Evaluation of Robust Timetables

Sebastian Albert

2012-03-05

Bachelor's thesis

Contents

1 Introduction

Throughout the past decades, delay resistant timetabling has evolved as an interesting application of mathematical models in Operations Research. Especially scheduling departures and arrivals in public transportation with suitable time buffers has gained attention due to annoying delays in the everyday operation of railway companies. Meanwhile, in Operations Research in general, many different robust optimization techniques have been proposed for problems containing uncertainties, some of which are applicable to timetabling problems as well. This work focuses on the evaluation of a few selected robust optimization approaches to aperiodic timetabling and their performance in the event of different random delay scenarios on a real-world instance.

The essential question from the timetabling point of view is how to place buffers between departures and arrivals in order to maximize the delay absorption capabilities of the timetable while keeping nominal travel times as small as possible in some sense. These contradicting objectives demand a trade-off between a timetable's delay-resistance and tightness. Generally speaking, increasing the total amount of buffering increases delay resistance and travel times simultaneously. However, different buffer placing techniques yield different behaviours of these increases and are to be compared.

Some of the results may be applicable to general scheduling problems as well, while others may not due to assumptions motivated by the public transportation application. Generalizability is not a target of this work. Some results may even owe their charecteristics to structural properties of the real-world example used for the numerical studies. Therefore, general conclusions shall only be drawn cautiously.

The remainder of this work is structured as simple as this: One section introduces the aperiodic timetabling problem, another section introduces robustness concepts and already applies them to aperiodic timetabling, and a third section studies numerical results on a real-world instance.

Similar work has been done for example in [Hö11] and [GS10].

2 The aperiodic timetabling problem

This section introduces the nominal aperiodic timetabling problem in public transportation, which is then confronted with uncertainty throughout the next sections. The aperiodic timetabling problem aims at finding departure and arrival times for vehicles like trains or buses that are to be published and thus binding once they are laid down.

All formulations in this section follow the general concept of [Sch06], although not always literally.

2.1 Stops, lines, events and activities

The properties of the infrastructure on which a timetable is to be calculated can be represented mathematically in Graphs.

Definition 2.1. A *Public Transportation Network* or *PTN* is a directed or undirected graph $G = (S, C)$ with the nodes (vertices) in S representing locations like bus stops or train stations, and the edges in $C \subset S \times S$ representing direct connections between them like roads or tracks. If G is undirected, these connections are meant to be usable in both directions.

Given this basic infrastructure, one can speak of lines:

Definition 2.2. A *path* in a graph $G = (S, C)$ is a finite sequence of consecutive edges $c_1 = (s_1, s_2), c_2 = (s_2, s_3), ..., c_{n-1} = (s_{n-1}, s_n) \in C$. $n \in \mathbb{N}$ is called the *length* of the path. Simplifying notations, a node $s \in S$ is said to be in a path P, i. e., $s \in P$, if there is some $s' \in S$ such that $(s, s') \in P$ or $(s', s) \in P$, and whenever all the edges of a path are uniquely determined by the nodes they connect, the path may be written as a sequence of nodes $(s_1, s_2, ..., s_n)$ as well. A path in the PTN is also called a *line*.

From the notion of a line on a PTN there is only one more step necessary to get an aperiodic timetabling instance: All lines that shall be run on the PTN shall be rolled out into the so-called Event-Activity-Network, a commonly used modelling structure for scheduling problems. Which lines are to be run on the network is the result of the lineplanning problem, which is considered to be solved before the timetabling step.[1]

Definition 2.3. Let $L_1, ..., L_m, m \in \mathbb{N}$ be lines on a PTN $G = (S, C)$. An *Event-Activity-Network* or *EAN* is a directed graph $N = (E, A)$, where the nodes in $E = E^{\text{dep}} \sqcup E^{\text{arr}}$ are *departure* and *arrival events*, respectively, each belonging to a stop from the PTN and a line:

$$E^{\text{dep}} = \{(s, L, \text{dep}) \in S \times \{L_1, ..., L_m\} \times \{\text{dep}\} : \exists s' \in S : (s, s') \in L\}$$

$$E^{\text{arr}} = \{(s, L, \text{arr}) \in S \times \{L_1, ..., L_m\} \times \{\text{arr}\} : \exists s' \in S : (s', s) \in L\}$$

The set of arcs (directed edges) $A = A^{\text{drive}} \sqcup A^{\text{wait}} \sqcup A^{\text{change}} \sqcup A^{\text{head}}$ consists of so-called *activities*: The *driving*, *waiting*, *changing* and *headway activities*, respectively.

$$A^{\text{drive}} = \big\{((s_1, L, \text{dep}), (s_2, L, \text{arr})) \in E^{\text{dep}} \times E^{\text{arr}} : (s_1, s_2) \in L\big\}$$

$$A^{\text{wait}} = \big\{((s, L, \text{arr}), (s, L, \text{dep})) \in E^{\text{arr}} \times E^{\text{dep}}\big\}$$

$$A^{\text{change}} \subseteq \big\{((s, L, \text{arr}), (s, L', \text{dep})) \in E^{\text{arr}} \times E^{\text{dep}} : L \neq L'\big\}$$

$$A^{\text{head}} \subseteq \big\{((s, L, \text{dep}), (s, L', \text{dep})) \in E^{\text{dep}} \times E^{\text{dep}} : L \neq L'\big\}$$

Driving and waiting activities refer to the operation of a line, while changing activities represent customers who must change from one vehicle to another in one station, and headway activities model security distances between the departures of different vehicles from the same station.
Let furthermore $l : A \to \mathbb{N}_0$ assign *lower bounds* to all activities and let $w : A \to \mathbb{Q}_0^+$ be a *weight* function. As A is finite, l and w can also be written as vectors with indices in A.
Whenever an EAN is referred to as a tuple (E, A, l, w), E and A are implicitly meant to be partitioned as above.

Note that the lower bounds can be interpreted as minimum durations,

[1] Although integrating lineplanning and timetabling into one optimization problem seems reasonable, this is generally not done due to the high computational complexity it would result in.

and weights could for example be expected passenger loads on the respective activities. Driving and waiting activities are derived directly from the trips on the PTN. Minimum durations of driving activities may depend on the vehicle and have thus not already been modelled in the PTN. Passenger amounts can be estimated and necessary changing activities determined by finding shortest paths in the PTN whenever so-called origin-destination information is available, i. e., how many customers want to travel between all pairs of locations on average. All the modelling decisions in this chapter were inferred from the framework used for the numerical studies.

2.2 A fundamental difference: periodic and aperiodic timetabling

After considering public transportation lines and paths of passengers, the next focus is on scheduling times for departure and arrival events corresponding to the EAN: the *timetabling* problem. A major distinction is to be made here between so-called *periodic* and *aperiodic* timetables. The former implements the idea of public transportation lines being operated on a regular basis with some periodicity T, which means that every event is repeated after a certain amount of time specified by T, e. g. $T = 60$min. This periodic approach is motivated by the ease, especially for customers, to remember the schedule, and by the smaller problem size, as only one instance of every train has to be scheduled. The *Periodic Event Scheduling Problem* (PESP) was introduced by Serafini and Ukovich in [SU89] and is subject to research to date. Unfortunately it turned out to be NP-hard and is therefore computationally intractable. Much effort has been put into finding good approximations.

The alternative is to drop periodicity. Since each line should nevertheless be operated several times a day, the problem size formally increases as all operations of each line have to be scheduled. However, the aperiodic timetabling problem is still easier to solve for it can be formulated as a standard linear program. Note that this is only the case whenever precedence decisions are made beforehands and fixed. If the timetabling instance had to decide on the order of, for example, two trains' departures from the same station, it becomes a mixed integer problem that might be essentially harder to solve.

Definition 2.4. An *aperiodic EAN* is an EAN that contains no directed cycles.

In the periodic case, directed cycles are not a contradiction since times are considered $\mod T$. A (periodic) EAN can be rolled out into an aperiodic EAN by periodically replicating all the lines over a certain span of time by dropping some incoming arcs in the first instances of the lines and outgoing arcs from the last instances and changing some arcs' targets to the next periodical instance of the respective line.

To this end, a periodic timetable has been employed and rolled out over one day of operation for the numerical studies on aperiodic timetables in section 4. This provides the additional advantage that the resulting timetables are still close-to-periodic regarding the order of line operations: Having all or most instances of one line run right at the beginning of the day would not yield a realistic solution.

2.3 Feasible and optimal aperiodic timetables

Definition 2.5. An *aperiodic timetable* for a given EAN (E, A, l, w) is a mapping $\pi : E \to \mathbb{R}$ that assigns a time to each event. Due to the finiteness, π can also be written as a vector in $\mathbb{R}^{|E|}$ with indices in E.
A timetable π for an EAN is *feasible* if for every activity $a = (i, j) \in A$ it satisfies $\pi_j - \pi_i \geq l_a$.

There are no upper bounds to the durations of activities throughout this entire work. This can be justified by the observation that upper bounds do not arise as directly and necessarily from the application as lower bounds do. Upper bounds would, in the case of delays, which will be the main focus here, most probably lead to infeasibility very soon, too. It is not unusual anyway to neglect upper bounds when treating delays or disruptions.

Lemma 2.6. *A feasible aperiodic timetable exists for a given EAN (E, A, l, w) if and only if it does not contain any directed cycle with a positive sum of minimum durations.*

Proof. Let $C = a_1, ..., a_n = (e_1, e_2), (e_2, e_3), ..., (e_{n-1}, e_n), (e_n, e_1), n \in \mathbb{N}$ be a directed cycle in (E, A) with $\sum_{i=1}^{n} l(a_i) > 0$. Then there is some integer j, $1 \leq j \leq n$, with $l(a_j) > 0$. A feasible timetable π would satisfy $\pi_i \leq \pi_{i+1 \mod n}$ for all integers i with $1 \leq i \leq n$ (since $l(a_i) \geq 0$), and $\pi_j < \pi_{j+1 \mod n}$. By transitivity follows $\pi_1 \leq \pi_2 \leq ... \leq \pi_j < \pi_{j+1} \leq ... \leq \pi_n \leq \pi_1$, yielding the contradiction $\pi_1 < \pi_1$.
Let now (E, A, l, w) be free of any directed cycle with a positive sum of weights. Then a feasible aperiodic timetable can be constructed by the following steps:

1. Perform a search for cycles. All cycles only constist of arcs with length 0. Since every arc represents a $\leq$-condition, all nodes among the cycle must be assigned the same time. Therefore, identify them by substituting it by a single node that gets all the incomig and outgoing arcs of all nodes in the cycle. No new cycles will appear because every imaginable new cycle would have been a cycle before.

2. The graph is now cycle-free. Nodes can thus be traversed in a topological ordering. Initially, set all times to 0.

3. For each node e (in topological order):

   - Fixate the time π_e of e. Due to the order of traversal, any incoming arc of e has been looked at.

   - For all outgoing arcs $a = (e, o)$, set the (not yet fixated) time of o to $\pi_e + l_a$ if this is later than the time of o as has been set up to now.

4. Re-substitute the cycle representation nodes, copying their times to all of the nodes in the respective cycle.

By construction within step 3, all feasibility constraints are met. $\square$

The algorithm sketched in this proof is also known as the *Critical Path Method*, see for example [KJ61]. It will be used as a delay-management technique in the numerical experiments in section 4 as well.

If upper bounds were imposed on the activities, finding a feasible aperiodic timetable would turn into the feasible differential problem as described for example in [Roc84].

So, feasibility is easy. Now what about optimality?

Definition 2.7. A feasible aperiodic timetable π^* for an EAN (E, A, l, w) is *optimal* if for every feasible aperiodic timetable π for this EAN the following optimality condition is fulfilled:

$$\sum_{a=(i,j)\in A} w_a(\pi_j^* - \pi_i^*) \leq \sum_{a=(i,j)\in A} w_a(\pi_j - \pi_i)$$

This objective gives meaning to the activity weights in the sense that the larger the weight, the more important it is to keep the duration of the respective activity small. With the weights being amounts of passengers using the activity on their way, optimality means minimal overall travelling time (in total and average).

Since the objective function as well as the constraints are linear, the *aperiodic timetabling problem* given an EAN with lower bounds and weights for activities (E, A, l, w) can be posed as a linear program:

$$\begin{array}{lrl} (\text{TT}) \ \min\limits_{\pi} & \tilde{w}^T\pi & \\ \quad\text{s. t.} & \tilde{A}\pi & \geq l \\ & \pi & \geq 0 \end{array}$$

$\tilde{A}$ denotes the incidence matrix, i. e.,

$$\tilde{A} = (\tilde{a}_{a,e})_{a\in A, e\in E} \text{ with } \tilde{a}_{a,e} = \begin{cases} -1, & a = (e, e'), e' \in E \\ 1, & a = (e', e), e' \in E \\ 0, & \text{otherwise} \end{cases}$$

and the coefficient vector $\tilde{w}$ is obtained by expanding and resorting the objective defined above, resulting in a weight vector for events containing the net amount of passengers reaching their destination minus passengers starting their trip with the respective event (because those who are neither beginning nor ending their trip add to 0 as they all have each one incoming and one outgoing activity on this event):

$$\tilde{w}_e = \left(\sum_{a=(i,e)\in A} w_a - \sum_{a=(e,j)\in A} w_a \right) \forall e \in E$$

Lemma 2.8. *The aperiodic timetabling problem (TT) yields indeed an optimal aperiodic timetable for an EAN (E, A, l, w) if it is feasible, i. e. there is at least one feasible aperiodic timetable.*

Proof. By definition of the incidence matrix $\tilde{A}$, the constraints in (TT) are obviously exactly the coniditions for an aperiodic timetable to be feasible. So it remains to show that the objective in (TT) equals the optimality criterion.

$$\begin{aligned}
\tilde{w}^T \pi &= \sum_{e\in E} \tilde{w}_e \pi_e \\
&= \sum_{e\in E} \left(\sum_{a=(i,e)\in A} w_a - \sum_{a=(e,j)\in A} w_a \right) \pi_e \\
&= \sum_{e\in E} \sum_{a=(i,e)\in A} w_a \pi_e - \sum_{e\in E} \sum_{a=(e,j)\in A} w_a \pi_e \\
&= \sum_{a=(i,e)\in A} w_a \pi_e - \sum_{a=(e,j)\in A} w_a \pi_e \\
&= \sum_{a=(i,j)\in A} w_a \pi_j - \sum_{a=(i,j)\in A} w_a \pi_i \\
&= \sum_{a=(i,j)\in A} w_a (\pi_j - \pi_i)
\end{aligned}$$

□

3 Robust Optimization

There is a gap between theory and praxis in most, if not all sciences, and operations research is not constituting an exception. Every optimization problem depends on parameters, like coefficients in a linear program or any other input data in any type of optimization one can think of. If such an OR method is to be applied to a real-world context, uncertainty is very likely to appear, for example when input data cannot be measured exactly or has to be estimated, or unforeseen disturbances may occur, possibly rendering the computed solution suboptimal or even infeasible.[2]

One famous attempt to bridge this gap is treating uncertainty by means of so-called *stochastic programming*.[3] These approaches rely on probability distributions and usually penalize possible a-posteriori constraint violation by the notion of recourse. Problem formulations tend to grow exponentially when continuous distributions are approximated by discrete sampling. These are major disadvantages in many cases and lead to the development of deterministic approaches known as *robust optimization*.

The following framework used throughout this entire section is based on [Sch09b].

Definition 3.1. Given a general optimization problem

$$\begin{aligned} (\mathrm{P}) \ \min_x \quad & f(x) \\ \text{s. t.} \quad & F(x) \geq 0 \end{aligned}$$

with any objective function f and an arbitrary condition function F, the corresponding *uncertain problem* is

$$\begin{aligned} (\mathrm{P}(\xi)) \ \min_x \quad & f(x, \xi) \\ \text{s. t.} \quad & F(x, \xi) \geq 0 \end{aligned}$$

[2]Optima oftentimes lie on the boundary of a set of feasible solutions, like in the simplex algorithm in linear programming, where even the slightest deviation in an active constraint can easily cause infeasibility. For more such considerations, see for example the motivation section in [BTN98].

[3]For an introduction on SP, [KW94] and [Pre95] are renowned textbook references.

with f and F being generalizations of the objective and constraint functions, taking into account the uncertain input data ξ, called the *scenario*, which is generally said to be out of an *uncertainty set* U. Usually there is some $\hat{\xi} \in U$ such that $f(\cdot) = f(\cdot, \hat{\xi})$ and $F(\cdot) = F(\cdot, \hat{\xi})$, called the *nominal scenario*. $\mathrm{P}(\hat{\xi}) = \mathrm{P}$ is called the *nominal problem*.

This generic formulation adopts for instance to general linear problems

$$\begin{aligned} (\mathrm{LP}) \ \min_{x} \quad & c^T x \\ \text{s. t.} \quad & Ax \leq b \end{aligned}$$

with uncertainty possibly lying in the coefficient matrix A, the right-hand side vector b, and the objective coefficients' vector c by choosing

$$f(x, \xi) = c^T x \text{ and } F(x, \xi) = b - Ax \text{ for } \xi = (A, b, c).$$

Plugging in the timetabling problem introduced in the previous section gives

$$f(\pi, \xi) = \tilde{w}^T \pi \text{ and } F(\pi, \xi) = \tilde{A}\pi - l \text{ for } \xi = l,$$

with $\tilde{w}$ and $\tilde{A}$ defined as above and not subject to uncertainty, as described next.

3.1 Scenarios and uncertainty sets in timetabling

In the timetabling application, uncertainty lies within the possibility of unforeseen delays occuring during operation.[4] To be more precise, the lower bounds that describe minimal durations of activities can increase, thus rendering the original timetable infeasible. Techniques of robust optimization that take the possible delay scenarios into account need the uncertainty set specified.

The definition of possible delay scenarios is essential for the resulting robustness. Whenever at operation time scenarios emerge that have not been

[4]Other uncertainties are possible, for example the weights in terms of travelling passengers may be uncertain. This is not considered here.

taken into account in the optimization stage, high repair cost or infeasibility may be incurred which the optimization method cannot be blamed for. However, identifying an uncertainty set covering most delay scenarios while not being too loose is highly non-trivial, even though robust optimization does not consider probabilities, which also have not been found yet for delays in public transportation to a satisfying degree.

Nevertheless, in order to conduct at least some investigation on robust techniques for aperiodic timetabling, two types of uncertainty sets are considered here. U_1 depends on a non-negative parameter s determining the possible increase of minimal durations for all driving activities. U_2 is a subset of U_1, depending on a second parameter k indicating how many of the driving activities can at most be delayed at all. The following definition puts this more formal:

Definition 3.2. Let (E, A, l, w) be a given EAN with lower bounds and weights on activities. For arbitrary $s \geq 0$ and $k \in \mathbb{N}_0$, the following sets of possible lower bounds with delays are the *uncertainty sets* considered throughout the remaining part of this work:

$$U_1(s) = \left\{ \tilde{l} \in \mathbb{R}^{|A|} : \tilde{l}_a \in [l_a, (1+s)l_a] \, \forall a \in A^{\text{drive}}, \tilde{l}_a = l_a \forall a \in A \setminus A^{\text{drive}} \right\}$$

$$U_2(s, k) = \left\{ \tilde{l} \in U_1(s) : |\{a \in A : \tilde{l}_a > l_a\}| \leq k \right\}$$

Note that U_1 is convex, while U_2 is generally not. $U_2(s, k) \subseteq U_1(s)$, and $U_2(s, k) = U_1(s)$ if $k \geq |A^{\text{drive}}|$. However, k is assumed to be rather small.

3.2 Strict robustness

The most basic idea of robust optimization is strict robustness: Solutions are required to be feasible for *all* possible scenarios in the given uncertainty set.

Definition 3.3. Given an uncertain optimization problem

$$\begin{aligned} (\mathrm{P}(\xi)) \quad \min_{x} \quad & f(x, \xi) \\ \text{s. t.} \quad & F(x, \xi) \geq 0 \end{aligned}$$

with any objective function f and an arbitrary constraint function F, its *strict robust counterpart* with regard to an uncertainty set U is the problem

$$\begin{aligned} \text{(SR)} \ \min_{x} \quad & \sup_{\xi \in U} f(x, \xi) \\ \text{s. t.} \quad & F(x, \xi) \geq 0 \ \forall \xi \in U. \end{aligned}$$

This notion goes back to the pioneering work of A. Ben-Tal and A. Nemirovski, who took the idea of strict robust feasibility from the field of (robust) control theory to mathematical programming in [BTN00]. They credit the first results of robust optimization, as widely accepted today, to [Soy73], but laid the foundation for modern robust optimization themselves by the generelization from *column-wise uncertainty* to arbitrary uncertainty sets, focusing on finite intersections of ellipsoids in order to ensure computational tractability by avoiding semi-infiniteness of the constraints. However, in the aperiodic timetabling application considered here, all of the constraints in the strict robust counterpart are dominated by finitely many scenarios in which the delays take on their individual worst values, so their main result is not needed here. For both $U_1(s)$ and $U_2(s, k)$ the robust counterpart to the aperiodic timetabling problem reads

$$\begin{aligned} \text{(TT-SR)} \ \min_{\pi} \quad & \tilde{w}^T \pi \\ \text{s. t.} \quad & \tilde{A}\pi \geq (1+s)l \\ & \pi \geq 0 \end{aligned}$$

because for every activity $a \in A$ in the underlying EAN (E, A, l, w) there is one scenario $\tilde{l} \in U_2(s, 1) \subset U_2(s, k) \subset U_1(s)$ having just the lower bound $\tilde{l}_a = (1 + s)l_a$ delayed as badly as possible and thus dominating all other scenarios with regard to a particular activitiy a. As there are only finitely many $a \in A$, the strict robust counterpart can be written as this linear program. Obviously, this is nothing but the nominal problem again simply with all lower bounds increased to the worst. This retains the good tractability of the aperiodic timetabling without even increasing the problem size but simultaneously leads to very conservative solutions of bad nominal quality,

i. e. strongly elongated travelling times for all customers. Especially in the case of $U_2(s,k)$ with small values for k, the increase is not reasonable at all. However, many transportation companies employ such overall buffering as a rule of thumb, for instance by adding 7% to all minimal driving durations as recommended by the International Union of Railways according to [LS06].

Since aperiodic timetabling is not the only problem yielding inacceptably bad solutions under strict robustness, several other robustness terms have been coined in the literature that try to achieve a better trade-off between nominal quality and robustness by relaxing the strict robustness constraints.

3.3 Light robustness

In [FM09], M. Fischetti and M. Monaci modify a robustness model introduced by D. Bertsimas and M. Sim towards a trade-off between robustness and nominal quality, enabling the user to control the nominal quality by a constraint and then seeking for the solution which, while obeying this nominal quality condition, violates the constraints of strict robustness as little as possible in some sense. To this end, they introduce a kind of slack variables measuring the relaxation of the robustness constraints, which they then aim to minimize. As this is somehow comparable to the recourse in two-stage stochastic programming, they consider their approach, that they call *Light Robustness*, as an intermediate method between robust optimization and stochastic programming. However, as no probabilities are involved, this is still rather to be considered a robust optimization technique.

The basic idea can be put into the robustness terms developed above as follows, although this deviates a little from their original formulation:

Definition 3.4. Given a robust optimization problem with objective function f and constraint function F, both depending on the scenario out of some specified uncertainty set U, the *light robust counterpart* is

$$
\begin{array}{llllll}
(\mathrm{LR}(\delta)) & \min\limits_{\gamma} & & \bar{w}^T\gamma & & \\
& \text{s. t.} & F(x,\hat{\xi}) & & \geq 0 & \\
& & f(x,\hat{\xi}) & & \leq (1+\delta)\bar{z} & \\
& & F(x,\xi)+ & \gamma & \geq 0 & \forall \xi \in U \\
& & & \gamma & \geq 0 &
\end{array}
$$

$\hat{\xi}$ denotes the nominal scenario and $\bar{z}$ its optimal value, $\delta \geq 0$ is a scalar parameter and $\bar{w}$ a weight vector.

A light robust solution must thus still satisfy strict feasibility for the nominal scenario and ensure that its quality is not more than a factor of δ apart from the optimum.

It is important to mention that the formulation of the original constraints is affecting the outcome of this model, and different but equivalent formulations can yield very different light robust solutions. This is due to the dependence on the behaviour of the constraint function F beyond feasibility, i. e. how its values behave in the infeasible case $F(x) \not\geq 0$. Values close to 0 should represent proximity to feasibility, independently in each component of F. But F could be modified by applying an arbitrary function u only respecting $u(x) \geq 0 \Leftrightarrow x \geq 0$ without changing the nominal problem. A meaningful requirement on F would be, for instance, affine linearity, as fulfilled in the case of linear programming, in which the light robust counterpart is again a linear program and reads as follows:

$$
\begin{array}{llllll}
(\text{LP-LR}(\delta)) & \min\limits_{\gamma} & & \bar{w}^T & \gamma & \\
& \text{s. t.} & \hat{b} - \hat{A}x & & \geq 0 & \\
& & c^T x & & \leq (1+\delta)\bar{z} & \\
& & b - Ax+ & \gamma & \geq 0 & \forall (A,b) = \xi \in U \\
& & & \gamma & \geq 0 &
\end{array}
$$

Still there are different equivalent formulations of an LP possible, so it must be somehow normalized. A further possibility to modify the meaning of robustness is the vector $\bar{w}$ weighting the constraints. However, the aperiodic timetabling problem is already normalized with the coefficients only being -1, 0 and 1. Since it is hard to tell how much impact the non-buffering of a possible delay on which activity would have, setting $\bar{w} = 1$ is not a bad idea. Nevertheless, setting the weights corresponding to the passenger loads on each respective activity as a first heuristic will be tried out in the numerical studies, too.

Due to the same linear constraint domination argument as in the previous section on strict robustness, the light robust counterpart of the aperiodic timetabling problem is computationally tractable with finitely many constraints, and again the assertion of U_2 that only a limited number of delays occur is not relaxing anything. By plugging in the incidence matrix $\tilde{A}$ and the worst delays as defined by parameter s, it reads for both $U_1(s)$ and $U_2(s,k)$:

$$
\begin{aligned}
(\text{TT-LR}(\delta)) \quad \min_{\gamma} \quad & \bar{w}^T\gamma \\
\text{s. t.} \quad \tilde{A}\pi &\geq l \\
\tilde{w}^T\pi &\leq (1+\delta)\bar{z} \\
\tilde{A}\pi &\geq (1+s)l - \gamma \\
\gamma,\pi &\geq 0
\end{aligned}
$$

3.4 Adjustable robustness

Another attempt to address the conservatism of strict robustness has been proposed by Ben-Tal and Nemirovski themselves together with A. Goryashko and E. Guslitzer. In [BTGGN04] they introduce the idea of separating the optimization variables into two disjoint sets, which may be possible in many applications: *non-adjustable* variables, whose values have to be fixed in the planning stage (also referred to as *here and now*), like all of the variables in strict robustness, and *adjustable* variables, whose values can be adjusted after the concrete scenario has become certain (also referred to as *wait and see*). They call this approach *Adjustable Robustness* and apply it to uncertain linear programs.

Definition 3.5. Given an uncertain linear program

$$
\begin{aligned}
(\mathrm{LP}(\xi)) \ \min_{p} \quad & c^T p \\
\text{s. t.} \quad & Pp + Qq \ \leq b
\end{aligned}
$$

with p and q being the optimization variables, the former considered *non-adjustable* and the latter considered *adjustable*, and $(P, Q, b) = \xi \in U$ uncertain, the *adjustable robust counterpart* is

$$
\begin{aligned}
(\text{LP-AR}) \ \min_{p} \quad & c^T p \\
\text{s. t.} \quad & \forall \xi = (P, Q, b) \in U \exists q : Pp + Qq \leq b
\end{aligned}
$$

Considering aperiodic timetabling with the uncertainty sets $U_1(s)$ and $U_2(s, k)$ as defined above, only the right-hand side is uncertain.[5] A closer look on $U_1(s)$ reveals that, no matter how the adjustable ones among the timetabling variables are chosen, the adjustable robust counterpart will be the same as the strict robust counterpart.[6] This is the basic motivation for the choice of $U_2(s, k)$ with k small. If some departure and arrival times do not have to be fixed in advance and are thus regarded somehow less important, consecutive activities between them can share the worst case such that not every single one has to get its worst-case buffer.[7]

Now, how shall the adjustable variables be chosen among all the events that are to be scheduled? The choice has to be made carefully, because adjustable robust optimization is even for linear programming generally NP-hard, as proven in [Gus02]. In order not to have to make complicated considerations about possible propagations of delays in the EAN, the adjustable events are chosen here to be only those that are no crossing-point in the

[5]The decomposition of the optimization variables into two vectors reminds of *two-stage stochastic programming with recourse*, and in these terms, the problem has *fixed recourse* since Q is not uncertain.

[6]This is also proven by Ben-Tal et al. in their introducing article [BTGGN04] as *Theorem 2.1.* $U_1(s)$ fulfills the condition of constraint-wise uncertainty, being the cartesian product of closed intervals, that are compact and convex.

[7]This breaks the condition of the aforementioned theorem by abolishing constraint-wise uncertainty.

network. This is now put into notions from graph theory:

Definition 3.6. On a directed graph (E, A), let

$$\delta^+(e) = \{(e', e) \in A : e' \in E\} \text{ and}$$

$$\delta^-(e) = \{(e, e') \in A : e' \in E\}$$

be the *incoming* and *outgoing* arcs, respectively, of node $e \in E$, with

$$d^+(e) = \left|\delta^+(e)\right| \text{ and}$$

$$d^-(e) = \left|\delta^-(e)\right|$$

denoting the number of such arcs.

Using these common shorthands, the basic idea of a not-too-complicated choice of adjustable events is the following.

Definition 3.7. Given a directed graph (E, A), a node $e \in E$ is called a *trivial node* if $d^+(e) = d^-(e) = 1$. A path with all but the first and the last node being trivial is called a *trivial path*.

Not hard to guess, the trivial nodes on an EAN (E, A, l, w) shall now be adjustable, such that all activities among a trivial path can share a buffer facing the worst case of delays. This actually coincides with the understanding of non-adjustable events being more important, as stops where passengers can change are more important to obey a schedule for customers not to miss their connection, and are anyway oftentimes larger cities with importance for more passengers than simple stop-by stations without possibility to change over.

Note that it may be necessary to drop headway constraints because otherwise they would make almost *all* departure events non-trivial. This is investigated on the real-world instance in the numerical studies.

A more general formulation of the concept of adjustable robustness with respect to the framework used before is according to [Sch09b]:

Definition 3.8. Given a general uncertain optimization problem (P(ξ)) like before with the scenario ξ out of some uncertainty set with objective function f and constraint function F, the *adjustable robust counterpart* is

$$\begin{array}{lll}(\mathrm{AR}) & \min\limits_{p} & \sup\limits_{\xi \in U} \min\limits_{q} \{f((p,q),\xi) : F((p,q),\xi) \geq 0\} \\ & \text{s. t.} & \forall \xi \in U \ \exists q : F((p,q),\xi) \geq 0\end{array}$$

Note that this formulation is more general, and plugging in the aperiodic timetabling problem may at first glance look different from the (LP-AR) notation above, because now again a worst case is aimed to be minimized in the objective function, that can depend on the optimal values taken on by the wait-and-see variables after the scenario becomes known. Nevertheless, the following reduction will reveal that this worst-case (second-stage) optimum is entirely independent from the non-adjustable (first-stage) variables when chosen as suggested, thus having the worst optimal wait-and-see decisions' impact on the objective value being constant with regard to the here-and-now decisions.

Definition 3.9. The *adjustably reduced counterpart* (E', A', l', w') of an EAN (E, A, l, w) with regard to an uncertainty set U consists of the non-trivial nodes

$$E' = \left\{e \in E : d^-(e)d^+(e) \neq 1\right\}$$

and the trivial paths in the original EAN

$$A' = \left\{ \begin{array}{rr} ((e_0, e_1), (e_1, e_2), ..., (e_{n-1}, e_n)) \in A^* : & n \in \mathbb{N}, e_0, e_n \in E', \\ & e_1, ..., e_{n-1} \in E \setminus E' \end{array} \right\}$$

with every path meant as an arc from the source node of the first original arc to the target node of the last original arc. There may be multiple arcs connecting the same pair of nodes that shall remain distinguishable. The lower bounds respect the sum of the corresponding lower bounds in the worst case corresponding to U:

$$l'_{a'} = \sup_{\xi \in U} \sum_{a \in a'} \xi_a$$

The weight of a path is the minimum weight along it:

$$w'_{a'} = \min_{a \in a'} w_a$$

This reduction and the following proof of equivalence are based on an idea in [Hö11], where the independence of the optimal solutions from the worst optimal adjustable variables is however shown for uniform activity weights only while derived for arbitrary non-negative weights on activities here.

Lemma 3.10. *A timetable π' for the non-trivial events E' of an adjustably reduced counterpart (E', A', l', w') to an EAN (E, A, l, w) with regard to an uncertainty set U can be extended to a feasible timetable π for (E, A, l, w) for all $l \in U$ if and only if it is feasible for (E', A', l', w').*

Proof. Let π' be feasible for (E', A', l', w'). Then for any trivial path $(a_1, ..., a_n) = a' \in A'$ with $a_i = (e_{i-1}, e_i) \in A$ and all scenarios $l \in U$ holds

$$\pi'_{e_n} - \pi'_{e_0} \geq l'_{a'} = \sup_{\xi \in U} \sum_{a \in a'} \xi_a \geq \sum_{a \in a'} l_a$$

Hence, there are $\pi_{e_1}, ..., \pi_{e_{n-1}}$ respecting $\pi_j - \pi_i \geq l_a$ for all $a = (i, j) \in a'$, for example

$$\pi_{e_j} = \pi_{e_{j-1}} + l_{(j-1,j)}$$

inductively for $j = 1, ..., n-1$, with $\pi_{e_0} = \pi'_{e_0}$, fulfilling all lower bounds of scenario l with equality for all but the last arc, and for the last arc

$$\begin{aligned}
\pi'_{e_n} - \pi_{e_{n-1}} &= \pi'_{e_n} - (\pi_{e_{n-2}} + l_{(e_{n-2}, e_{n-1})}) \\
&= ... = \pi'_{e_n} - (\pi'_{e_0} + \sum_{j=0}^{n-2} l_{(e_j, e_{j+1})}) \\
&= \pi'_{e_n} - \pi'_{e_0} - \sum_{j=0}^{n-2} l_{(e_j, e_{j+1})} \\
&\geq \sum_{j=0}^{n-1} l_{(e_j, e_{j+1})} - \sum_{j=0}^{n-2} l_{(e_j, e_{j+1})} \\
&= l_{(e_{n-1}, e_n)}
\end{aligned}$$

No further constraints must hold for the times of any trivial node e since $d^+(e) = 1$.

Contrarily, if π' is not feasible for (E', A', l', w'), then there is some constraint violated, i. e. there is $a' = (a_1, ..., a_n) \in A'$ with $a_i = (e_{i-1}, e_i) \in A$ such that

$$\pi'_{e_n} - \pi'_{e_0} < l'_{a'} = \sup_{\xi \in U} \sum_{a \in a'} \xi_a$$

which implies that there must be some scenario $\xi \in U$ with

$$\sum_{a \in a'} \xi_a > \pi'_{e_n} - \pi'_{e_0} \Longleftrightarrow \pi'_{e_n} < \pi'_{e_0} + \sum_{j=1}^{n} \xi_{a_j}$$

If π was a feasible timetable for (E, A, l, w) for scenario $l = \xi$, it would fulfill

$$\begin{aligned}
\pi_{e_n} &\geq \pi_{e_{n-1}} + l_{a_n} \\
&\geq \pi_{e_{n-2}} + l_{a_{n-1}} + l_{a_n} \\
&\geq ... \geq \pi_{e_0} + \sum_{j=1}^{n} l_{a_j}
\end{aligned}$$

such that π cannot be an extension of π'. □

This permits for the adjustable robust counterpart of the aperiodic timetabling problem to have the constraints formulated as in the nominal linear program regarding the adjustably reduced counterpart.

Theorem 3.11. *A solution π' to the adjustable robust aperiodic timetabling problem on an aperiodic EAN (E, A, l, w) with uncertainty set U and all non-trivial nodes being non-adjustable,*

$$\begin{aligned}
\textit{(TT-AR)} \quad & \min_{\pi'} \sup_{\xi \in U} \min_{\pi \supset \pi'} \left\{ \tilde{w}^T \pi : \tilde{A}\pi \geq \xi \right\} \\
& s.\ t. \forall \xi \in U \exists \pi \supset \pi' : \tilde{A}\pi \geq \xi
\end{aligned}$$

is optimal for this problem if and only if it is optimal for the nominal timetabling *problem on the adjustably reduced counterpart* (E', A', l', w'),

$$\begin{aligned} \min \quad & \tilde{w}'^T \pi' \\ s.\ t. \quad & \tilde{A}'\pi' \quad \geq l' \end{aligned}$$

where $\pi \supset \pi'$ *means that* π *is an extension of* π'*, and* $\tilde{A}$*,* $\tilde{A}'$ *are the incidence matrices of* (E, A) *and* (E', A')*, respectively, and* $\tilde{w}$*,* $\tilde{w}'$ *are the corresponding event weight vectors to* w *and* w'*.*

Proof. The equivalence of feasibility has been shown in the lemma. Let now π' be any feasible solution to both problems, i. e. $\tilde{A}'\pi' \geq l'$. Then for any scenario $\xi \in U$

$$
\begin{aligned}
& \min_{\pi \supset \pi'} \left\{ \tilde{w}^T \pi : \tilde{A}\pi \geq \xi \right\} \\
\overset{(1)}{=} & \min_{\pi \supset \pi'} \left\{ \sum_{a=(i,j)\in A} (\pi_j - \pi_i) w_a : \tilde{A}\pi \geq \xi \right\} \\
\overset{(2)}{=} & \min_{\pi \supset \pi'} \left\{ \sum_{a' \in A'} \sum_{a=(i,j)\in a'} (\pi_j - \pi_i) w_a : \tilde{A}\pi \geq \xi \right\} \\
\overset{(3)}{=} & \sum_{a' \in A'} \min_{\pi \supset \pi'} \left\{ \sum_{a=(i,j)\in a'} (\pi_j - \pi_i) w_a : \tilde{A}\pi \geq \xi \right\} \\
\overset{(4)}{=} & \sum_{a' \in A'} \min_{\pi \supset \pi'} \left\{ \sum_{a=(i,j)\in a'} (\xi_a + \pi_j - \pi_i - \xi_a) w_a : \pi_j - \pi_i \geq \xi_a \forall a = (i,j) \in a' \right\} \\
\overset{(5)}{=} & \tilde{c}_\xi + \sum_{a' \in A'} \min_{\pi \supset \pi'} \left\{ \sum_{a=(i,j)\in a'} (\pi_j - \pi_i - \xi_a) w_a : \pi_j - \pi_i \geq \xi_a \forall a = (i,j) \in a' \right\} \\
\overset{(6)}{=} & \tilde{c}_\xi + \sum_{a'=(e_0,\dots,e_n)\in A'} \min_{d} \left\{ \sum_{a \in a'} d_a w_a : d_a \geq 0 \ \forall a \in a',\ \sum_{a \in a'} d_a = \pi'_{e_n} - \pi'_{e_0} \right\} \\
\overset{(7)}{=} & \tilde{c}_\xi + \sum_{a'=(e_0,\dots,e_n)\in A'} (\pi'_{e_n} - \pi'_{e_0}) \min_{a \in a'} w_a \\
\overset{(8)}{=} & \tilde{c}_\xi + \sum_{a'=(e_0,\dots,e_n)\in A'} (\pi'_{e_n} - \pi'_{e_0}) w'_{a'} \\
\overset{(9)}{=} & \tilde{c}_\xi + \tilde{w}'^T \pi'
\end{aligned}
$$

Equation (1) follows from Lemma 2.8, (2) is essentially due to the arc-disjointness of the trivial paths and (3) due to the inner-node-disjointness of trivial paths and the fixation of any non-trivial node by the given timetable π', such that all trivial paths can be optimized independently in the second stage. (4) focuses every independent path optimization on its own constraints and adds a zero, (5) pulls out the costs of minimal durations imposed by the scenario ξ:

$$\tilde{c}_\xi = \sum_{a' \in A} \sum_{a \in a'} \xi_a w_a = \sum_{a \in A} \xi_a w_a$$

Finally, (6) rewrites the remaining minimization problem with slack variables

$$d_a = \pi_j - \pi_i - \xi_a \ \forall a = (i,j) \in a' \in A'$$

(7) employs its obvious solution, and (8) rewrites it according to the definition of the adjustably reduced EAN and (9) again by Lemma 2.8.

Since $\tilde{w}'^T \pi'$ is not depending on the scenario ξ, the objective function of (TT-AR) reduces to

$$\begin{aligned} \sup_{\xi \in U} \min_{\pi \supset \pi'} \left\{ \tilde{w}^T \pi : \tilde{A}\pi \geq \xi \right\} &= \sup_{\xi \in U} \left\{ \tilde{c} + \tilde{w}'^T \pi' \right\} \\ &= \tilde{w}'^T \pi' + \sup_{\xi \in U} \tilde{c}_\xi \end{aligned}$$

As the two objective functions now only differ in the additive constant $\sup_{\xi \in U} \tilde{c}_\xi$, that is not depending on the solution π', the optimal solutions of both problems are the same. □

It is essential to note that this reducibility to a nominal instance is due to the choice of at least all non-trivial nodes being non-adjustable. The objective function with minimal arc weight per path employs the idea that any additional slack and buffer on a path will go entirely on its *cheapest* arc(s), which is possible due to the absence of upper bounds.

Although an arbitrary choice of non-adjustable variables for the here-and-now decisions might still retain the adjustable robust counterpart a linear program because the reduction of the constraints is not depending on the disjointness of the paths, the linearisation of the objective function with the supremum over all scenarios would make the problem grow vastly even for uncertainty set $U_2(s,k)$. The worst case cannot be controlled as simply as before, and therefore, for every choice of k activities to be delayed as much as possible, the entire second-stage optimization has to be moved to the constraints in order to implement the supremum for the worst-case optimization.

A further annotation has to be made here. The concept of not fixing some variables before the scenario becomes known has two drawbacks in the application to timetabling. First, customers almost always get a schedule for all stops of the transportation means, and the operation should never deviate from it towards *earlier* departures. Therefore, a timetable would rather have to put the entire shared buffer of a path on the very last activity. A second reason for this is that delays often only reveal immideately during the activities they affect, and hence even a trivial path could not be optimized in the second stage before the vehicle would enter it. Given these practical restrictions, the adjustable robust counterpart as laid out here is not really minimizing the worst case, but it may still be a good heuristic.

4 Numerical studies

In order to evaluate timetables with regard to their robustness against delays, the very first and most important question is how to measure delay robustness at all. The possibly most generic approach is to try and calculate the impact that certain delay scenarios have on the customers under the timetable that is to be evaluated. Due to the complexity and size of the timetabling problem, this is done here by heuristically applying so-called *delay-management* to a finite set of scenarios chosen randomly from the respective uncertainty set that the robustness concepts aimed to optimize the timetable against. This delay-management step is sketched in the following subsection.

For the numerical studies, a framework called LinTim has been employed. It provides algorithms for several planning stages in public transportation, such as lineplanning and timetabling as well as delay-management. The robust timetabling methods described above have been plugged into this framework and then carried out and evaluated on a public transportation network instance built upon real-world data provided by Deutsche Bahn AG. This network approximates the German high-speed intercity train infrastructure and concists of 249 stations and 325 connections between them.

For the numerical studies, a total of six sets of scenarios has been created randomly by a uniform distribution over the driving activities and over the respective interval the delayed minimal duration may lie in. Five sets have been chosen from $U_1(0.1), U_1(0.2), U_1(0.3), U_1(0.4), U_1(0.5)$ with exactly ten percent of the driving activities being delayed in each scenario. One set has been chosen from $U_2(0.5, 1)$. Each of those six sets consists of 100 scenarios.

4.1 Delay-management techniques

The scenarios that are chosen randomly from the uncertainty sets are imposing some increased lower bounds on driving durations of trains between stations. These *source delays* do not only have their direct impact on the passengers in this train reaching the next station late, but rather propagate to many subsequent events like further departures and arrivals of this train, and even onto other trains when there are passengers who have to change over in the following or a subsequent station. To measure the impact of a scenario, those *secondary delays* have to be considered as well.

To this end, so-called *disposition timetables* have to be calculated for each scenario, which indicate new departure and arrival times for all trains in all stations respecting the occured delays. In a disposition timetable, no event may be scheduled to an earlier time than it had been scheduled in the timetable before, because customers are usually given and relying on the guarantee that means of public transportation never leave earlier than announced.

The simplest method to compute this disposition timetable is calling an algorithm known as the critical path method from project planning, as sketched and referenced in the proof of Lemma 2.6. It traverses all events in topological ordering and assures the new lower bounds to be met by setting back the timetable as needed.

There are two further prominent considerations to be made. The first one is the common question for trains: to wait or not to wait if a feeder train is delayed. To wait would result in additional delay for other customers, while not to wait could increase the inconvenience of the already delayed passengers who would miss their connection. Usually rules of thumb are applied like a maximum waiting time or a classification of trains into local and high-speed trains, the latter never wating for the former.

The second consideration is concerned with the headways, the security distances between trains that had been fixed and are used to model capacity constraints on tracks. These would essentially delay other trains even if there are no changing customers between the primarily delayed train and a train which is scheduled to depart from a certain station after it and could therefore be turned around.

Both of these questions make the delay-management problem hard as they result in decision variables turning the linear program into a mixed-integer program. This can be solved exactly or heuristically, as described for example in [SS10] and [Sch09a].

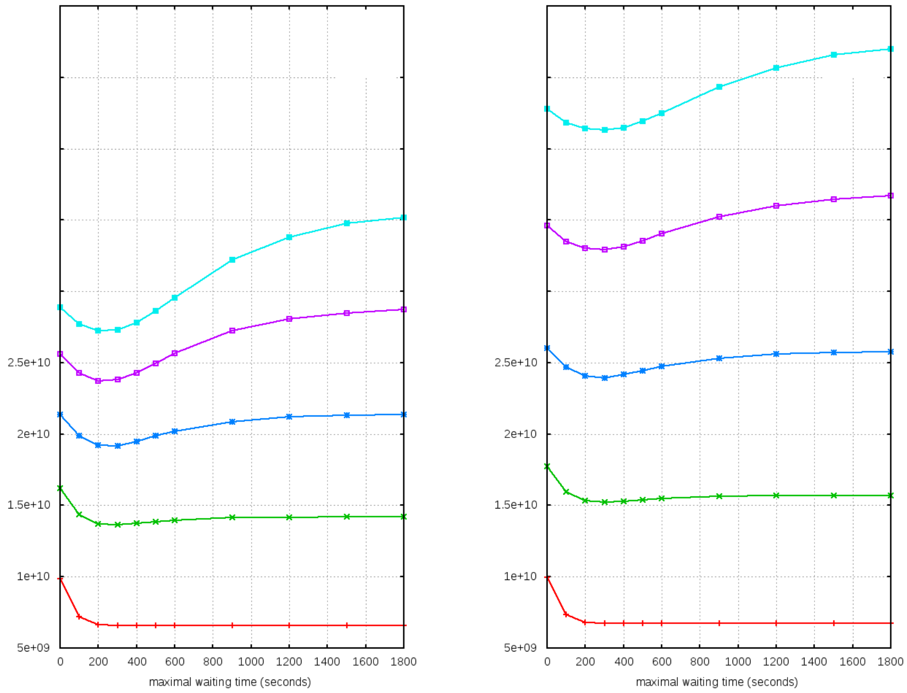
However, the main investigation on delays in the aperiodic case here shall be made using the following propagation algorithm, which is given the EAN

(E, A, l, w) and original timetable π together with the occured scenario ξ as input as well as a maximum waiting time t, and computes a disposition timetable $\tilde{\pi}$ complying to the new minimal durations ξ.

1. Work on a copy (E', A') of the EAN. Create an empty to-do list Q.
2. Iterate over all $e \in E'$, set $\tilde{\pi}_e := \pi_e$ and add e to Q if $d^+(e) = 0$.
3. As long as there is $e \in Q$:
   (a) Remove e from Q.
   (b) For all outgoing activities $a = (e, o) \in A'$, calculate $n_o = \tilde{\pi}_e + \xi_a$.
   (c) If $n_o > \tilde{\pi}_o$ and $a \in A^{\text{drive}} \sqcup A^{\text{wait}}$, set $\tilde{\pi}_o := n_o$.
   (d) Else if $n_o > \tilde{\pi}_o$ and $a \in A^{\text{change}}$ and $n_o \leq \pi_o + t$, set $\tilde{\pi}_o := n_o$.
   (e) Else if $n_o > \tilde{\pi}_o$ and $(e, o) \in A^{\text{head}}$ and $\tilde{\pi}_o + \xi_a > \tilde{\pi}_e$, set $\tilde{\pi}_o := n_o$.
   (f) Else if $(e, o) \in A^{\text{head}}$ and $\tilde{\pi}_o + \xi_a \leq \tilde{\pi}_e$, remove (e, o) from A' and add (o, e) to A'.
   (g) Else if $(o, e) \in A^{\text{head}}$ and $n_o > \tilde{\pi}_o$, remove (e, o) from A' and add (o, e) to A', and add e to the head of the to-do list Q.
   (h) Remove a from A'.
   (i) If now $\delta^+(o) \cap A' = \emptyset$, add o to the end of Q.

Lines 2, 3(h) and 3(i) manage the traversal in a topological ordering. Line 3(f) turns a headway around if the originally later train can leave before the delayed train without imposing an additional delay on it. Line 3(g) swaps a headway back whenever a swapped headway would cause additional delay to the train that should originally go first.

A closer look is now taken on the maximum waiting time parameter t. For different values of t, the algorithm above has been executed on the nominally optimal aperiodic timetabling with the five random delay sets out of U_1. A second run of this has been done without having the algorithm swap around any headways in order to examine the impact of this method. The resulting delays have been averaged over each scenario set according to the objective



(a) with swapping of headways (b) without swapping of headways

Figure 1: Delay-management objective value for different delay intensities

function of the delay-management optimization problem, which is described next.

Figure 1 shows the result. With increasing maximal waiting time parameter, the objective value first decreases to a minimum around $t = 200$ seconds, almost uniformly for all s from 0.1 to 0.5. Up to here, the incurring delays for passengers of subsequent trains is less than the benefit of changing passengers not missing their connection. Then the objective increases again until it eventually becomes constant when the maximum waiting time is large enough for all trains to wait for each other, which is of course later the case when the occuring delays can become larger. The qualitative behaviour is apparently not changed much by the admission of headway swaps.

The delay-management objective function aims at measuring the delay for passengers. To this end, it calculates the increase of travel time, weighted by the number of customers on the respective activities. Furthermore, for missed connections, the period length T of the underlying periodic EAN is added per customer. In the case of the so-called *all-wait policy* where no connection is missed, this represents the real increase of travel time. When connections are missed, this is only an approximation to the real delay the customers have to face, for two reasons. First, although the customers miss their connection, a possible delay of the missed train still accounts for them, as they are contained in the original weight, although the periodically next train may arrive on time. Second, customers may not need to wait an entire period, since there could be alternative lines for them to reach their destination. This problem is faced by *delay-management with re-routing of passengers*, see for example [DHSS12]. However, the shortcomings of the objective function are accepted here, and most investigation is done with the all-wait policy, in which no connections are missed at all.

4.2 Light robustness results

For the examination of the light robustness concept, light robust timetables have been computed for different values of parameters $\delta = 0.1s, 0.2s, ..., 1.0s$, each for uncertainty sets $U_1(s)$ for $s = 0.1, ..., 0.5$. The strict robust solution is contained as the special case $\delta = s$. The weight function is set to $w = 1$.

Plots are showing the objective value of delay-management against the nominal quality in terms of total weighted slack contained in the timetable. From left to right, the tightness of the schedule decreases, while from bottom to top its delay-resistance decreases. Thus, an ideal timetable would reside in the bottom left corner of the plot.

Figure 2 compares the objective value achieved by different delay-management techniques: the propagation algorithm with an all-wait and no-wait policy as well as with maximum waiting time of 200 seconds. Furthermore, the best of several heuristics to the exact ip-based method due to [Sch09a] shows that there is still some room to the best possible treatment of delays, but the propagation method is performing not too bad and runs in polynomial time.

Figure 3 shows the behaviour of the light robustness concept in detail. Each colour represents another scenario set with different values for s, from 10 percent increase up to 50 percent increase of the minimal durations. Figure 4

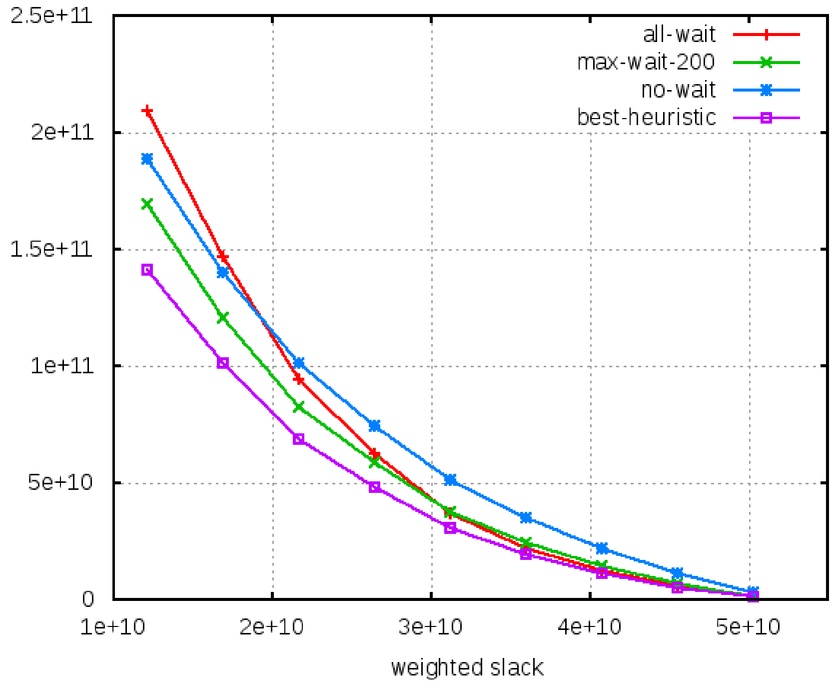


Figure 2: Objective value for different delay-management techniques

tries to visualize the impact of several factors that the investigation relies on. The lineplanning stage may have a large influence, as different objectives such as the direct traveller's approach may yield entirely different lines. Usually the cost model for lineplanning has been employed. For further details on lineplanning, see for example [Sch06]. The complete neglection of headways seems not to make a great qualitative difference, nor does the weighted light approach with a light robustness modell with weights on the constraints according to the respective passenger loads. However, the next subsection with the overall comparison reveals that there is a great quantitative difference between the unweighted and the weighted light robust optimization.

The common character of the light robust curve shows that, within the setting of light robustness, the first relaxation of the nominal optimality yields a rather strong improvement in terms of delay robustness, while the curve flattens towards the strict robust solution at the bottom right end of the curve. The trade-off between nominal quality and robustness seems worthwile.

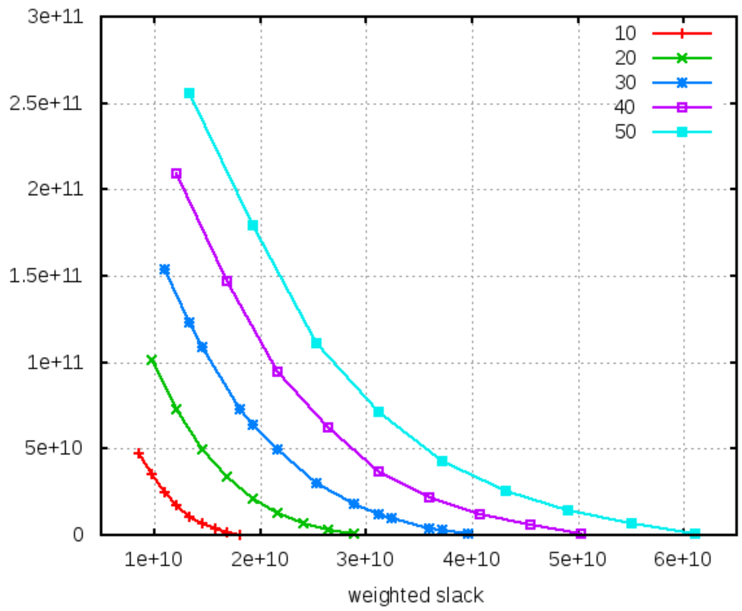
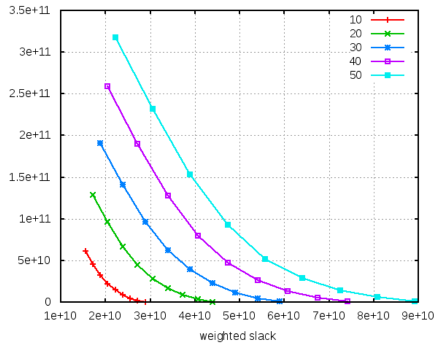
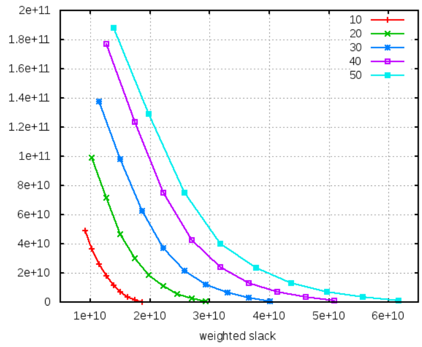


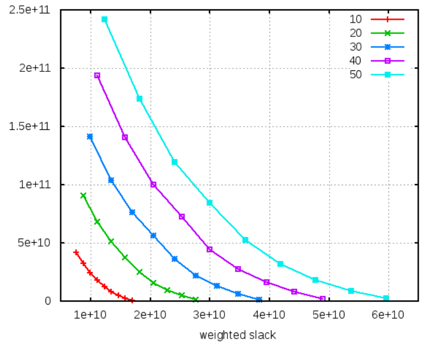
Figure 3: Objective value for light robustness on cost model lines



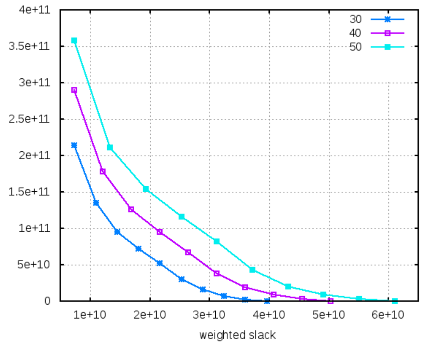
(a) on a larger network



(b) direct travellers lines



(c) without headways



(d) weighted light robustness

Figure 4: Delay-management objective value for different variations

4.3 Adjustable robustness results and comparison

Finally enter the adjustable robustness. In order to make it comparable by means of delay-management, the timetables had to be extended with wait-and-see decisions made before the scenarios became known. This has been done in two ways: First by putting all the buffer on the cheapest arc per path, and then with putting the necessary part of the buffer on the last arc per path. Since there are no trivial paths in the EAN in the case of headways, these two versions are compared in the cases of no headways, on the sets out of $U_1(s)$ as well as on a set out of $U_2(0.5, 1)$.

The results can be seen in figures 5, 6 and 7. The weighted light robust solution outperforms the unweighted leight robust solution in the area of much and little relaxation; however, with medium relaxation, the unweighted light robust solution is a trifle better with regard to delay resistance. On $U_1(0.5)$ without headways, the adjustable robust solution with the necessary buffer on the last arc per path can compete with them, and in the case of $U_2(0.5, 1)$, which is somehow tailored to adjustable robustness, it is actually better.

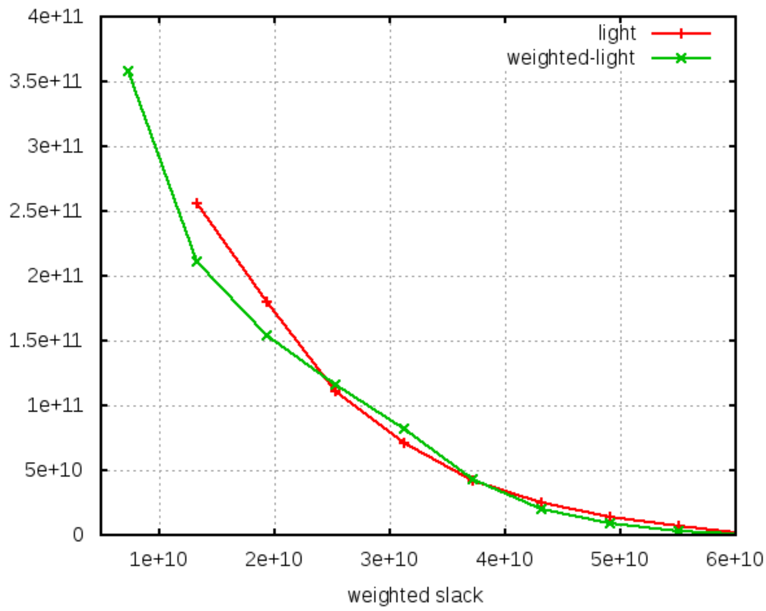


Figure 5: $U_1(0.5)$ with headways

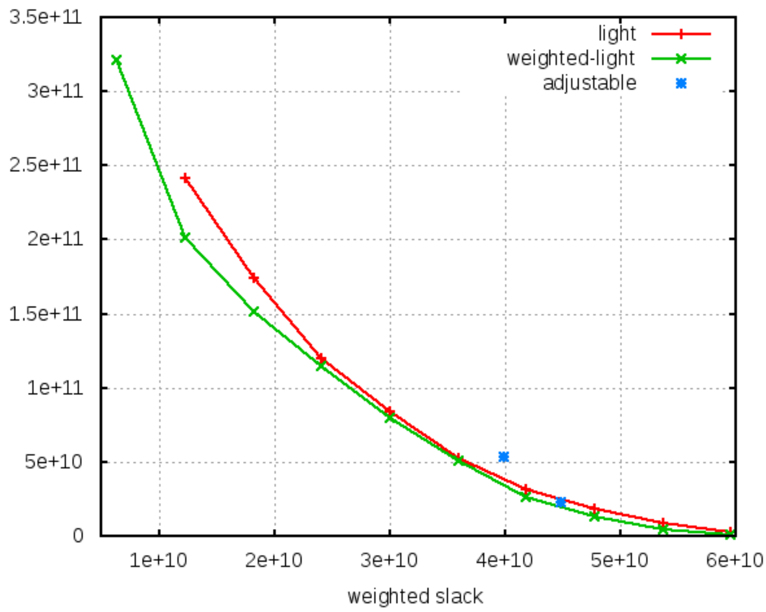


Figure 6: $U_1(0.5)$ without headways

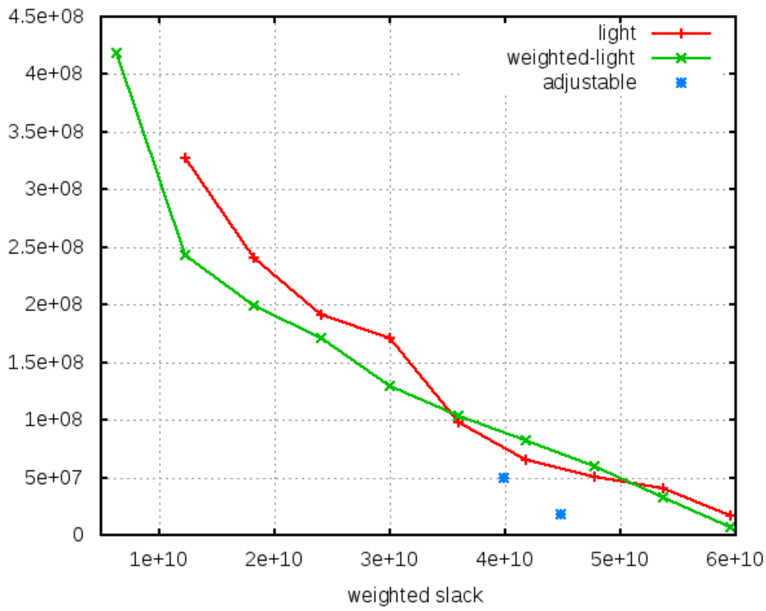


Figure 7: $U_2(0.5, 1)$ without headways

5 Conclusion

The conduced work shows that robust optimization models are not generally inapplicable to the aperiodic timetabling problem. The conservatism of strict robustness can well be addressed by common relaxations. Still the choice of the right relaxation depends on the assumptions about the uncertainty set, and as long as the behaviour of disturbances in public transport remains unpredictable, the approach is of rather theoretical interest.

Especially the case of U_2 shows that the simulation on random scenarios cannot be regarded as proof, as the curve gets rather unsmooth, and such unsmoothness represents to some extent the inpredictability of delays and their impact.

Nevertheless, the concepts of light robustness, especially with weights on the constraints, and the adjustable robustness appear to be suitable for further investigations beyond the scope of this work.

6 Implementation details

The software packet LinTim has been used in its v2 branch as of svn revision 1586. All the files that have been additionally put in or altered are supplied on the accompanying CD as well as the data produced. Note that the bash scripts have been used somehow interactively, renaming folders here and there and making slight changes to the scripts as needed. The implementation details are provided in an "as-is" fashion, and convenience and ease of use are no objectives in providing them.

7 Acknowledgements

I want to thank everyone involved in this thesis: First of all to my supervisor, Prof. Dr. Anita Schöbel, for her valuable advice and patience; furthermore to the current and former co-workers on the LinTim project for their main contributions, thanks to Marc Goerigk, Dr. Michael Schachtebeck and Michael Siebert, whose implementations this work is heavily based on.

References

[BTGGN04] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351 – 376, 2004.

[BTN98] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769 – 805, 1998.

[BTN00] A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88(3):411, 2000.

[DHSS12] T. Dollevoet, D. Huisman, M. Schmidt, and A. Schöbel. Delay management with rerouting of passengers. *Transportation Science*, 46(1):74–89, February 2012.

[FM09] M. Fischetti and M. Monaci. Light robustness. In Ravindra Ahuja, Rolf Möhring, and Christos Zaroliagis, editors, *Robust and Online Large-Scale Optimization*, volume 5868 of *Lecture Notes in Computer Science*, pages 61–84. Springer Berlin / Heidelberg, 2009.

[GS10] M. Goerigk and A. Schöbel. An empirical analysis of robustness concepts for timetabling. In Thomas Erlebach and Marco Lübbecke, editors, *Proceedings of the 10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, volume 14 of *OpenAccess Series in Informatics (OASIcs)*, pages 100–113, Dagstuhl, Germany, 2010. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Gus02] E. Guslitzer. Uncertainty-immunized solutions in linear programming. Master's thesis, Technion, Israeli Institute of Technology, 2002.

[Hö11] T. Höche. Robuste aperiodische fahrplangestaltung. Diploma thesis, Georg-August-Universität Göttingen, 2011.

[KJ61] J. E. Kelley Jr. Critical-path planning and scheduling: Mathematical basis. *Operations Research*, 9(3):296 – 320, 1961.

[KW94] P. Kall and S. W. Wallace. *Stochastic programming.* New York, NY: Wiley., 1994.

[LS06] C. Liebchen and S. Stiller. Delay resistant timetabling. Technical report, ARRIVAL Project, November 2006. Presented at CASPT 2006.

[Pre95] A. Prekopa. *Stochastic programming.* Mathematics and its Applications (Dordrecht). 324. Dordrecht: Kluwer Academic Publishers., 1995.

[Roc84] R. T. Rockafellar. *Network Flows and Monotropic Optimization.* A Wiley-Interscience publication. John Wiley & Sons., 1984.

[Sch06] A. Schöbel. *Optimization in public transportation. Stop location, delay management and tariff zone design in a public transportation network.* Springer Optimization and Its Applications 3. New York, NY: Springer, 2006.

[Sch09a] M. Schachtebeck. *Delay Management in Public Transportation: Capacities, Robustness, and Integration.* PhD thesis, Georg-August-Universität Göttingen, 2009.

[Sch09b] A. Schöbel. Robust optimization. Lecture notes, 2009.

[Soy73] A. L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5):1154 – 1157, 1973.

[SS10] M. Schachtebeck and A. Schöbel. To wait or not to wait– and who goes first? delay management with priority decisions. *Transportation Science*, 44(3):307 – 321, 2010.

[SU89] P. Serafini and W. Ukovich. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2(4):550–581, 1989.