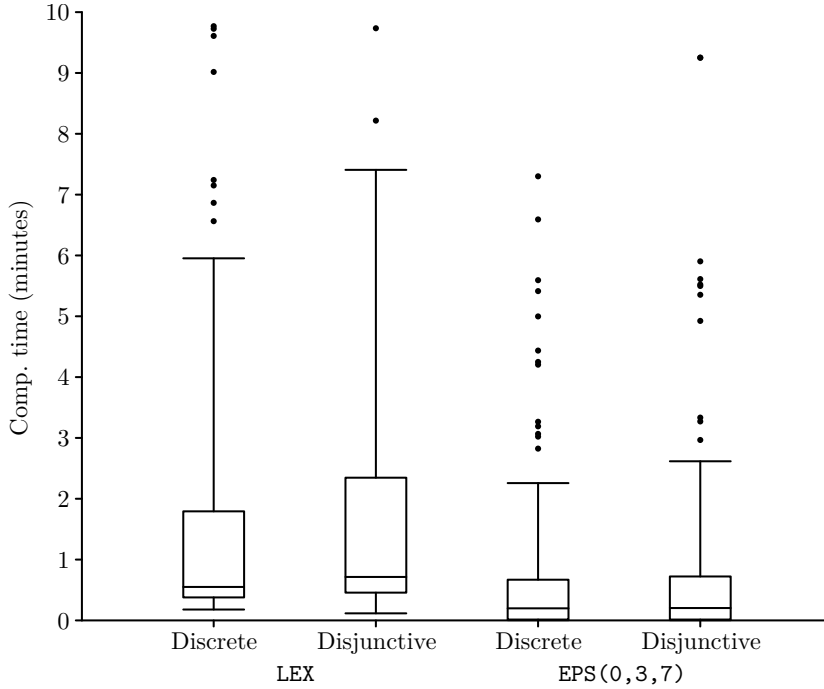


Figure 5: Box plots of computation times for the 100 instances in the class **Base**.

mances of the two models proposed in this paper, we use *performance profiles*, introduced by Dolan and Moré (2002) as a way of comparing methods w.r.t. relative performance as opposed to absolute performance. After solving a particular instance using both the discrete and disjunctive models and observing computation times of μ_1 and μ_2 , respectively, the *performance ratios* of the models are defined as $\mu_m / \min(\mu_1, \mu_2)$ for $m = 1, 2$. The *performance profile* of a method is then the cumulative distribution function of the performance ratio over all instances, and since a smaller performance ratio means better performance, the faster a performance profile converges to 1, the better the relative performance of the method.

Figure 5 shows a box plot of the computation times for each model/scalarization pair over all 100 instances in the class **Base**. Outliers are defined as computation times of more than three times the interquartile range, and are not shown in Figure 5 but instead given in Table 3. Using the scalarization $\text{EPS}(0, 3, 7)$ resulted in no outliers. Looking at all points within the 3rd quartile, it can be seen that around 75% of all instances are solved within 2 minutes using the scalarization **LEX** and within 1 minute using the scalarization $\text{EPS}(0, 3, 7)$. Moreover, most instances are solved within ten minutes, whereas the longest computation time observed is around 26 minutes.

The performance profiles of the models for the instance class **Base** are given in Figure 6. Overall the discrete model shows better performance compared to the disjunctive model, with the improvement being more significant for the scalarization **LEX** than for $\text{EPS}(0, 3, 7)$. The discrete model had the shortest computation time (i.e. a performance ratio of 1) in roughly 70% of the cases for the scalarization **LEX** and in roughly 60% of the cases for the scalarization $\text{EPS}(0, 3, 7)$.

The slightly better performance of the discrete model observed for the class **Base** is likely due to the fact that the instances in this class have a relatively high density of 0.79, whereas the time horizon of 20 time periods is not very long. As mentioned before, the discrete model is indeed expected to outperform the disjunctive model on dense instances

LEX	
Discrete	Disjunctive
14:58.89	10:18.50
16:27.25	12:40.58
	13:6.36
	17:35.26
	18:42.63
	18:49.35
	24:30.26
	25:10.63
	26:6.17

Table 3: Outliers beyond three times the interquartile range corresponding to the box plot given in Figure 5.

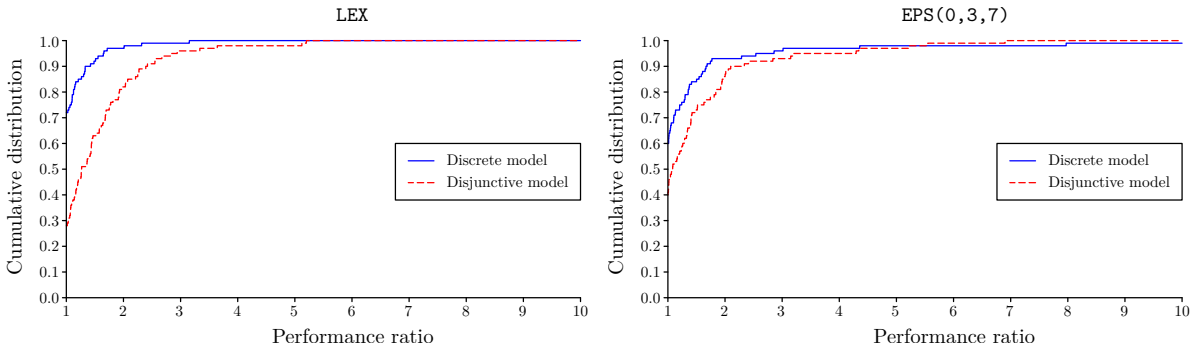


Figure 6: Performance profiles for the class Base.

with few time periods. This hypothesis is further supported by comparing the performance of the models on the instance classes given in Table 2. The performance profiles of the models for these instances are shown in Figure 7. Note that the classes designated here as “sparse” have a density of $\frac{1}{3}$ while those designated as “dense” have a density of $\frac{2}{3}$. Furthermore, for each class the performance profiles were computed using the four scalarizations LEX, EPS(0, 2, 3), EPS(0, 2, 5) and EPS(0, 2, 7) for each instance, resulting in a total of 400 comparisons of the two models for each of the six classes.

From Figure 7 it is observed that for the class Dense10, which contains dense instances with few time periods, the discrete model once again outperforms the disjunctive model. However, considering the classes Dense15 and Dense20 where the number of time periods are increased while the density remains fixed, the relative performance of the discrete model decreases, as expected. In fact, for the class Dense20 no model clearly dominates the other.

On the other hand, for the class Sparse20, which contains sparse instances with more time periods, it is observed that the disjunctive model now shows the best performance, as expected. Considering the classes Sparse15 and Sparse10 where the number of time periods are decreased while the density remains fixed, the relative performance of the discrete model increases, and for the class Sparse10 again no model clearly dominates the other.

Finally, we consider the effect of the minimum break length on the performance of the models, in particular with regards to feasibility and computation time. Table 4 shows, for different horizon lengths $|T|$, the percentage of instances belonging to the classes given in

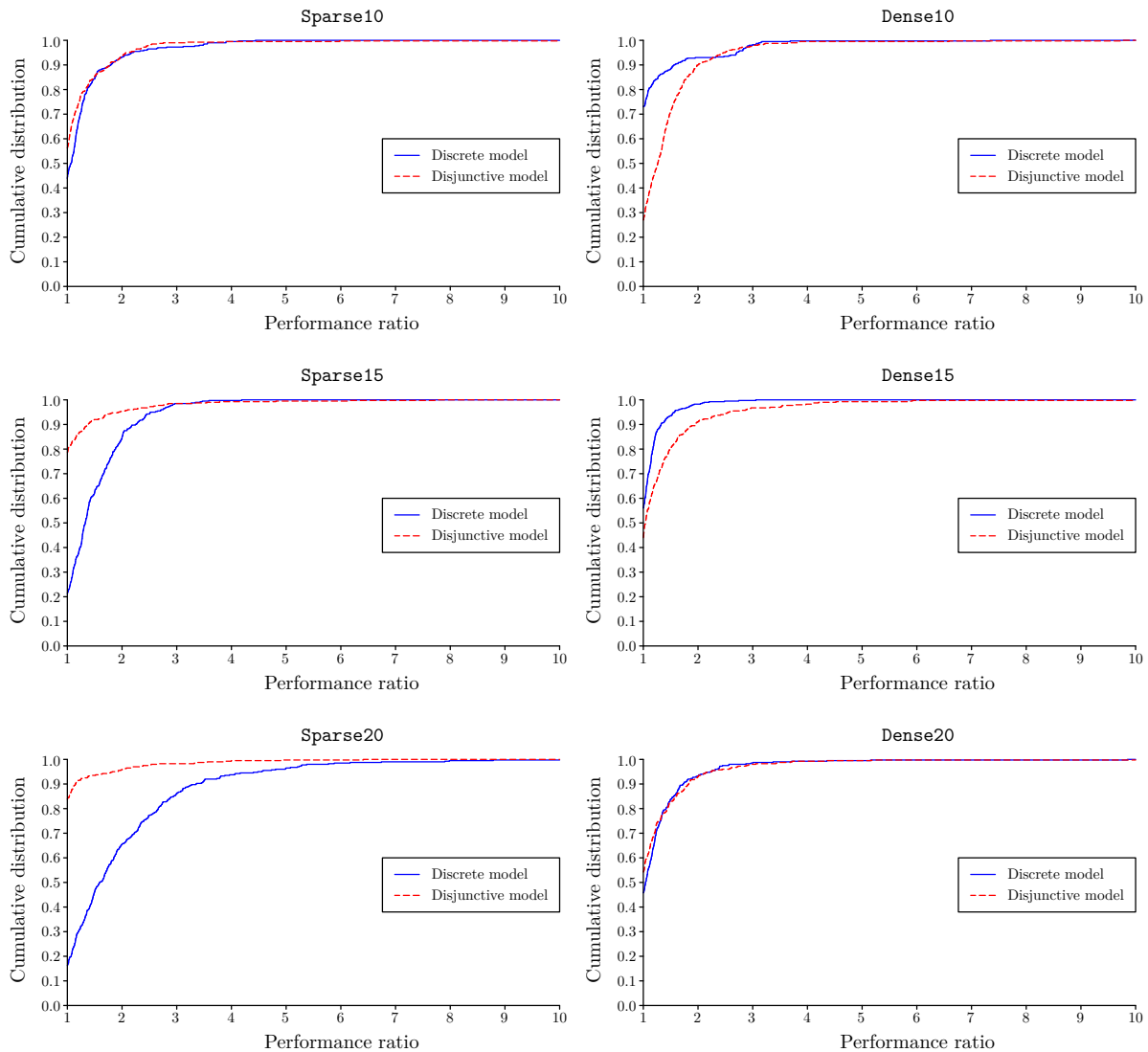


Figure 7: Performance profiles for the instances classes given in Table 2.

Table 2 that had no feasible solution given the minimum break lengths \bar{z}_{P_3} considered. Moreover, Figure 8 shows box-plots of the computation times (of both models) for the same categories of instances, here denoted by the ratio $\bar{z}_{P_3}/|T|$. It can be observed that, as the minimum break length increases, the number of infeasible instance increases while the computation time decreases. This is to be expected, as a decrease in the size of the search space due to tighter constraints typically leads to lower computation times together with a higher probability of infeasibility.

4.4 Discussion

Since we propose an optimization framework that is intended to be used in an interactive way, it is important that the computation times expected in practice are not prohibitively large. The results indeed show that in most cases a nondominated solution can be found within only a few minutes using a modest computational resource.

Moreover, we show that different modelling approaches could be used depending on the specific case. A discrete-time approach to modelling Principles 1 and 2 is promising since it allows one to include arbitrary cost functions on the lengths of occupation windows

	$\bar{z}_{P3} = 3$	$\bar{z}_{P3} = 5$	$\bar{z}_{P3} = 7$
$ T = 10$	28%	44%	64%
$ T = 15$	21%	28%	31%
$ T = 20$	25%	34%	35%

Table 4: Percentage of instances with a specific horizon length that become infeasible when imposing a specific lower bound for the minimum break length.

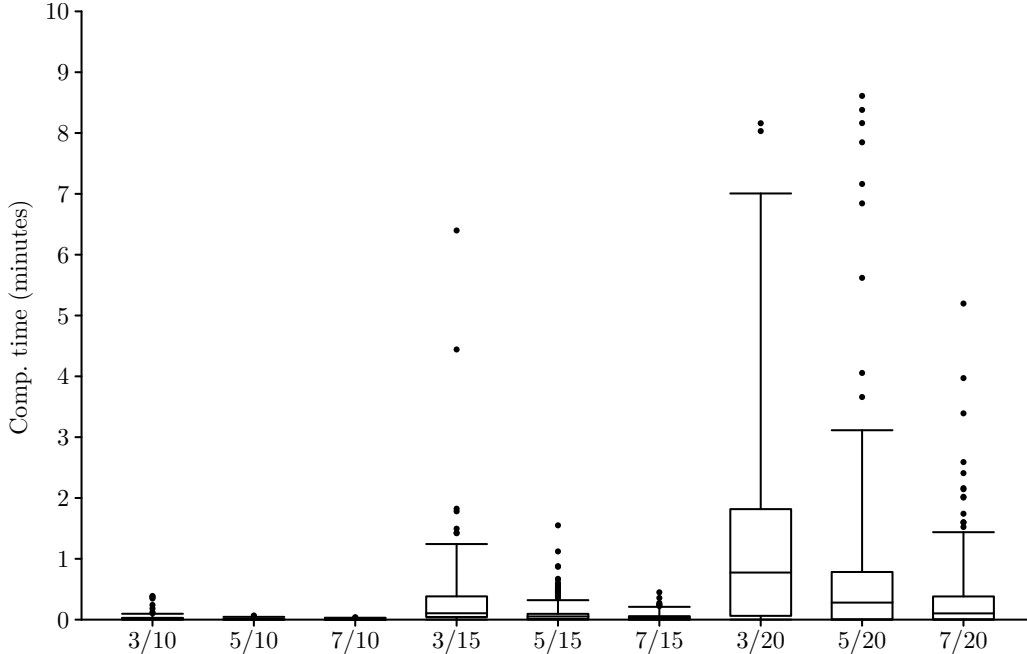


Figure 8: Box plots of computation times (for both models) for different configurations, denoted by $\bar{z}_{P3}/|T|$, i.e. the ratio of the minimum break length to the length of the time horizon (outliers of more than 10 minutes not shown).

and incentive functions on the lengths of breaks. However, its performance is restricted by the granularity of the discretization used. In cases where many time periods are considered, an approach based on the pairwise coordination of activities (occupying the same infrastructure resource) is a viable alternative, as it does not require a discretization of the time horizon. The performance of this approach, on the other hand, is sensitive to the number of activities occupying a single resource.

Finally, it is advisable for the planners to initiate an interactive approach with a very ambitious minimum break length. The results show that, even though infeasibility is more likely, it can be detected quickly by the models. The minimum break length can then be relaxed slightly, and the process continued until a feasible solution is found.

5 Conclusion

In this paper we propose three general principles to be taken into account in the strategic planning of railway infrastructure projects in order to reduce the disruptive effects these projects can have on the infrastructure users. By modelling these principles as objectives in addition to a resource levelling objective, we propose two models based on mixed integer programming and techniques from multiobjective optimization that can be used

in an interactive way by planners as decision support. Our results show that the models are capable of finding optimal (i.e. nondominated) solutions in short computation times, and that the two models have complimentary strengths based on specific features of the instances.

References

- A. R. Albrecht, D. M. Panton, and D. H. Lee. Rescheduling rail networks with maintenance disruptions using problem space search. *Computers and Operations Research*, 40(3):703–712, 2013.
- A. R. Andrade and P. F. Teixeira. Biobjective optimization model for maintenance and renewal decisions related to rail track geometry. *Transportation Research Record: Journal of the Transportation Research Board*, 2261:163–170, 2011.
- Banedanmark. An attractive railway — today and tomorrow, 2017a. URL http://uk.bane.dk/db/filarkiv/11981/Banedanmark_UK_net.pdf. Accessed on 2017-02-09.
- Banedanmark. Projects, 2017b. URL http://uk.bane.dk/visArtikel_eng.asp?artikelID=907. Accessed on 2017-02-09.
- N. Boland, T. Kalinowski, H. Waterer, and L. Zheng. Mixed integer programming based maintenance scheduling for the Hunter Valley coal chain. *Journal of Scheduling*, 16(6):649–659, 2013.
- N. Boland, T. Kalinowski, H. Waterer, and L. Zheng. Scheduling arc maintenance jobs in a network to maximize total flow over time. *Discrete Applied Mathematics*, 163:34–52, 2014.
- N. Boland, H. Charkhgard, and M. Savelsbergh. A criterion space search algorithm for biobjective integer programming: The balanced box method. *INFORMS Journal on Computing*, 27(4):735–754, 2015.
- S. Boğ, A. K. Nemani, and R. K. Ahuja. Iterative algorithms for the curfew planning problem. *Journal of the Operational Research Society*, 62(4):593–607, 2010.
- G. Budai, D. Huisman, and R. Dekker. Scheduling preventive railway maintenance activities. *Journal of the Operational Research Society*, 57(9):1035–1044, 2006.
- L. F. Caetano and P. F. Teixeira. Availability approach to optimizing railway track renewal operations. *Journal of Transportation Engineering*, 139(9):941–948, 2013.
- D. E. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.
- M. Ehrgott. A discussion of scalarization techniques for multiple objective integer programming. *Annals of Operations Research*, 147(1):343–360, 2006.
- S. E. Elmaghraby and J. Kamubowski. The analysis of activity networks under generalized precedence relations (GPRs). *Management Science*, 38(9):1245–1263, 1992.

- M. Forsgren, M. Aronsson, and S. Gestrelus. Maintaining tracks and traffic flow at the same time. *Journal of Rail Transport Planning and Management*, 3(3):111–123, 2013.
- A. Higgins. Scheduling of railway track maintenance activities and crews. *Journal of the Operational Research Society*, 49(10):1026–1033, 1998.
- A. R. Jenema. *An optimization model for a Train-Free-Period planning for ProRail based on the maintenance needs of the Dutch railway infrastructure*. Master thesis, Delft University of Technology, 2011.
- R. Kolisch and A. Sprecher. PSPLIB — A project scheduling problem library. *European Journal of Operational Research*, 96(1):205–216, 1997.
- M. R. Lake, L. Ferreira, and E. Kozan. Heuristic techniques for scheduling railway track maintenance. In E. Kozan and A. Ohuchi, editors, *Operations Research/Management Science at Work*, pages 177–187. Springer US, Boston, MA, 2002.
- T. Lidén. Railway infrastructure maintenance - a survey of planning problems and conducted research. *Transportation Research Procedia*, 10:574–583, 2015.
- T. Lidén and M. Joborn. An optimization model for integrated planning of railway traffic and network maintenance. *Transportation Research Part C: Emerging Technologies*, 74:327–347, 2017.
- K. Miettinen, J. Hakanen, and D. Podkopaev. Interactive nonlinear multiobjective optimization methods. In S. Greco, M. Ehrgott, and J. R. Figueira, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 927–976. Springer, 2016.
- A. K. Nemani, S. Boğ, and R. K. Ahuja. Solving the curfew planning problem. *Transportation Science*, 44(4):506–523, 2010.
- F. Peng and Y. Ouyang. Track maintenance production team scheduling in railroad networks. *Transportation Research Part B: Methodological*, 46(10):1474–1488, 2012.
- F. Peng and Y. Ouyang. Optimal clustering of railroad track maintenance jobs. *Computer-Aided Civil and Infrastructure Engineering*, 29(4):235–247, 2014.
- F. Peng, S. Kang, X. Li, Y. Ouyang, K. Somani, and D. Acharya. A heuristic approach to the railroad track maintenance scheduling problem. *Computer-Aided Civil and Infrastructure Engineering*, 26(2):129–145, 2011.
- H. Pouryousef, P. Teixeira, and J. Sussman. Track maintenance scheduling and its interactions with operations: Dedicated and mixed highspeed rail (HSR) scenarios. In *Proceedings of the 2010 Joint Rail Conference*, pages 1–10, 2010.
- RailNetEurope. Guidelines for coordination/publication of planned temporary capacity restrictions, version 2.0, 2015. URL <http://www.rne.eu/rneinhalt/uploads/2017/03/2015-12-03-Guidelines-CoTCR-V2.0.pdf>. Accessed on 2017-05-01.
- J. Rieck and J. Zimmermann. Exact methods for resource leveling problems. In C. Schwindt and J. Zimmermann, editors, *Handbook on Project Management and Scheduling Vol. 1*, pages 361–387. Springer Cham, 2015.

- J. Rieck, J. Zimmermann, and T. Gather. Mixed-integer linear programming for resource leveling problems. *European Journal of Operational Research*, 221(1):27–37, 2012.
- J. I. Van Zante-De Fokkert, D. Den Hertog, F. J. Van Den Berg, and J. H. M. Verhoeven. The netherlands schedules track maintenance to improve track workers' safety. *Interfaces*, 37(2):133–142, 2007.
- J. Zhao, A. H. C. Chan, and M. P. N. Burrow. A genetic-algorithm-based approach for scheduling the renewal of railway track components. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 223(6):533–541, 2009.