# An Empirical Analysis of Robustness Concepts for Timetabling*

**Marc Goerigk and Anita Schöbel**

**Institute for Numerical and Applied Mathematics**
**University of Göttingen**
**Lotzestr. 16-18**
**D-37083 Göttingen, Germany**
**{m.goerigk,schoebel}@math.uni-goettingen.de**

**Abstract**

Calculating timetables that are insensitive to disturbances has drawn considerable research efforts due to its practical importance on the one hand and its hard tractability by classical robustness concepts on the other hand. Many different robustness concepts for timetabling have been suggested in the literature, some of them very recently. In this paper we compare such concepts on real-world instances. We also introduce a new approach that is generically applicable to any robustness problem. Nevertheless it is able to adapt the special characteristics of the respective problem structure and hence generates solutions that fit to the needs of the respective problem.

**1998 ACM Subject Classification** G.2.2 Graph Theory - Network problems

**Keywords and phrases** Timetabling, Robust Optimization, Algorithm Engineering

**Digital Object Identifier** 10.4230/OASIcs.ATMOS.2010.100

## 1 Introduction

The *aperiodic timetabling problem* has received considerable attention in recent robust optimization literature (see, e.g., [7, 9, 11]) as one of significant importance in real-world applications where it is needed to create timetables that stay "good" under the unavoidable small disturbances of daily railway operations. Robust solutions usually lead to high buffer times, which in turn yield high traveling times and thus unattractive timetables. Newly introduced concepts are all in between the extremes of the best nominal timetable, which is least robust, and the strictly robust timetable, which tends to be too conservative.

In this paper we compare for the first time the most prominent robustness concepts for timetabling numerically on a real-world instance. We furthermore present a new concept for finding robust solutions with an easily applicable algorithm, yielding timetables that are a good compromise between traveling time and robustness. In general, this algorithm can be used whenever a solver for the nominal problem is at hand, which gives the possibility to make use of existing, powerful methods with small effort of software rewriting.

We analyze two different types of uncertainty, one that allows small delays on all edges, and one that allows heavy delays on a restricted set of edges, and show empirically that the structure of these determine which robustness concept fits best.

The problem we consider is the following: Let an *event-activity-network* (EAN) be given, that is, a directed graph $G = (\mathcal{E}, \mathcal{A})$ consisting of departure and arrival *events* $\mathcal{E} = \mathcal{E}^{\text{arr}} \cup \mathcal{E}^{\text{dep}}$

* This work was partially supported by grant SCHO 1140/3-1 within the DFG programme *Algorithm Engineering*.


10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS '10).
Editors: Thomas Erlebach, Marco Lübbecke; pp. 100–113
OpenAccess Series in Informatics
Schloss Dagstuhl Publishing, Germany

and waiting, driving, changing and headway *activities* $\mathcal{A} = \mathcal{A}^{\text{wait}} \cup \mathcal{A}^{\text{drive}} \cup \mathcal{A}^{\text{change}} \cup \mathcal{A}^{\text{head}}$. Driving activities $\mathcal{A}^{\text{drive}} \subseteq \mathcal{E}^{\text{dep}} \times \mathcal{E}^{\text{arr}}$ represent traveling from one station to another, while waiting activities $\mathcal{A}^{\text{wait}} \subseteq \mathcal{E}^{\text{arr}} \times \mathcal{E}^{\text{dep}}$ represent staying of a train at a station while passengers board and deboard. Changing activities $\mathcal{A}^{\text{change}} \subset \mathcal{E}^{\text{arr}} \times \mathcal{E}^{\text{dep}}$ model passengers who plan to change from one train to another at the same station, while headways $\mathcal{A}^{\text{head}} \subset \mathcal{E}^{\text{dep}} \times \mathcal{E}^{\text{dep}}$ are introduced to model safety distances between trains sharing the same infrastructure. Assigned to each of these activities $(i, j) \in \mathcal{A}$ is a minimal duration $\hat{l}_{ij} \in \mathbb{N}$ representing the technically possible lower time bound for an activity to take place, and a number $w_{ij}$ of passengers using activity $(i, j) \in \mathcal{A}$. The task is to find node potentials $\pi_i \in \mathbb{R}$ for all $i \in \mathcal{E}$, such that the sum of passenger traveling times $w_{ij}(\pi_j - \pi_i)$ over all activities $(i, j) \in \mathcal{A}$ is minimized for given passenger weights $w_{ij}$ under the time restrictions $\pi_j - \pi_i \geq \hat{l}_{ij}$ for each activity $(i, j) \in \mathcal{A}$. Its well-known mathematical formulation is

$$\text{(TT)} \quad \min \sum_{(i,j)\in\mathcal{A}} w_{ij}(\pi_j - \pi_i) \tag{1}$$

$$\text{s.t.} \quad \pi_j - \pi_i \geq \hat{l}_{ij} \quad \forall (i, j) \in \mathcal{A} \tag{2}$$

$$\pi_i \geq 0 \quad \forall i \in \mathcal{E}. \tag{3}$$

We will sometimes simply write $\hat{l} = (\hat{l}_{ij})_{(i,j)\in\mathcal{A}}$ as the vector of all lower bounds, and similary $\pi = (\pi_i)_{i\in\mathcal{E}}$ as the vector of node potentials, for any given edge- and node order. Note that the time restrictions form a totally unimodular matrix, i.e. even though real node potentials might be considered as unrealistic in railway operations, we will always find an integer optimal solution. Furthermore, (TT) is feasible for all possible activity durations $\hat{l} \geq 0$ if the network does not contain any directed cycle with positive length, infeasible otherwise.

## 2 Robustness Concepts

In order to hedge (TT) against delays in operation, we have to model the possible disturbances first. Which (source) disturbances occur is in practice not known beforehand, since this depends on exterior influences like weather conditions or technical failures. Hence the activity durations are uncertain. In this paper we assume that the passenger distribution $w = (w_{ij})_{(i,j)\in\mathcal{A}}$, i.e., the number of passengers using each activity, is known.

The first type of uncertainty we consider is one of *uniform deviation*. Imagine, for example, bad weather conditions that *slightly* delay all trains on track equally. We model this behavior with the following set of scenarios depending on $s \in \mathbb{R}^+$, where $s$ controls the level of uncertainty:

$$\mathcal{U}_1(s) := \{l : \hat{l}_{ij} \leq l_{ij} \leq (1 + s)\hat{l}_{ij} \ \forall (i, j) \in \mathcal{A}^{\text{drive}} \cup \mathcal{A}^{\text{wait}}, \\ l_{ij} = \hat{l}_{ij} \ \forall (i, j) \in \mathcal{A}^{\text{change}} \cup \mathcal{A}^{\text{head}}\}$$

The second type of uncertainty we analyze models the situation that only a restricted number of activities may be delayed at the same time, but *heavier*. E.g., this may be the case when good weather conditions hold but single trains are delayed by blocked tracks or technical failures. For $k \geq 1$, we define

$$\mathcal{U}_2(k, s) := \{l : \hat{l}_{ij} \leq l_{ij} \leq (1 + s)\hat{l}_{ij} \ \forall (i, j) \in D \subseteq \mathcal{A}^{\text{drive}} \cup \mathcal{A}^{\text{wait}}, \ |D| = k, \\ l_{ij} = \hat{l}_{ij} \ \forall (i, j) \in \mathcal{A} \setminus D\}$$

Using $\mathcal{U}_2$ we assume that not all, but at most $k$ lower bounds change to their worst values in the same scenario, which can be interpreted in the sense of Bertsimas and Sim [3] in the dual problem.

We now survey recent robustness concepts and show how they can be applied to the timetabling problem. To this end, let us consider a general optimization problem $(P)$ $\min\{f(x) : F(x) \leq 0\}$ with an objective function $f : \mathbb{R}^n \to \mathbb{R}$ and constraints $F : \mathbb{R}^n \to \mathbb{R}^m$. Its uncertain version is given as

$$\text{(P}(\xi)) \quad \min\ f(x,\xi) \tag{4}$$

$$\text{s.t.} \quad F(x,\xi) \leq 0 \tag{5}$$

$$x \in \mathbb{R}^n, \tag{6}$$

depending on the scenario parameter $\xi$ from a given uncertainty set $\mathcal{U} \subseteq \mathbb{R}^M$. There may be a specific element $\hat{\xi} \in \mathcal{U}$ that models the problem as it would be without the existence of disturbances. This element is called the *nominal scenario* and $(P(\hat{\xi}))$ is called the *nominal problem.* We consider problem (TT) as an uncertain optimization problem w.r.t $l$, where the objective function $f(\pi) = \sum_{(i,j)\in\mathcal{A}} w_{ij}(\pi_j - \pi_i)$ does not depend on $l$ and the constraints are given as $F(\pi, l) = (l - A^t\pi)$, where $A$ is the node-arc incident matrix of $G$, that is, $a_{ie} = 1$, if $e = (j,i)$ for a $j \in \mathcal{E}$, $a_{ie} = -1$, if $e = (i,j)$ for a $j \in \mathcal{E}$, and $a_{ie} = 0$ else, and $l \in \mathbb{R}^{|\mathcal{A}|}$ contains the minimum activity durations. Note that for $\mathcal{U}_1$ and $\mathcal{U}_2$, we have $M = m$.

## 2.1 Strict Robustness

Strict robustness might be considered as the oldest and most conservative approach to uncertainty. It was introduced by Soyster [13] and significantly extended by Ben-Tal, Ghaoui and Nemirovski, see [2, 1] and references therein. The concept requires feasibility of a robust solution under all possible scenarios, i.e. that $F(x,\xi) \leq 0$ for all $\xi \in \mathcal{U}$. For (TT) we obtain

$$\text{(S-TT)} \quad \min \sum_{(i,j)\in\mathcal{A}} w_{ij}(\pi_j - \pi_i) \tag{7}$$

$$\text{s.t.} \quad \pi_j - \pi_i \geq l_{ij} \quad \forall (i,j) \in \mathcal{A} \text{ and } \forall l \in \mathcal{U} \tag{8}$$

$$\pi \geq 0. \tag{9}$$

(S-TT) is called the strict robust counterpart of (TT). In general, this leads to infinitely many constraints, depending on the choice of $\mathcal{U}$. It is shown in [2] that if $\mathcal{U} = conv\{\xi^1, \dots, \xi^N\}$, where *conv* denotes the convex hull, and $F(x,\cdot), f(x,\cdot)$ are quasiconvex in $\xi$, then the strict robust counterpart is equivalent to a program where the constraints only have to be satisfied for $\xi^1, \dots, \xi^N$. This is evidently the case for (TT) with $\mathcal{U}_1$ as defined above. Omitting redundant constraints we hence gain the following strict robust formulation for $\mathcal{U}_1$:

$$\text{(S-TT)} \quad \min \sum_{(i,j)\in\mathcal{A}} w_{ij}(\pi_j - \pi_i) \tag{10}$$

$$\text{s.t.} \quad \pi_j - \pi_i \geq (1+s)\hat{l}_{ij} \quad \forall\ (i,j) \in \mathcal{A} \tag{11}$$

$$\pi \geq 0 \tag{12}$$

In case of $\mathcal{U}_2$, the same result holds due to the fact that all but the listed constraints (11) become dominated by other scenarios. Remark that the guaranteed feasibility comes at a high price, as the maximum buffer is put on every edge even though only a few may become delayed.

## 2.2 Light Robustness

Fischetti and Monaci introduced in [9] an approach that relaxes the constraints of the strict robust counterpart to gain more flexibility. As before, let $m$ be the number of constraints.

Variables $\gamma_i$ are introduced for each constraint $i = 1, \ldots, m$ of the nominal problem that measure the degree of relaxation needed for strict robustness. The goal is to minimize the sum of these $\gamma_i$ while guaranteeing a certain quality of the solution. Let the nominal scenario be denoted by $\hat{\xi} \in \mathcal{U}$ and let $z^* > 0$ be the optimal objective of the nominal problem. Then, for a given $\delta$, the light robustness approach is

$$\text{(LR)} \quad \min \sum \gamma_i \tag{13}$$
$$\text{s.t.} \quad F(x, \hat{\xi}) \leq 0 \tag{14}$$
$$f(x, \hat{\xi}) \leq (1 + \delta) z^* \tag{15}$$
$$F_i(x, \xi) \leq \gamma_i \quad \forall i = 1, \ldots, m, \ \forall \xi \in \mathcal{U} \tag{16}$$
$$\gamma \geq 0 \tag{17}$$

Constraint (14) ensures nominal feasibility, while (15) controls the nominal quality by the parameter $\delta$. Constraints (16) allow infeasibility for the other scenarios $\xi \in \mathcal{U}$, which will be minimized by the objective function.

Applying this scheme to the timetabling problem (TT) with uncertainty $\mathcal{U}_1$ and dropping dominated constraints gives the following program:

$$\text{(L-TT)} \quad \min \sum \gamma_{ij} \tag{18}$$
$$\text{s.t.} \quad \sum w_{ij}(\pi_j - \pi_i) \leq (1 + \delta) z^* \tag{19}$$
$$\pi_j - \pi_i \geq \hat{l}_{ij} \quad \forall (i, j) \in \mathcal{A} \tag{20}$$
$$\pi_j - \pi_i \geq (1 + s)\hat{l}_{ij} - \gamma_{ij} \quad \forall (i, j) \in \mathcal{A} \tag{21}$$
$$\gamma, \pi \geq 0 \tag{22}$$

Note that $\hat{l}$ is used as the nominal scenario. Constraint (16) simplifies to (21), as all other scenarios $l \in \mathcal{U}$ become dominated. Also here $\mathcal{U}_2$ yields the same formulation as we can again drop dominated constraints.

## 2.3 Recoverable Robustness

The concept of Recoverable Robustness was introduced by Liebchen et al. in [11] and by Cicerone et al. in [4, 8, 5], both groups also proposing applications to timetabling. The basic idea is to find a robust solution that can be "repaired" (i.e., made feasible by delaying events) with low costs as soon as the real scenario becomes known. In both papers [11, 6], the sum of all arrival delays of the passengers and the maximum delay of each arrival event are restricted by budget parameters $\lambda_1$ or $\lambda_2$. As these budget parameters might be difficult to estimate in advance, they are regarded as variables in [11] and become part of the objective function with according weights, say $g_1$ and $g_2$. Denoting by $\tilde{w}_i$, $i \in \mathcal{E}^{\text{arr}}$, the number of passengers de-boarding at event $i$, and assuming a finite set of scenarios $\mathcal{U}$, [11] suggest the following program:

$$\text{(R-TT)} \quad \min \sum_{(i,j) \in \mathcal{A}} w_{ij}(\pi_j - \pi_i) + g_1 \lambda_1 + g_2 \lambda_2 \tag{23}$$
$$\text{s.t.} \quad \pi_j - \pi_i \geq \hat{l}_{ij} \quad \forall (i, j) \in \mathcal{A} \tag{24}$$
$$\pi_j^l - \pi_i^l \geq l_{ij} \quad \forall l \in \mathcal{U}, \ \forall \ (i, j) \in \mathcal{A} \tag{25}$$
$$\pi_i^l \geq \pi_i \quad \forall l \in \mathcal{U}, \ \forall i \in \mathcal{E}^{\text{dep}} \tag{26}$$

$$\sum_{i \in \mathcal{E}^{\text{arr}}} \tilde{w}_i(\pi_i^l - \pi_i) \leq \lambda_1 \quad \forall l \in \mathcal{U} \tag{27}$$

$$\pi_i^l - \pi_i \leq \lambda_2 \quad \forall l \in \mathcal{U},\ \forall i \in \mathcal{E}^{\text{arr}} \tag{28}$$

$$\lambda_1, \lambda_2, \pi^l, \pi \geq 0 \tag{29}$$

Regarding the number of variables, note that for each scenario a timetabling problem has to be solved. The concept was originally designed for an uncertainty of type $\mathcal{U}_2$, meaning that $\binom{|\mathcal{A}|}{k} + |\mathcal{A}| + 2$ variables need to be created. For $k > 1$ this becomes quickly intractable. For $k = 1$ exactly one activity is delayed per scenario and we may write $\mathcal{U} \cong \mathcal{A}^{\text{wait}} \cup \mathcal{A}^{\text{drive}}$ for short. The authors present a possibility to reformulate the recovery robust timetabling problem in a more compact way by setting $g_2 = 0$ and introducing a fixed recovery budget $D$ instead of using $\lambda_1$. For every scenario $e$ variables $y^e = \pi^e - \pi \in \mathbb{R}^{|\mathcal{E}|}$ are needed. Using slack variables $f$ one obtains

$$\text{(R2-TT)} \quad \min_{\pi, f} \sum_{(i,j) \in \mathcal{A}} w_{ij}(\pi_j - \pi_i) \tag{30}$$

$$\text{s.t.} \quad \pi_j - \pi_i - f_{ij} = \hat{l}_{ij} \quad \forall (i,j) \in \mathcal{A} \tag{31}$$

$$f_{ij} + y_j^e - y_i^e \geq s\chi_{ij}(e) \quad \forall (i,j) \in \mathcal{A},\ \forall e \in \mathcal{A}^{\text{wait}} \cup \mathcal{A}^{\text{drive}} \tag{32}$$

$$D \geq \|y^e\|_1 \quad \forall e \in \mathcal{A}^{\text{wait}} \cup \mathcal{A}^{\text{drive}} \tag{33}$$

$$f, y^e, \pi \geq 0, \tag{34}$$

where $\chi_{ij}(e) = 1$ if $e = (i,j)$ and zero else. In this formulation we changed the weights $\tilde{w}$ to be 1 for all nodes for better comparability with other models; however, also other weights may be considered.

For the uncertainty $\mathcal{U}_1$ we obtain a different formulation. Here it is sufficient to find a recovery solution $\pi^{worst}$ for only the worst-case scenario in which all activity durations take their worst values. Hence, by setting $\tilde{w}_i = 1$ for all $i \in \mathcal{E}$ again, (R-TT) simplifies to

$$\text{(R1-TT)} \quad \min \sum_{(i,j) \in \mathcal{A}} w_{ij}(\pi_j - \pi_i) + g_1\lambda_1 + g_2\lambda_2 \tag{35}$$

$$\text{s.t.} \quad \pi_j - \pi_i \geq \hat{l}_{ij} \quad \forall (i,j) \in \mathcal{A} \tag{36}$$

$$\pi_j^{worst} - \pi_i^{worst} \geq (1+s)\hat{l}_{ij} \quad \forall (i,j) \in \mathcal{A} \tag{37}$$

$$\pi_i^{worst} \geq \pi_i \quad \forall\ i \in \mathcal{E} \tag{38}$$

$$\|\pi^{worst} - \pi\|_1 \leq \lambda_1 \tag{39}$$

$$\pi_i^{worst} - \pi_i \leq \lambda_2 \quad \forall i \in \mathcal{E} \tag{40}$$

$$\lambda_1, \lambda_2, \hat{\pi}, \pi \geq 0. \tag{41}$$

## 3 A New Approach: Recover to Optimality

In this section we consider a new type of robust approach that aims to minimize the expected or the maximum repair costs to an *optimal* solution of a scenario, measured in terms of a distance function. The robust counterpart of this setting is in general notation given as

$$\text{(RecOpt)} \quad \min_x \sup_{\xi \in \mathcal{U}} d(x, x^\xi)$$

$$\text{s.t.} \quad x^\xi \text{ is an optimal solution to } (\text{P}(\xi)),$$

where $d(x, x^\xi)$ represents the recovery costs needed to update a timetable $x$ to another timetable $x^\xi$. Instead of the supremum also the average recovery costs may be considered.

Since we recover not to a feasible, but to an *optimal* solution $x^\xi$, a strictly robust solution has no recovery costs in (R-TT), but especially in timetabling will usually have high recovery costs in the sense of (RecOpt). Recovery to optimality may also mean to let events take place *earlier* which is reasonable when a timetable needs not be adapted to the scenario during the operational phase, but the scenario is known some time before (like in the case of track maintenance or exceptional weather forecasts).

This concept therefore generalizes several well-known approaches of robust optimization theory. As an example, in *min max regret* literature (see [10] for an overview, or [14] where it is called *deviation robustness*), one considers the problem of minimizing the difference between the objective value of the current solution and the one that would have been best for the scenario, that is:

(MinMaxReg) $\min \max_{\xi \in \mathcal{U}} \; f(x, \xi) - f^*(\xi)$,

where $f^*(\xi)$ denotes the best possible solution for scenario $\xi$. In contrast to this, our approach aims at minimizing the distance between the current *solution* $x$ and the solution that would have been best for the scenario. Minmax-regret robustness is hence a special case of (RecOpt) by using the difference in the objectives as a distance measure. Also, the problem of finding a *strict robust* solution can be considered as a (RecOpt) problem, where the objective value is required to be zero. Compared to *recoverable robustness*, we recover to optimality, not to feasibility, and allow any distance measure $d$.

Instead of solving (RecOpt) to optimality we suggest the following heuristic in which we create a number of scenarios $\xi$, solve them separately, and find the robust solution by solving a location problem in which the given facilities are the respective optimal solutions of the instances (P($\xi$)). Thus we apply the following algorithm to the timetabling problem:

**Algorithm** `RecOpt-TT`:

→ **Input:** A robust aperiodic timetabling instance (TT), a sample size $\nu \in \mathbb{N}$ and a distance measure $d : \mathbb{R}^{|\mathcal{E}|} \times \mathbb{R}^{|\mathcal{E}|} \to \mathbb{R}$.

1. Choose a subset $S \subseteq \mathcal{U}$ of $\nu$ elements at random.
2. Create vectors $\pi^l \in \mathbb{R}^{|\mathcal{E}|}$ by solving TT($l$) for each $l \in S$.
3. Find a vector $\pi \in \mathbb{R}^{|\mathcal{E}|}$ by minimizing the sum/maximum of distances $d(\pi, \pi^l)$ for all $l \in S$.

← **Output:** A robust solution $\pi$.

This algorithm is generically applicable to any other robust problem, but has to be specified to its respective needs. In particular, we have to determine which distance measure represents the recovery costs best, how many scenarios should be chosen, and how they should be created.

Note that this heuristic is easily applicable whenever a method for solving the nominal problem is available. Only a generic location problem solver has to be used, while existing algorithms need not be changed.

For our numerical evaluation we used the same recovery costs as in (R1-TT) and (R2-TT), which is the $\|\cdot\|_1$-distance, either in combination with a sum or a maximum, and we added the squared Euclidean distance as third alternative. The resulting combinations are shown in Table 1.

Note that we are free to add further restrictions to the location of $\pi$. Since a nominal infeasible timetable would not be of practical use, we additionally impose nominal feasibility constraints and solve restricted location problems. We remark that there can be an optimal $d_1$ center or median for the timetabling problem, that is not feasible for the nominal scenario. In contrast to this, the centroid is always feasible, as Lemma 1 shows.

| Distance (recovery costs) | sum/max | Name | Calculation |
|---|---|---|---|
| $d_1(x,y) = \|x-y\|_1$ | sum | $d_1$ median | $\text{argmin}_\pi \sum_{l\in S}\sum_{i\in\mathcal{E}} \|\pi_i - \pi_i^l\|$ |
| $d_1(x,y) = \|x-y\|_1$ | max | $d_1$ center | $\text{argmin}_\pi \max_{l\in S}\sum_{i\in\mathcal{E}} \|\pi_i - \pi_i^l\|$ |
| $d_2^2(x,y) = \|x-y\|_2^2$ | sum | centroid | $\frac{1}{\|S\|}\sum_{l\in S}\pi^l$ |

**Table 1** Evaluated distance - sum/max combinations.

▶ **Lemma 1.** *Let $(P(b))$ be an uncertain problem with constraints $Ax \geq b$ only depending on the right-hand side. Let $\hat{b} \in \mathcal{U}$ be the nominal scenario with $\hat{b} \leq b$ for all $b \in \mathcal{U}$. Let $S \subseteq \mathcal{U}$ be a finite set and let $x^b$ be an optimal solution to (P(b)) for all $b \in S$. Then the centroid, i.e. the solution to $\min_x \sum_{b\in S}\|x - x^b\|_2^2$, is nominal feasible.*

**Proof.** Let $x \in \mathbb{R}^n$ be the centroid. For the $k$th constraint, we obtain:

$$\sum_{i=1}^n a_{ki}x_i = \sum_{i=1}^n a_{ki}\frac{1}{|S|}\sum_{b\in S}x_i^b = \frac{1}{|S|}\sum_{b\in S}\sum_{i=1}^n a_{ki}x_i^b \geq \frac{1}{|S|}\sum_{b\in S}b_k \geq \frac{1}{|S|}\sum_{j=1}^{|S|}\hat{b}_k = \hat{b}_k$$

◀

▶ **Corollary 2.** *Let a (TT) instance with an uncertainty set $\mathcal{U}_1$ or $\mathcal{U}_2$ be given, and let $d = d_2^2$. Then the robust solution calculated by the sum version of (RecOpt-TT) is nominal feasible for any finite set $S \subseteq \mathcal{U}$.*

This result naturally extends to interval-based uncertainties of the form $[\hat{b}-\epsilon, \hat{b}+\delta]$ with $\delta > \epsilon$, i.e., the nominal scenario does not need to be the smallest one. By the law of large numbers the centroid is nominal feasible for $\nu \to \infty$ and a uniformly distributed choice of scenarios.

Concerning the amount of scenarios, we tested numerically how many scenarios were needed for a convergence of solutions. This was already the case for less than 100 instances on the instances described in Section 4.

Finally, we have to decide how to choose the subset $S \subseteq \mathcal{U}$. For finite $\mathcal{U}$, we may simply choose the whole set, but this approach is not possible anymore for infinite sets. We now present a sufficient condition under which the choice of a finite subset solves (RecOpt) exactly.

▶ **Theorem 3.** *Let $\mathcal{U} = \text{conv}\{\xi^1,\ldots,\xi^N\} \subseteq \mathbb{R}^M$ and let $d(x,\cdot)$ be convex in its second argument. Let $x : \mathbb{R}^M \to \mathbb{R}^n$ assign an optimal solution $x(\xi)$ to any scenario $\xi$, and assume that $x$ is affine linear. By writing $x^i := x(\xi^i)$ for short we have*

1. *For all $\xi \in \mathcal{U}$: $x(\xi) \in \text{conv}\{x^1,\ldots,x^N\}$.*
2. *The center of $x^1,\ldots,x^N$ with respect to the distance measure $d$ solves (RecOpt).*

**Proof.** Let $\xi \in \mathcal{U}$, i.e. there exist $\lambda_i$, $i = 1,\ldots,N$ with $0 \leq \lambda_i \leq 1$, $\sum_{i=1}^N \lambda_i = 1$ and $\xi = \sum_{i=1}^N \lambda_i\xi^i$. Then we obtain

$$x(\xi) = x\left(\sum_{i=1}^N \lambda_i\xi^i\right) = \sum_{i=1}^N \lambda_i x(\xi^i) = \sum_{i=1}^N \lambda_i x^i,$$

i.e. $x(\xi) \in \text{conv}\{x^1,\ldots,x^N\}$. Concerning the second part of the theorem, define $r^* := \max_{i=1,\ldots,N} d(x^*, x^i)$ as the radius of the center $x^*$ and let $\bar{r}$ be the best possible objective value for (RecOpt). Since $r^* \leq \bar{r}$ it remains to show that the recovery radius of $x^*$ with respect to $\mathcal{U}$ equals $r^*$, i.e. that $d(x^*, x(\xi)) \leq r^*$ for all $\xi \in \mathcal{U}$.

To this end, let $\xi \in \mathcal{U}$. Then $x(\xi) \in conv\{x^1, \ldots, x^N\}$ and hence there are $\lambda_i$, $i = 1, \ldots, N$, with $0 \le \lambda_i \le 1$, $\sum_{i=1}^N \lambda_i = 1$ and $\sum_{i=1}^N \lambda_i x^i = x(\xi)$. Quasi-convexity of $d(x^*, \cdot)$ yields

$$d(x^*, x(\xi)) = d(x^*, \sum_{i=1}^N \lambda_i x^i) \le \max_{i=1}^N d(x^*, x^i) = r^*,$$

hence $x^*$ is in fact optimal for (RecOpt). ◀

This raises the question, when the solution mapping $x$ is indeed affine linear. We present some results on general linear programs with uncertain right-hand side, i.e.

$$(P(b)) \quad \min\{c^t x : Ax = b, x \ge 0, x \in \mathbb{R}^n\}, b \in \mathcal{U}, \tag{42}$$

where $A \in \mathbb{R}^{m \times n}$.

▶ **Lemma 4.** *Consider $(P(b))$ with a convex uncertainty set $\mathcal{U} \subseteq \mathbb{R}^M$ and assume that $int(\mathcal{U}) \neq \emptyset$, where $int(\mathcal{U})$ denotes the interior of $\mathcal{U}$. Then $x : \mathbb{R}^M \to \mathbb{R}^n$ as defined in Theorem 3 is an affine linear function if and only if there exists a basis $B \subseteq \{1, \ldots, n\}$ with non-negative reduced costs*[1] *and $A_B^{-1} b \ge 0$ for all $b \in \mathcal{U}$.*

**Proof.**
- "if": Let $B$ be such a basis. Since the reduced costs $c_n^t - c_B^t A_B^{-1} A_n \ge 0$ are independent of $b$ and feasibility of the corresponding basic solution is ensured for all $b \in \mathcal{U}$ we know from linear programming theory that $x(b) := (A_B^{-1} b, 0)$ is optimal for (P($b$)). Hence, $x(b)$ is an affine linear function.
- "only if": Choose any $b^0 \in \text{int}(\mathcal{U})$ and solve the linear program. This yields a basis $B$ with nonnegative reduced costs and $A_B^{-1} b^0 \ge 0$, i.e. $x(b^0) = (A_B^{-1} b^0, 0)$ is an optimal solution.
  As $b^0 \in \text{int}(\mathcal{U})$ we can find for every unit vector $e_i \in \mathbb{R}^M$ an $\epsilon_i$ and a direction $d_i \in \{-1, +1\}$ such that
  $$b^i := b^0 + \epsilon_i d_i e_i \in \mathcal{U}$$
  and $A_B^{-1} b^i \ge 0$. Hence, $B$ is an optimal basis for $b^0, b^1, \ldots, b^M$, i.e. we have $x(b^i) = (A_B^{-1} b^i, 0)$ for $i = 0, 1, \ldots, M$. Due to our assumption $x(b)$ is affine linear; hence it is uniquely determined on the set of $\{b^0, b^1, \ldots, b^M\}$ of $M+1$ affinely independent points. This yields $x(b) = (A_B^{-1} b, 0)$ for all $b \in \mathcal{U}$, in particular we have $A_B^{-1} b \ge 0$ for all $b \in \mathcal{U}$.

◀

Note that the uncertainty $\mathcal{U}_1$ is a polyhedral set with a finite number of extreme points, while $\mathcal{U}_2$ is not convex for fixed $k, s$. By introducing slack variables $f$ as in (R2-TT), we may rewrite the constraints $\pi_j - \pi_i \ge l_{ij}$ of the timetabling problem to $\pi_j - \pi_i - f_{ij} = b_{ij}$. We hence gain the following corollary to Lemma 4:

▶ **Corollary 5.** *Let a (TT) instance with an uncertainty set $\mathcal{U} = conv\{l^1, \ldots, l^N\}$ be given. Assume that there is a basis $B$ that is optimal for each scenario $l \in \mathcal{U}$. Then the $d_1$ center with respect to the solutions $x^{l^1}, \ldots x^{l^N}$ solves (RecOpt) applied to the timetabling problem optimally, i.e. the choice $S = \{l^1, \ldots, l^N\}$ in step 1 of (RecOpt-TT) leads to an exact optimal solution.*

[1] For the definition of reduced costs, see any introductory textbook on linear optimization, e.g., *Linear Optimization and Extensions: Theory and Algorithms*, by Fang and Puthenpura, Prentice Hall, 1993.

In the following we will investigate again (P($b$)) but with the additional assumption that the uncertainty set $\mathcal{U} \subseteq \mathbb{R}^m$ (in this case, $M = m$) is symmetric with respect to some specified vector $b^* \in \mathbb{R}^m$, that is, for all $b \in \mathcal{U}$ there is a $\hat{b} \in \mathcal{U}$, such that $b - b^* = b^* - \hat{b}$. We will show that in this case $b^*$ solves (RecOpt). To this end, we first need the following lemma about the center of a symmetric location problem.

▶ **Lemma 6.** *Let $C \subseteq \mathbb{R}^n$ be a compact set of points that is symmetric with respect to $x^* \in \mathbb{R}^n$. Let $d$ be a distance measure that has been derived from a norm, i.e. $d(x, y) = \|y - x\|$ for some norm $\|\cdot\|$. Then $x^*$ is a $d$-center of $C$.*

**Proof.** Let $\max_{x \in C} d(x, x^*) = r$ and let $y_1, y_2 \in C$ be a pair of symmetric points (i.e. $y_1 - x^* = x^* - y_2$) that maximizes the distance to $x^*$. Let $x'$ be any point. Applying the triangle inequality and using that $y_1, x^*, y_2$ are collinear yields

$$2r = d(y_1, x^*) + d(x^*, y_2) = d(y_1, y_2) \leq d(y_1, x') + d(x', y_2)$$

and therefore either $r \leq d(y_1, x')$ or $r \leq d(x', y_2)$ holds. We conclude that

$$\max_{x \in C} d(x, x') \geq \max\{d(y_1, x'), d(y_2, x')\} \geq r,$$

hence $x'$ cannot be better than $x^*$. ◀

▶ **Theorem 7.** *Let $(P(b))$, $b \in \mathcal{U}$ be an uncertain linear program (42) and let $\mathcal{U}$ be symmetric with respect to $b^* \in \mathbb{R}^m$. Let $B$ be an optimal basis for $(P(b^*))$ and assume that $A_B^{-1}b \geq 0$ for all $b \in \mathcal{U}$. Then $x(b^*)$ solves (RecOpt).*

**Proof.** $B$ is an optimal basis for every $b \in \mathcal{U}$, as $A_B^{-1}b \geq 0$. Thus $x(b) = A_B^{-1}b$.
As $\mathcal{U}$ is a symmetric set with respect to $b^*$ and $x$ an affine linear mapping, the set of optimal solutions is symmetric with respect to $x(b^*)$ and we can apply Lemma 6. ◀

This directly gives a result for all interval-based uncertainty sets.

▶ **Corollary 8.** *Let $(P(b))$, $b \in \mathcal{U} = \{b \in \mathbb{R}^m : \underline{\eta} \leq b \leq \overline{\eta}\}$ be an uncertain linear program (42) and let $\underline{\eta}, \overline{\eta} \in \mathbb{R}^m$. Let $b \in \mathcal{U}$ and let $B$ be an optimal basis for $(P(b))$. If $A_B^{-1}\underline{\eta} \geq 0$ and $A_B^{-1}\overline{\eta} \geq 0$ both hold, then an optimal solution of (RecOpt) can be found by solving $(P(b^*))$ with $b^* := \frac{\underline{\eta}+\overline{\eta}}{2}$.*

Applied to the timetabling problem, we may conclude:

▶ **Corollary 9.** *Let a (TT) instance with uncertainty set $\mathcal{U}_1$, be given. Let $l^* := (1 + s/2)\hat{l}$ and assume that there is a basis that is optimal for $TT(\hat{l})$ and $TT((1 + s)l)$. Then any optimal solution to $TT(l^*)$ solves (RecOpt) for the timetabling problem for every distance $d$ that stems from a norm.*

## 4 Numerical Studies

### 4.1 Problem Instance and Parameters

The instance was created using the *LinTim* toolbox [12] for optimization in public transportation based on an intercity train network with the size of the German IC/ICE railway system. The time horizon under consideration consists of the eight-hour service period from 8 a.m. to 4 p.m., resulting in an EAN with 379 activities and 377 events. All computations were carried out on a Quad-Core AMD Opteron Processor running at 2.2 GHz using the C++ - interface of *Gurobi* v. 3.00.

We set for (R1-TT) $g_1 = 50$, $g_2 = 10,000$ to gain a solution which is a good compromise in robustness as well as in objective value. The budget $D$ for (R2-TT) was set to 2000. The budget $\delta$ for light robustness was set to 0.1, meaning that the objective value of the light robust solution is allowed to deviate up to 10 percent with respect to the nominal optimality. Furthermore, we tested a simple *uniformly buffered* solution by multiplying all node potentials of the nominal optimum with 1.06, which increased all activity durations by 6 percent, a method which is often applied in practice.

Concerning the choice of $S \subseteq \mathcal{U}$ for (RecOpt-TT), we tested two versions. In the first, we restricted the choice to extreme points of $\mathcal{U}$, in the second we chose uniformly over the whole uncertainty set. Our results showed a better performance of the latter approach regarding recovery costs to feasibility and optimality for $\mathcal{U}_1$, but a slightly better performance for the extreme points approach for $\mathcal{U}_2$. For the following evaluations we present the (RecOpt-TT) solutions under this respective scenario choice: For $\mathcal{U}_1$, the scenarios were chosen uniformly over the whole uncertainty set, for $\mathcal{U}_2$ only from the extreme points.

## 4.2 Setting

We tested the $\mathcal{U}_1$ algorithms for $s = 0, \ldots, 0.3$ and the algorithms for $\mathcal{U}_2$ with $s = 0, \ldots, 1$ and $k = 1$. For each algorithm and iteration the following values were measured:

- Objective value: $\sum_{(i,j)\in\mathcal{A}} w_{ij}(\pi_j - \pi_i)$
- Average relative buffer: $1/|\mathcal{A}|(\sum_{(i,j)\in\mathcal{A}}(\pi_j - \pi_i)/l_{ij}) - 1$
- Average costs, when recovering to feasibility: A large number of scenarios $l^q$, $q = 1, \ldots, Q$, (in that case $Q = 1,000$) chosen randomly from $\mathcal{U}_1$ was created, and for each of these scenarios the recovery costs were calculated by solving

$$\begin{aligned} \min \ & \sum_{i\in\mathcal{E}} \pi_i^q - \pi_i \\ \text{s.t.} \ & \pi_j^q - \pi_i^q \geq l_{ij}^q \quad \forall\, (i,j) \in \mathcal{A} \\ & \pi_i^q \geq \pi_i \quad \forall\, i \in \mathcal{E}.. \end{aligned}$$

  Afterwards, the average of these objective values was taken.
- Worst-case costs when recovering to optimality: As for the calculation of the recovery costs, scenarios $l^q$ for $q = 1, \ldots, Q$ were created. Then the respective timetable problem TT($l^q$) was solved and the $d_1$-distance to the given solution measured. The maximum of these distances is the optimality distance, an approximation to the $d_1$ radius.
- Feasibility: A large number of scenarios is chosen at random by an exponential distribution of average 0.1. We did not choose uniform distribution, as solutions easily tend to be infeasible and less insight is gained. For every scenario we tested if the robust solution is feasible or not and averaged the feasibility.
- Running times.

## 4.3 Evaluation

### 4.3.1 Objective value.

In Figure 1 the objective values of the robustness concepts for $\mathcal{U}_1$ and $\mathcal{U}_2$ are plotted against the control parameter $s$, describing the increasing uncertainty of the input data. The values of the nominal solution are as expected constant throughout $s$, just like the buffered and the light robust solution. The fastest growing costs are those of the strictly robust solution. They might be still acceptable for the small disturbances of $\mathcal{U}_1$, but they are clearly far too high for $\mathcal{U}_2$. The costs of the recovery robust solutions are moderate in both cases. Concerning the (RecOpt-TT) solutions, the costs grow moderately, though a bit faster than those of the recovery robust solution, on $\mathcal{U}_1$, while they stay extremely low for $\mathcal{U}_2$.

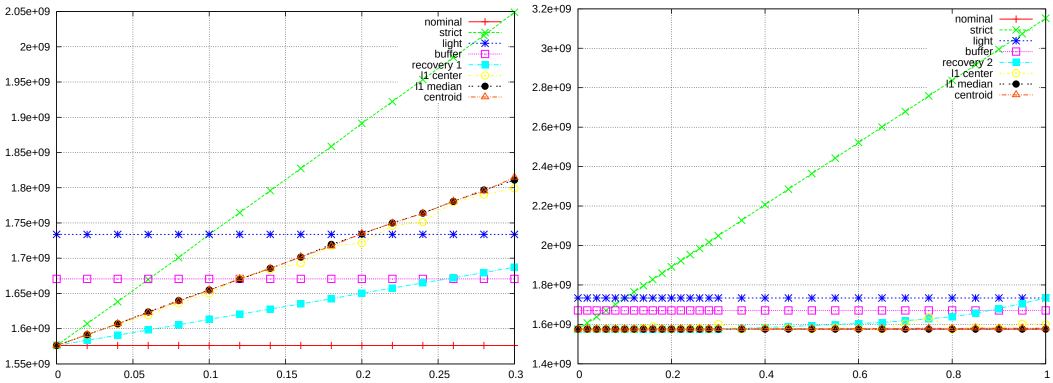


**Figure 1** Objective function for $\mathcal{U}_1$ (left) and $\mathcal{U}_2$ (right) solutions against $s$.

## 4.3.2 Average buffer.

The average buffers are shown in Figure 2. Most strikingly, the recovery robust solution for $\mathcal{U}_1$ has even larger buffers than the strictly robust solution, which is due to the fact that less weighted edges are buffered more. The light robust solution shows an interesting behavior by being not monotone. The centroid, $d_1$ center and median show a much larger increase in buffer times for $\mathcal{U}_1$ than for $\mathcal{U}_2$.

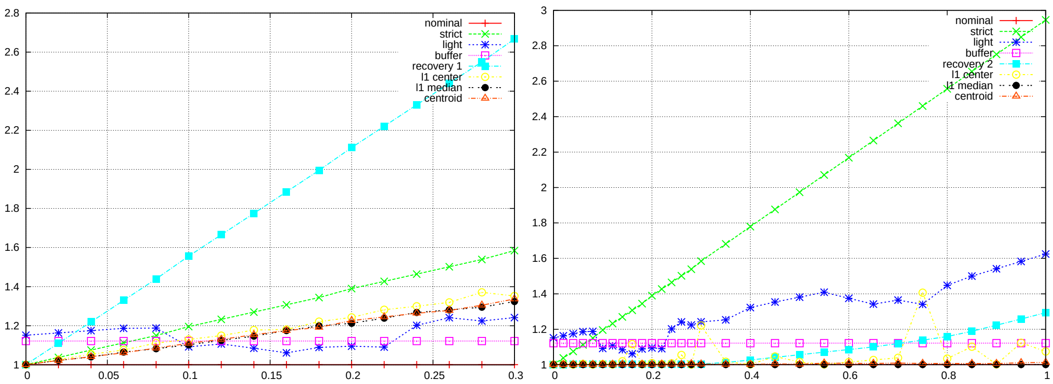


**Figure 2** Average buffer for $\mathcal{U}_1$ (left) and $\mathcal{U}_2$ (right) against $s$.

## 4.3.3 Average recovery costs when recovering to feasibility.

The recovery costs for $\mathcal{U}_1$ and $\mathcal{U}_2$ algorithms are depicted in Figure 3. Note the larger scale of the right figure: Recovery costs are generally much higher for $\mathcal{U}_2$-type uncertainties. The nominal solution performs worst for $\mathcal{U}_1$, being followed by the buffered solution with a constant offset stemming from the added 6 percent to activity durations. The recovery costs of the light robust solution stay slightly below those of the recovery robust solution, while the (RecOpt-TT) solutions show the slightest increase. On the other hand, they perform similar to the nominal solution for $\mathcal{U}_2$. Here the recovery robust solution has slightly lower costs, being exceeded by the buffered and especially the light solutions still.

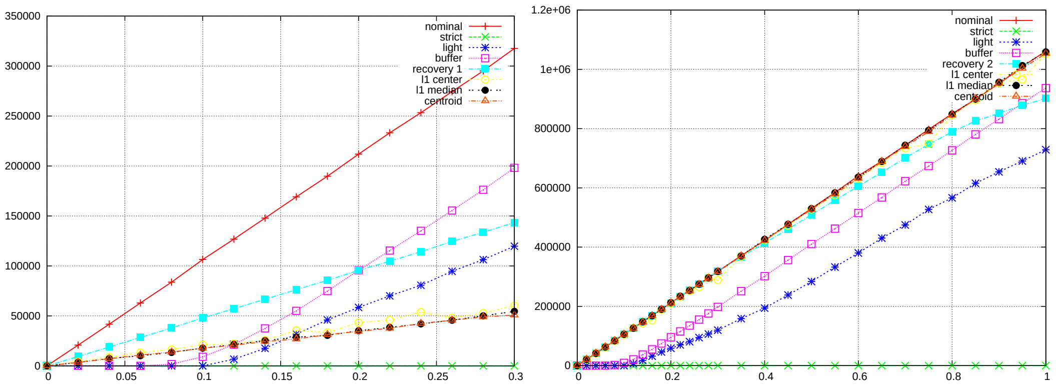


**Figure 3** Average recovery costs to feasibility for $\mathcal{U}_1$ (left) and $\mathcal{U}_2$ (right) against $s$.

### 4.3.4 Worst-case recovery costs when recovering to optimality.

Figure 4 shows the approximate maximum $d_1$-distances to the optimal solutions of the uncertainty set. The (RecOpt-TT) solutions perform very good in this category which shows that our heuristic approach (RecOpt-TT) can be used to minimize this distance. For $\mathcal{U}_1$ the solutions gained by (RecOpt-TT) clearly outperform the other robust solutions while they are comparable with some others for $\mathcal{U}_2$. Note that the strict robust solution performs poorly under this measure, as solutions are generally over-buffered.

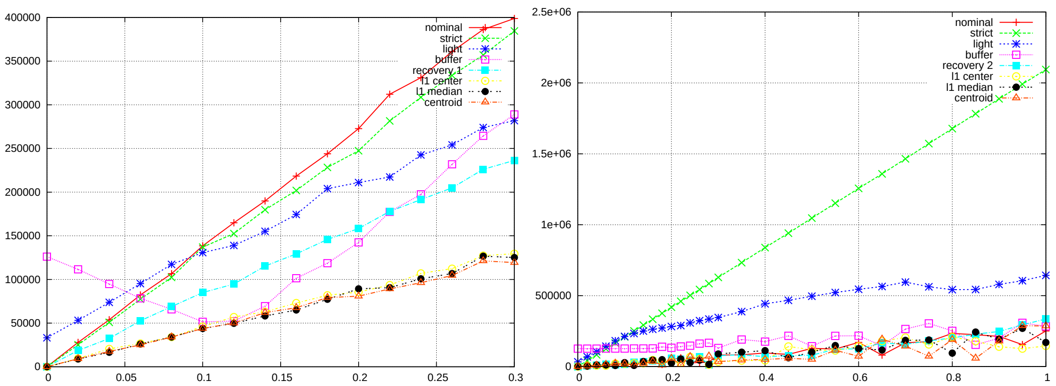


**Figure 4** Worst-case recovery costs to optimality for $\mathcal{U}_1$ (left) and $\mathcal{U}_2$ (right) against $s$.

### 4.3.5 Feasibility.

Figure 5 shows the average feasibility under exponential scenario distribution. Note that all solutions except of the strictly robust solution strongly decrease their feasibility for growing $s$ in $\mathcal{U}_1$. The light robust solution becomes infeasible as soon as its budget is completely used, which is exactly when its objective value equals the strictly robust solution (see Fig. 1).

Only the centroid keeps a small probability of feasibility throughout all values of $s$. For the $\mathcal{U}_2$ uncertainty, the situation changes completely. The buffered and the light robust solution keep moderate feasibility even for high values of $s$, while all other solutions (except of the strictly robust) stay low. This is exactly the intension of the recovery-robust approaches: They improve their nominal quality by allowing a repair phase and hence not aiming at feasibility for all scenarios.

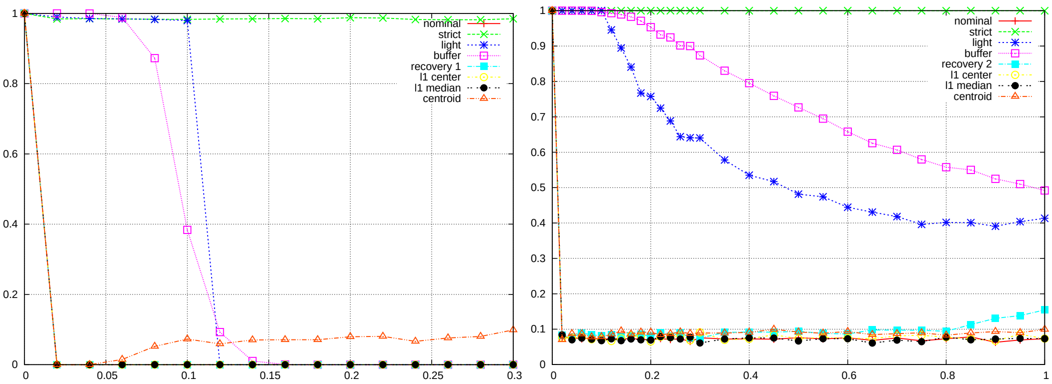


**Figure 5** Feasibility exponentially distributed for $\mathcal{U}_1$ (left) and $\mathcal{U}_2$ (right) against $s$, $\mu = 0.1$.

### 4.3.6 Running times.

Figure 6 shows the running times of the algorithms. Most time-consuming were the calculations of the $d_1$ center followed by the $d_1$ median and (R2-TT). The higher running times for the $d_1$-median and the centroid are due to the presolving phase in which the optimal solutions of all scenarios in $S$ needs to be calculated. Improving the running time of the $d_1$-center will be part of future research.

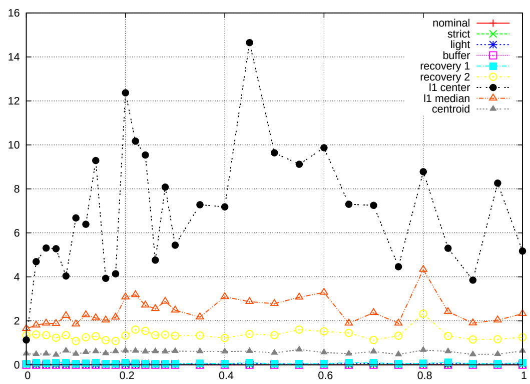


**Figure 6** Running times in seconds against $s$.

## 5 Conclusion

We applied the most prominent robustness to timetabling and compared them on a real-world instance. Furthermore, we introduced a new approach, minimizing the recovery distances to a subset of scenarios, that is easily applicable to any robustness problem, whenever a method for solving the original problem is at hand. We have shown that there are significant differences in the performance of the concepts depending on the type of uncertainty under consideration. Strict robustness, as an example, is a considerable concept for $\mathcal{U}_1$ uncertainty, but not an option for $\mathcal{U}_2$. Concerning the (RecOpt-TT) solutions, especially the centroid approach gives good feasibility and recovery properties with average costs on $\mathcal{U}_1$, while the same approach for $U_2$ sticks too closely to the nominal solution for having good robustness properties. We conclude that it is crucial to choose the robustness concept to be applied to the specific problem structure and the uncertainty set. Future research will include investigating improved ways of choosing the scenario subset $S \subseteq \mathcal{U}$ and theoretical results on the quality of the gained solution, as well as applications to PESP models with applications to *periodic* timetabling.

## References


1 A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton University Press, Princeton and Oxford, 2009.

2 A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

3 D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

4 S. Cicerone, G. D'Angelo, G. Di Stefano, D. Frigioni, and A. Navarra. Recoverable robust timetabling: Complexity results and algorithms. Technical Report ARRIVAL-TR-0172, ARRIVAL project, 2008.

5 S. Cicerone, G. D'Angelo, G. Di Stefano, D. Frigioni, and A. Navarra. Recoverable Robustness for Train Shunting Problems. *Algorithmic Operations Research*, 2009.

6 S. Cicerone, G. Di Stefano, M. Schachtebeck, and A. Schöbel. Dynamic algorithms for recoverable robustness problems. In Matteo Fischetti and Peter Widmayer, editors, *ATMOS 2008 - 8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems*, Dagstuhl Seminar proceedings, 2008.

7 G. D'Angelo, G. Di Stefano, and A. Navarra. Recoverable robust timetables on trees. Technical Report ARRIVAL-TR-0163, ARRIVAL project, 2008.

8 G. D'Angelo, G. Di Stefano, A. Navarra, and C. M. Pinotti. Recoverable robust timetables: An algorithmic approach on trees. *IEEE Transactions on Computers*, 2010. to appear.

9 M. Fischetti and M. Monaci. Light robustness. In R. K. Ahuja, R.H. Möhring, and C.D. Zaroliagis, editors, *Robust and online large-scale optimization*, volume 5868 of *Lecture Note on Computer Science*, pages 61–84. Springer, 2009.

10 P. Kouvelis and G. Yu. *Robust Discrete Optimization and Its Applications*. Kluwer Academic Publishers, 1997.

11 C. Liebchen, M. Lüebbecke, R. H. Möhring, and S. Stiller. The concept of recoverable robustness, linear programming recovery, and railway applications. In R. K. Ahuja, R.H. Möhring, and C.D. Zaroliagis, editors, *Robust and online large-scale optimization*, volume 5868 of *Lecture Note on Computer Science*. Springer, 2009.

12 M. Schachtebeck and A. Schöbel. Lintim – a toolbox for the experimental evaluation of the interaction of different planning stages in public transportation. Technical report, ARRIVAL Report 206, 2009.

13 A.L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21:1154–1157, 1973.

14 H. Yaman, O.E. Karasan, and M.C. Pinar. The robust spanning tree problem with interval data. *Operations Research Letters*, 29:31–40, 2001.