Online Delay Management

Sven O. Krumke* Clemens Thielen* Christiane Zeck*

Abstract

We present extensions to the Online Delay Management Problem on a Single Train Line. While a train travels along the line, it learns at each station how many of the passengers wanting to board the train have a delay of δ. If the train does not wait for them, they get delayed even more since they have to wait for the next train. Otherwise, the train waits and those passengers who were on time are delayed by δ. The problem consists in deciding when to wait in order to minimize the total delay of all passengers on the train line.

We provide an improved lower bound on the competitive ratio of any deterministic online algorithm solving the problem using game tree evaluation. For the extension of the original model to two possible passenger delays δ_1 and δ_2, we present a 3-competitive deterministic online algorithm.

Moreover, we study an objective function modeling the refund system of the German national railway company, which pays passengers with a delay of at least Δ a part of their ticket price back. In this setting, the aim is to maximize the profit. We show that there cannot be a deterministic competitive online algorithm for this problem and present a 2-competitive randomized algorithm.

Keywords: delay management, online optimization, competitive analysis, public transportation

1 Introduction

The Online Delay Management Problem on a Single Train Line was introduced by Gatto et al. [8]. The input of this problem is a train line with n stations such that the train stops at each station $1, 2, ..., n$. Also known in advance are fixed *passenger trails* p_{ij}, which stand for a number of passengers who want to take the train from station i to station j for $1 \leq i < j \leq n$. When the train arrives at a station k, it learns how many of the passengers p_{ij} with $i = k$ planning to get on the train at station k are late. The delayed passengers have a delay of δ minutes due to delayed feeder trains. We call δ the *source delay* of the delayed passengers. By d_{ij} we denote the *delayed passengers* traveling

*Department of Mathematics, University of Kaiserslautern, Paul-Ehrlich-Str. 14, 67663 Kaiserslautern, Germany. {krumke,thielen,zeck}@mathematik.uni-kl.de

from i to j and by o_{ij} the *on time passengers*. Hence, $p_{ij} = d_{ij} + o_{ij}$. The problem is to decide whether the train should wait for the delayed passengers or depart on time. If it waits, the on time passengers get delayed by δ minutes. If it departs on time, the delayed passengers have to wait for the next train that runs in $T > \delta$ minutes, so their delay then increases to T minutes.

1.1 Previous Work

In the paper of Gatto et al. [8] the aim is to minimize the total passenger delay. For this problem, the authors present a family of 2-competitive online algorithms and prove that the golden ratio is a lower bound on the competitive ratio of any deterministic online algorithm. Anderegg et al. [2] also investigate an online transportation problem. They consider a sequence of buses serving a station one after another and assume that passengers arrive at the station constantly at a certain rate. Assuming that the last bus is delayed, the question is how long the other buses should wait in order to minimize the total delay of all passengers. To our knowledge, these are the only papers treating delay management problems as theoretical online problems. For details on online optimization, we refer the reader to Borodin et al. [5].

Other authors consider complex realistic models for the delay management problem. Adenso-Díaz et al. [1] apply heuristics and provide a decision tool that can be used in practice. Biederbick et al. [4] discuss decision support tools which are based on agent-based simulation. Berger et al. [3] provide a simulation platform for testing different heuristics and give a PSPACE-hardness proof for the online delay management problem they consider.

Other papers focusing on complexity are [6, 7]. The authors investigate different models for the delay management problem and examine the border between the polynomially solvable and $\mathcal{NP}$-hard variants.

Another approach to the delay management problem is the formulation as mixed integer program [11, 12].

1.2 Our Contribution

In Section 2, we study the objective function of minimizing the total delay of all passengers. We show that, for the case of 3 stations, the lower bound on the competitive ratio by Gatto et al. [8] is tight by giving an algorithm which actually achieves this competitive ratio. It remains an open question to close the gap for more than 3 stations. Viewing the problem as a game and using game tree evaluations, we can determine an optimal online algorithm and present an improved lower bound for $n > 3$ stations. Moreover, we extend the model to two possible source delays δ_1 and δ_2 and present a 3-competitive online algorithm for this more general setting.

In Section 3, we introduce an objective function maximizing the profit in a setting where passengers delayed by at least Δ get a refund

on their ticket price. We prove that there does not exist a deterministic competitive online algorithm for this problem and present a 2-competitive randomized online algorithm.

2 Minimizing Total Delay

2.1 One Possible Source Delay

In this section, we assume that all passengers who are delayed have the same source delay δ. In this setting, the train can wait at most once at a certain station. From this point on, it is delayed by δ minutes and can, hence, pick up all passengers (whether delayed or not) that get on the train later on. If the train waits at station $k \in \{1, \ldots, n\}$, the total delay of all passengers amounts to

$$D(k) = T \sum_{i<k,j} d_{ij} + \delta \sum_{i \geq k,j} d_{ij} + \delta \sum_{i,j>k} o_{ij}.$$

The delayed passengers who want to board the train before k have a delay of $T > \delta$, whereas all delayed passengers boarding the train after $k-1$ have a delay of δ. The on time passengers who leave the train after k end up having a delay of δ. Note that the option $k = n$ signifies that the train does not wait at all.

In order to increase readability, we omit double sums for i and j. Also, we do not indicate that we always assume $i < j$ in our sum notation since this is clear from the definition of the passenger trails.

We start by considering the Online Delay Management Problem for $n = 3$ train stations, i.e.,

$$\begin{aligned} D(1) &= \delta(p_{12} + p_{13} + p_{23}), \\ D(2) &= T(d_{12} + d_{13}) + \delta(p_{13} - d_{13} + p_{23}), \\ D(3) &= T(d_{12} + d_{13} + d_{23}). \end{aligned}$$

At station 1, we already know whether $D(1)$ is larger or smaller than $D(2)$ since none of the terms include the only unknown variable d_{23}. If $D(1) \geq D(2)$, waiting at 1 cannot be the (unique) optimal solution and, hence, it makes no sense to choose $D(1)$. If, however, $D(1) < D(2)$, we know that waiting at 2 yields a worse objective value than waiting at 1, but, still, waiting at 3 might be the optimal strategy. So we have to decide if we wait at station 1, or if it might be better not to wait with the chance of getting a better outcome by waiting at 3. Our algorithm makes this decision by determining "how much" smaller $D(1)$ is compared to $D(2)$.

ALG 1: Wait at station 1 if and only if $D(2) > \alpha \cdot D(1)$, where $\alpha \geq 1$ is a parameter to be chosen later.
In case we do not wait at station 1, wait at station 2 iff $D(2) < D(3)$, which is known then.

The competitive analysis shows that this online algorithm achieves a competitive ratio equal to the lower bound by Gatto et al. [8] and is in this sense best possible.

Theorem 2.1. *For $\alpha = \frac{1+\sqrt{5}}{2}$, ALG 1 is $\frac{1+\sqrt{5}}{2}$-competitive.*

Proof. In case that $D(2) \leq \alpha \cdot D(1)$, ALG 1 does not wait at station 1 and yields a cost of $\min\{D(2), D(3)\}$. If the offline optimal solution is $D(1)$, we get a ratio of

$$\frac{\text{ALG 1}}{\text{OPT}} = \frac{\min\{D(2), D(3)\}}{D(1)} \leq \frac{D(2)}{D(1)} \leq \alpha,$$

where OPT denotes the cost of an optimal (offline) solution.

If $D(2) > \alpha \cdot D(1)$, ALG 1 waits at station 1 which leads to a cost of $D(1)$. If the optimum is $D(3)$, we have

$$\begin{aligned}
\frac{\text{ALG 1}}{\text{OPT}} = \frac{D(1)}{D(3)} &= \frac{\delta(p_{12} + p_{13} + p_{23})}{T(d_{12} + d_{13} + d_{23})} \\
&\leq \frac{\delta(p_{12} + p_{13} + p_{23})}{T(d_{12} + d_{13})} \\
&\overset{D(2)>\alpha D(1)}{\leq} \frac{\delta(p_{12} + p_{13} + p_{23})}{\alpha\delta(p_{12} + p_{13} + p_{23}) - \delta(p_{13} - d_{13} + p_{23})} \\
&\leq \frac{\delta(p_{12} + p_{13} + p_{23})}{(\alpha - 1)\delta(p_{12} + p_{13} + p_{23})} \\
&\leq \frac{1}{\alpha - 1}.
\end{aligned}$$

For $\alpha = \frac{1+\sqrt{5}}{2}$, we have $\alpha = \frac{1}{\alpha-1} = \frac{1+\sqrt{5}}{2}$. □

Competitive analysis is often described as a game between an online algorithm ALG and an evil adversary ADV [5]. According to that, the Online Delay Management Problem can be written as a game in extensive form.

The root of the tree represents the situation at station 1. Given the delayed passengers, the online algorithm has two choices: Waiting at station 1 or not. Then, the evil adversary reveals how many of the passengers getting on the train at station 2 are delayed. This can be any number between 0 and the total number of people connecting at station 2. Then, if the online player has already waited before, it cannot wait any more. If she has not waited, she still has the choice to either wait or not wait. At the next station, the adversary again reveals the number of delayed passengers at station 3, and so on. The payoff of a leaf is defined as the corresponding competitive ratio, i.e., the total delay occurring when the actions of the players correspond to the path leading to the respective leaf divided by the cost of an optimal solution obtainable if the online player had known the decisions of the adversary in advance.

Figure 1 shows a game tree for $n = 2$ stations and 2 passengers connecting at station 2. At first, the online algorithm chooses between waiting (w) and not waiting (n). Then the adversary reveals if 0,1, or 2 passengers are delayed at station 2. If the algorithm has not waited yet, it again has the option to wait or not. Note that, also in case

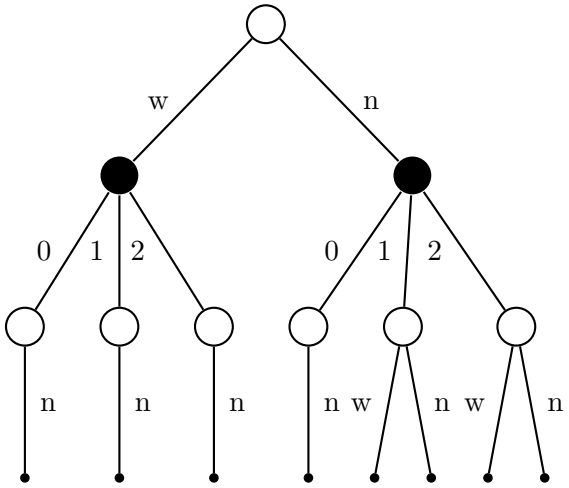

Figure 1: A game tree for 2 stations: The white nodes belong to the online algorithm, the black nodes to the adversary.

it has not waited and there are no delayed passengers at station 2, it makes no sense to wait there, so this branch can be omitted.

Now the Online Delay Management Problem translates to a two-person zero sum game in which the online algorithm wants to minimize the payoff, while the adversary tries to maximize it (for details on these games see [10]). Hence, the best option for the players is to act according to a minmax strategy: The online player will always choose the subtree in which the maximum payoff is minimal, whereas the adversary chooses the subtree in which the minimal payoff is maximal. Thus, for any given instance, the minmax strategy for the corresponding game tree determines the best online strategy and the best competitive ratio. The downside, however, is that the size of a game tree is exponential in the encoding length of the game. At each node, ALG has up to two possible choices, and ADV has $1 + p_i$ choices, where p_i is the number of passengers getting on the train at station i. Hence, we have $\mathcal{O}(2^{n-1} \prod_{i=1}^{n-1}(p_i + 1))$ leaves.

In order to find a lower bound for $n = 4$ stations, we implemented the minmax procedure and enumerated over the parameters of the game tree in order to gain insight into which could be "worst case" instances.

One scenario we have found is the following: Let $\delta = 1$, $p_{12} = d_{12} = 1, p_{14} = d_{14} = 12, p_{24} = 7$, and let T and p_{34} be large.

If an online algorithm ALG waits at station 1, let p_{24} and p_{34} be on time. Then it would have been optimal to not wait at all and we get $\frac{\text{ALG}}{\text{OPT}} = \frac{20+p_{34}}{13T}$.

If ALG does not wait at station 1, let p_{24} be delayed. If, in this case, ALG waits at station 2, let p_{34} be on time. Then again not

waiting at all would have been optimal and $\frac{\text{ALG}}{\text{OPT}} = \frac{13T+7+p_{34}}{20T}$.

If ALG does not wait at station 2, let p_{34} be delayed. Then, ALG waits at station 3, whereas waiting at station 1 would have been the optimal solution and $\frac{\text{ALG}}{\text{OPT}} = \frac{20T+p_{34}}{20+p_{34}}$.

Writing everything in a nonlinear program yields

$$\begin{aligned} 20 + p_{34} &\geq c \cdot 13T \\ 13T + 7 + p_{34} &\geq c \cdot 20T \\ 20T + p_{34} &\geq c \cdot (20 + p_{34}). \end{aligned}$$

Our aim is to find the maximum c such that the program is feasible. Note that, for a fixed c, the program becomes linear and we can check feasibility easily. We found that the program is still feasible for $c = 1.83733373$ (with a precision of 8 digits) using binary search. Values that are feasible for the corresponding program are $p_{34} = 1111318826$ and $T = 46527238.72702379$. This improves the lower bound of Gatto et al. [8].

Theorem 2.2. *No deterministic online algorithm for the Online Delay Management Problem on a Single Train Line can have a competitive ratio smaller than 1.83733373.*

2.2 Two Possible Source Delays

In this section, we allow two different source delays, i.e., passengers who want to board the train might either be on time, have a delay of δ_1, or a delay of δ_2, where $\delta_1 < \delta_2 < T$. In this new scenario, the train can wait at most twice. It can wait δ_1 minutes at station k and an additional $\delta_2 - \delta_1$ minutes at a station $l \geq k$. If $l = k$, this means that the train does not wait until station l where it waits δ_2 minutes. As in the original model, the train cannot catch up on its delay, so, starting from station k, the train has a delay of δ_1 minutes, whereas its delay amounts to δ_2 minutes from station l on.

Similar to the notation used before, we write p_{ij} for all passengers traveling from station i to station j. These passengers are divided into the on time passengers o_{ij}, the passengers d^1_{ij} with source delay δ_1 and the passengers d^2_{ij} with source delay δ_2.

By $D(k,l)$ we denote the total delay of all passengers occurring when the train waits δ_1 minutes at station k and $\delta_2 - \delta_1$ minutes at station $l \geq k$, i.e.,

$$\begin{aligned} D(k,l) = \quad & \delta_1 \sum_{i,j} d^1_{ij} + \delta_2 \sum_{i,j} d^2_{ij} \\ & + (T - \delta_1) \sum_{i<k,j} d^1_{ij} + (T - \delta_2) \sum_{i<l,j} d^2_{ij} \\ & + \delta_1 \sum_{i,j>k} o_{ij} + (\delta_2 - \delta_1) \sum_{i,j>l} o_{ij} \\ & + (\delta_2 - \delta_1) \sum_{i \geq k, j>l} d^1_{ij}, \end{aligned}$$

where the particular sums can be explained as follows: The first line includes the source delays of all delayed passengers, which cannot be optimized. The second line contains the extra delay of the passengers with source delay who miss the train, i.e., passengers with source delay δ_1 who want to board the train before station k and passengers with source delay δ_2 who want to board it before l. The next line contains the delay of the on time passengers. They reach their destination with a delay of δ_1 minutes if they leave the train after station k and with a delay of δ_2 minutes if they leave after station l. The last line represents the delay of the passengers with source delay δ_1 who board the train at or after station k and leave it after station l and end up getting delayed by $\delta_2 - \delta_1$ more minutes.

We are now ready to extend the algorithm by Gatto et al. [8] for the online delay management problem on a single train line to our new scenario with two different source delays:

ALG 2:

- If the train has not waited before and, at station k,

$$T \sum_{\substack{i \leq k \\ j}} d_{ij}^1 \geq \delta_1 (\sum_{\substack{i \leq k \\ j > k}} o_{ij} + \sum_{\substack{i > k \\ j}} p_{ij}), \tag{1}$$

  wait until the delay of the train is at least δ_1 minutes.

- If the train has not waited before and, at station l,

$$T \sum_{\substack{i \leq k \\ j}} d_{ij}^2 \geq \delta_2 (\sum_{\substack{i \leq k \\ j > k}} o_{ij} + \sum_{\substack{i > k \\ j}} p_{ij}) + (\delta_2 - \delta_1) \sum_{\substack{i = k \\ j > k}} d_{ij}^1, \tag{2}$$

  wait δ_2 minutes.

- If the train has waited before δ_1 minutes at station k and, at station l,

$$T \sum_{\substack{i \leq l \\ j}} d_{ij}^2 \geq (\delta_2 - \delta_1) (\sum_{\substack{i \leq l \\ j > l}} o_{ij} + \sum_{\substack{k \leq i \leq l \\ j > l}} d_{ij}^1 + \sum_{\substack{i > l \\ j}} p_{ij}) \tag{3}$$

  holds, wait $\delta_2 - \delta_1$ minutes.

In all cases, the algorithm compares the accumulated delay of passengers with source delay δ_1 or δ_2, respectively, who have missed the train because it did not wait for them with the maximum delay that could be caused by waiting δ_1 or δ_2 minutes at the current station. In the first case, waiting δ_1 minutes delays all on time passengers who are currently on the train and possibly all passengers who board the train in the future by δ_1 minutes. The second case reflects the same circumstance for δ_2. In the third case, waiting δ_2 minutes delays the passengers who are already on the train and possibly all passengers getting on the train in the future by $\delta_2 - \delta_1$ minutes.

Theorem 2.3. *ALG 2 is 3-competitive for the Online Delay Management Problem on a Single Train Line with two different source delays.*

Proof. Let k be the station where ALG 2 waits δ_1 minutes and $l \geq k$ the station where it waits $\delta_2 - \delta_1$ minutes. By k^* and $l^* \geq k^*$ denote the stations where OPT waits δ_1 and $\delta_2 - \delta_1$ minutes, respectively.

Case 1: $k < k^*,\ l > l^*$: Comparing the objective values of the two solutions, we get

$$\begin{aligned} D(k,l) =& D(k^*,l^*) - (T-\delta_1) \sum_{\substack{k \leq i < k^* \\ j}} d^1_{ij} + (T-\delta_2) \sum_{\substack{l^* \leq i < l \\ j}} d^2_{ij} \\ &+ \delta_1 \sum_{\substack{i \\ k < j \leq k^*}} o_{ij} \underbrace{-(\delta_2 - \delta_1) \sum_{\substack{i \\ l^* < j \leq l}} o_{ij}}_{\leq 0} \\ &+ (\delta_2 - \delta_1) \sum_{\substack{k \leq i < k^* \\ j > l}} d^1_{ij} \underbrace{-(\delta_2 - \delta_1) \sum_{\substack{i \geq k^* \\ l^* < j \leq l}}^{l} d^1_{ij}}_{\leq 0}. \end{aligned}$$

- Note that

$$(\delta_2 - \delta_1) \sum_{\substack{k \leq i < k^* \\ j > l}} d^1_{ij} - (T - \delta_1) \sum_{\substack{k \leq i < k^* \\ j}} d^1_{ij} \leq 0.$$

- Since Condition (3) was not fulfilled at station $l-1$, we have

$$\begin{aligned} (T-\delta_2) \sum_{\substack{l^* \leq i < l \\ j}} d^2_{ij} &\leq T \sum_{\substack{i \leq l-1 \\ j}} d^2_{ij} \\ &\overset{(3)}{<} (\delta_2 - \delta_1)\Big(\sum_{\substack{i \leq l-1 \\ j > l-1}} o_{ij} + \sum_{\substack{k \leq i \leq l-1 \\ j > l-1}} d^1_{ij} + \sum_{\substack{i > l-1 \\ j}} p_{ij} \Big) \\ &\overset{l^* \leq l-1}{\leq} D(k^*,l^*). \end{aligned}$$

- At station k, Condition (1) was fulfilled which yields

$$\begin{aligned} \delta_1 \sum_{\substack{i \\ k < j \leq k^*}} o_{ij} &\leq \delta_1 \Big(\sum_{\substack{i \leq k \\ j > k}} o_{ij} + \sum_{\substack{i > k \\ j}} p_{ij} \Big) \\ &\overset{(1)}{\leq} T \sum_{\substack{i \leq k \\ j}} d^1_{ij} \\ &\overset{k < k^*}{\leq} D(k^*,l^*) \end{aligned}$$

and, hence, $D(k,l) \leq 3D(k^*,l^*)$.

Case 2: $k > k^*,\ l < l^*$:

$$\begin{aligned} D(k,l) =& D(k^*,l^*) + (T-\delta_1) \sum_{\substack{k^* \le i < k \\ j}} d^1_{ij} \underbrace{-(T-\delta_2) \sum_{\substack{l \le i < l^* \\ j}} d^2_{ij}}_{\le 0} \\ & \underbrace{-\delta_1 \sum_{\substack{i \\ k^* < j \le k}} o_{ij}}_{\le 0} + (\delta_2 - \delta_1) \sum_{\substack{i \\ l < j \le l^*}} o_{ij} \\ & \underbrace{-(\delta_2-\delta_1) \sum_{\substack{k^* \le i < k \\ j > l}} d^1_{ij}}_{\le 0} + (\delta_2 - \delta_1) \sum_{\substack{i \ge k \\ l < j \le l^*}} d^1_{ij}. \end{aligned}$$

- ALG 2 did not wait at station $k-1$ and, hence, Condition (1) was not fulfilled.

$$\begin{aligned} (T-\delta_1) \sum_{\substack{k^* \le i < k \\ j}} d^1_{ij} &< T \sum_{\substack{i \le k-1 \\ j}} d^1_{ij} \\ &< \delta_1 \Big(\sum_{\substack{i \le k-1 \\ j > k-1}} o_{ij} + \sum_{\substack{i > k-1 \\ j}} p_{ij} \Big) \\ &\overset{k^* \le k-1}{\le} D(k^*,l^*). \end{aligned}$$

- ALG 2 waited at station l and, hence, if $k < l$, Condition (3) was fulfilled. If $k = l$, this means that Condition (2) was fulfilled. In both cases, we get

$$\begin{aligned} (\delta_2-\delta_1) \sum_{\substack{i \ge k \\ l < j \le l^*}} d^1_{ij} + (\delta_2-\delta_1) \sum_{\substack{i \\ l < j \le l^*}} o_{ij} &\overset{(3)}{\le} T \sum_{i \le l, j} d^2_{ij} \\ &\overset{l < l^*}{\le} D(k^*,l^*). \end{aligned}$$

We proved, that for this case, ALG 2 is 3-competitive.

Case 3: $k < k^*,\ l < l^*$:

$$\begin{aligned} D(k,l) =& D(k^*,l^*) - (T-\delta_1) \sum_{\substack{k \le i < k* \\ j}} d^1_{ij} \underbrace{-(T-\delta_2) \sum_{\substack{l \le i < l^* \\ j}} d^2_{ij}}_{\le 0} \\ & + \delta_1 \sum_{\substack{k < i \le k^* \\ j}} o_{ij} + (\delta_2 - \delta_1) \sum_{\substack{i \\ l < j \le l^*}} o_{ij} \\ & + (\delta_2-\delta_1) \sum_{\substack{k \le i < k^* \\ j > l}} d^1_{ij} + (\delta_2 - \delta_1) \sum_{\substack{i \ge k^* \\ l < j \le l^*}} d^1_{ij}. \end{aligned}$$

- First, note that

$$(\delta_2 - \delta_1) \sum_{\substack{k \le i < k^* \\ j > l}} d^1_{ij} - (T - \delta_1) \sum_{\substack{k \le i < k* \\ j}} d^1_{ij} \le 0.$$

- If $k < l$, Condition (1) was fulfilled at k, and, hence

$$\begin{aligned} \delta_1 \sum_{\substack{k < i \le k^* \\ j}} o_{ij} &\le \delta_1 \sum_{\substack{i > k \\ j}} p_{ij} \\ &\overset{(1)}{\le} T \sum_{\substack{i \le k \\ j}} d^1_{ij} \\ &\overset{k<k^*}{\le} D(k^*, l^*). \end{aligned}$$

  The train waited at l. Thus,

$$\begin{aligned} (\delta_2 - \delta_1) \sum_{\substack{k \le i < k^* \\ j > l}} d^1_{ij} + (\delta_2 - \delta_1) \sum_{\substack{i \ge k^* \\ l < j \le l^*}} d^1_{ij} &\overset{(3)}{\le} T \sum_{i \le l, j} d^2_{ij} \\ &\overset{l<l^*}{\le} D(k^*, l^*). \end{aligned}$$

- In case that $k = l$,

$$\begin{aligned} &\delta_1 \sum_{\substack{k < i \le k^* \\ j}} o_{ij} + (\delta_2 - \delta_1) \sum_{\substack{i \\ l < j \le l^*}} o_{ij} \\ +(\delta_2 - \delta_1) \sum_{\substack{k \le i < k^* \\ j > l}} d^1_{ij} + (\delta_2 - \delta_1) \sum_{\substack{i \ge k^* \\ l < j \le l^*}} d^1_{ij} &\overset{(2)}{\le} T \sum_{\substack{i \le k \\ j}} d^2_{ij} \\ &\overset{l^*>l}{\le} D(k^*, l^*). \end{aligned}$$

Again, ALG 2 is 3-competitive in this case.

Case 4: $k > k^*, l > l^*$:

$$\begin{aligned} D(k,l) =& D(k^*, l^*) + (T - \delta_1) \sum_{\substack{k^* \le i < k \\ j}} d^1_{ij} + (T - \delta_2) \sum_{\substack{l^* \le i < l \\ j}} d^2_{ij} \\ & \underbrace{-\delta_1 \sum_{\substack{i \\ k^* < j \le k}} o_{ij} - (\delta_2 - \delta_1) \sum_{\substack{i \\ l^* < j \le l}} o_{ij}}_{\le 0} \\ & \underbrace{-(\delta_2 - \delta_1) \sum_{\substack{i \ge k \\ l^* < j \le l}} d^1_{ij} - (\delta_2 - \delta_1) \sum_{\substack{k^* \le i < k \\ j > l}} d^1_{ij}}_{\le 0} \end{aligned}$$

- At $k-1$, Condition (1) was not fulfilled, which yields

$$\begin{aligned}(T-\delta_1)\sum_{\substack{k^*\le i<k\\ j}} d^1_{ij} &\le T\sum_{\substack{i\le k-1\\ j}} d^1_{ij}\\ &< \delta_1\Big(\sum_{\substack{i\le k-1\\ j>k-1}} o_{ij} + \sum_{\substack{i>k-1\\ j}} p_{ij}\Big)\\ &\overset{k^*\le k-1}{\le} D(k^*,l^*).\end{aligned}$$

- At $l-1$, Condition (2) was not fulfilled. Thus,

$$\begin{aligned}(T-\delta_2)\sum_{\substack{l^*\le i<l\\ j}} d^2_{ij} &\le T\sum_{\substack{i\le l-1\\ j}} d^2_{ij}\\ &< (\delta_2-\delta_1)\Big(\sum_{\substack{i\le l-1\\ j>l-1}} o_{ij} + \sum_{\substack{k\le i\le l-1\\ j>l-1}} d^1_{ij} + \sum_{\substack{i>l-1\\ j}} p_{ij}\Big)\\ &\overset{l^*\le l-1}{\le} D(k^*,l^*),\end{aligned}$$

and, hence, ALG 2 is 3-competitive.

So far, we have omitted the cases with $k = k^*$ or $l = l^*$. Their analysis goes along the same lines only that sum of the terms in the summation are equal to zero. Hence, we have proved that ALG 2 is 3-competitive. □

3 Maximizing Profit

In this section, we consider a modification of the delay management problem motivated by the refund system of the German national railway company (*Deutsche Bahn*), which gives passengers a discount of 25% or 50% of the ticket price if they reach their destination with a delay of at least 60 or 120 minutes, respectively.

For that purpose, we consider the delay management problem with one possible source delay described in the introduction with a different objective function. If passengers reach their destination with a delay of at least Δ, they get a certain percentage of their ticket price refunded. So passengers with a delay of at least Δ pay a discounted ticket price of $p > 0$, whereas passengers who reach their destination with a delay less than Δ pay the full ticket price of $\alpha \cdot p$, where $\alpha > 1$ (for the *Deutsche Bahn* example, we have $\alpha \in \{4/3, 2\}$).

We assume $\delta < \Delta \le T$. Hence, if there are passengers with a source delay of δ and we do not wait for them, their delay increases to $T \ge \Delta$ and they get a refund. By waiting for delayed passengers, their delay stays smaller than Δ. But waiting involves a delay of δ for the on time passengers, which causes them to miss their connecting trains and leads to an even higher delay $\ge \Delta$. Note that we neglect on time passengers that still reach their destination with a delay $< \Delta$ even if we delay them by δ, since, in each case, they are not eligible for a refund.

Similarly, we do not consider passengers with source delay δ that will miss their connecting train and reach their destination with delay $\geq \Delta$ even if we wait for them, since they get refunded anyway, independently of our decision. In other words, we only consider those passengers for which the question if they get refunded or not is dependent of decision. Hence, if we wait at station k, the profit amounts to

$$P(k) = \alpha \sum_{i,j \leq k} p \cdot o_{ij} + \sum_{i,j > k} p \cdot o_{ij} + \alpha \sum_{i \geq k, j} p \cdot d_{ij} + \sum_{i < k, j} p \cdot d_{ij}.$$

We get the full profit from the delayed passengers we waited for and from the on time passengers who do not get delayed by us. Hence, if we wait at station k, we get full profit from the delayed passengers who connect at station $i \geq k$ and from the on time passengers who get off at station $j \leq k$. Note that the objective function does not change if we multiply the passengers per trail by p, and, therefore, we can assume that without loss of generality $p = 1$. Also note that, by definition, any online strategy for this problem is at least $\alpha-$competitive.

For the aim of minimizing the refund, there cannot be a competitive online algorithm. Assume that there are two passenger trails, $p_{13} < p_{23}$, and let the p_{13} be delayed. If ALG does not wait, assume that the p_{23} are late. Then ALG has to pay a refund of $(\alpha - 1) \cdot p_{13}$, whereas OPT waits at station 1 and has a cost of 0, so ALG cannot be competitive in this case. Hence, in order to avoid this scenario, ALG has to wait at station 1. Now assume that the passengers p_{23} are on time. Then ALG has to pay a refund of $(\alpha - 1)p_{23}$, whereas OPT does not wait at all which leads to a cost of $(\alpha - 1)p_{13}$. So we get ALG/OPT $= p_{23}/p_{13}$ which approaches ∞ as p_{23} goes to ∞.

Thus, we consider the Max Profit Online Delay Management Problem on a Single Train Line. We start with the case of 3 stations and give a lower bound on the competitive ratio of any deterministic online algorithm for the problem.

Theorem 3.1. *No deterministic online algorithm on a single train line with at least 3 stations can have a competitive ratio less than* $\frac{2\alpha+1}{\alpha+2}$ *when maximizing the profit.*

Proof. Let $p_{13} = 2$, $d_{13} = 1$, and $p_{23} = 1$. If ALG waits at station 1, then p_{23} is on time and OPT does not wait at all, so we get OPT/ALG $= (2\alpha + 1)/(\alpha + 2)$. If ALG does not wait, p_{23} is late and, hence, OPT waits in station 1. This also yields OPT/ALG $= (2\alpha + 1)/(\alpha + 2)$. □

We develop an online algorithm for the case of 3 stations. It holds

$$P(1) = o_{12} + o_{13} + o_{23} + \alpha d_{12} + \alpha d_{13} + \alpha d_{23}.$$
$$P(2) = \alpha o_{12} + o_{13} + o_{23} + d_{12} + d_{13} + \alpha d_{23}.$$
$$P(3) = \alpha o_{12} + \alpha o_{13} + \alpha o_{23} + d_{12} + d_{13} + d_{23}.$$

In station 1, we already know whether $P(1) \leq P(2)$ since the only unknown variables are o_{23} and d_{23}, which have the same coefficients in both objective functions. If $P(1) \leq P(2)$, the best strategy is not to wait in station 1. Then, in station 2, when all information is available, one can choose between $P(2)$ and $P(3)$ to obtain the optimal solution.

Hence, assume that, in station 1, $P(1) > P(2)$ or equivalently $d_{12} + d_{13} > o_{12}$. If it is already clear at this point that $P(3)$ cannot be the optimal solution, we can solve the problem to optimality by waiting at station 1. We know that

$$P(1) \geq P(3) \Leftrightarrow d_{12} + d_{13} + d_{23} \geq o_{12} + o_{13} + o_{23}$$

and, hence, we can conclude that $P(1) \geq P(3)$ (without knowing o_{23} and d_{23}) if

$$d_{12} + d_{13} \geq o_{12} + o_{13} + p_{23}.$$

In this case, we know that the optimal solution is $P(1)$.

Hence, it remains to consider the case that $P(1) > P(2)$ and $d_{12} + d_{13} < o_{12} + o_{13} + p_{23}$, in which case we need to decide whether the train should wait or not.

ALG 3: If $P(1) \leq P(2)$, do not wait at station 1.
If $P(1) > P(2)$, or equivalently $d_{12} + d_{13} > o_{12}$, wait at station 1 if

- $o_{12} + o_{13} + p_{23} \leq \beta(d_{12} + d_{13})$ where $\beta \geq 1$

If the train did not wait at station 1, it chooses the better among the solutions $P(2)$ and $P(3)$ at station 2.

Theorem 3.2. *ALG 3 is* $\min\{\alpha, 2 + \frac{1}{\alpha}\}$*-competitive for* $\beta = 2$.

Proof. In case that $P(1) \leq P(2)$, ALG 3 achieves the optimal objective value. It remains to consider the case $P(1) > P(2)$. If ALG 3 waits at station 1, $P(2)$ cannot be optimal by definition of the algorithm. In case that $P(3)$ is the optimal objective value, we obtain

$$\begin{aligned}\frac{\text{OPT}}{\text{ALG 3}} = \frac{P(3)}{P(1)} &= \frac{\alpha o_{12} + \alpha o_{13} + \alpha o_{23} + d_{12} + d_{13} + d_{23}}{o_{12} + o_{13} + o_{23} + \alpha d_{12} + \alpha d_{13} + \alpha d_{23}} \\ &\leq \frac{\beta\alpha(d_{12} + d_{13}) + d_{12} + d_{13}}{\alpha(d_{12} + d_{13})} \\ &= \beta + \frac{1}{\alpha}.\end{aligned}$$

If ALG 3 does not wait at station 1, it achieves the value $\max\{P(2), P(3)\}$. If $P(1)$ is the optimal solution, and $P(3) \geq P(2) \Leftrightarrow o_{13} + o_{23} \geq d_{23}$,

we have

$$\begin{aligned}\frac{\text{OPT}}{\text{ALG 3}} = \frac{P(1)}{P(3)} &= \frac{o_{12}+o_{13}+o_{23}+\alpha d_{12}+\alpha d_{13}+\alpha d_{23}}{\alpha o_{12}+\alpha o_{13}+\alpha o_{23}+d_{12}+d_{13}+d_{23}} \\ &\leq \frac{1}{\alpha}+\frac{\alpha(d_{12}+d_{13}+d_{23})}{\alpha o_{12}+\alpha o_{13}+\alpha o_{23}+d_{12}+d_{13}+d_{23}} \\ &\leq \frac{1}{\alpha}+\frac{\alpha(\frac{1}{\beta}(o_{12}+o_{13}+o_{23}+d_{23})+d_{23})}{\alpha o_{12}+\alpha o_{13}+\alpha o_{23}+d_{12}+d_{13}+d_{23}} \\ &\leq \frac{1}{\alpha}+\frac{\alpha(\frac{1}{\beta}(o_{12}+o_{13}+o_{23}+o_{12}+o_{13})+o_{12}+o_{13})}{\alpha o_{12}+\alpha o_{13}+\alpha o_{23}+d_{12}+d_{13}+d_{23}} \\ &\leq \frac{1}{\alpha}+\frac{3}{\beta}.\end{aligned}$$

If $P(1)$ is optimal and $P(3) < P(2)$, or equivalently $o_{13}+o_{23} < d_{23}$,

$$\begin{aligned}\frac{\text{OPT}}{\text{ALG 3}} = \frac{P(1)}{P(2)} &= \frac{o_{12}+o_{13}+o_{23}+\alpha d_{12}+\alpha d_{13}+\alpha d_{23}}{\alpha o_{12}+o_{13}+o_{23}+d_{12}+d_{13}+\alpha d_{23}} \\ &\leq 1+\frac{\alpha d_{12}+\alpha d_{13}}{\alpha o_{12}+o_{13}+o_{23}+d_{12}+d_{13}+\alpha d_{23}} \\ &\leq 1+\frac{\frac{\alpha}{\beta}(o_{12}+o_{13}+o_{23}+d_{23})}{\alpha o_{12}+o_{13}+o_{23}+d_{12}+d_{13}+\alpha d_{23}} \\ &\leq 1+\frac{\frac{\alpha}{\beta}(o_{12}+d_{23}+d_{23})}{\alpha o_{12}+o_{13}+o_{23}+d_{12}+d_{13}+\alpha d_{23}} \\ &\leq 1+\frac{2}{\beta}.\end{aligned}$$

For $\beta = 2$ we get an $2 + 1/\alpha-$competitive algorithm. Since any algorithm is α-competitive, ALG 3 is $\min\{\alpha, 2+\frac{1}{\alpha}\}$- competitive. □

Theorem 3.3. *For arbitrarily many stations, there cannot be a deterministic online algorithm with competitive ratio smaller than* $\frac{\alpha+1}{2}$.

Proof. Let there be n stations and let there be $n-1$ passenger trails $p_{i,i+1}, i = 1, \ldots, n-1$ consisting of 1 passenger each. Let all passengers be late until the online algorithm ALG decides to wait. From that time on, let all passengers be on time. Let k be the station where the online algorithm waits.

We have $\text{ALG} = (n-2)+\alpha$ for the online algorithm. Depending on k, the optimal solution is to wait at station 1 or not at all. We have $\text{OPT} = \max\{k, n-1-k\}\alpha + \min\{k, n-1-k\}$, so

$$\begin{aligned}\frac{\text{OPT}}{\text{ALG}} &= \frac{\max\{k, n-1-k\}\alpha + \min\{k, n-1-k\}}{(n-2)+\alpha} \\ &\geq \frac{\left\lceil \frac{n-1}{2} \right\rceil \alpha + \left\lfloor \frac{n-1}{2} \right\rfloor}{(n-2)+\alpha}.\end{aligned}$$

The quotient is increasing in n and approaches $\frac{\alpha+1}{2}$ as n goes to ∞. □

Since there cannot be any deterministic competitive online algorithm, we consider randomized algorithms. In the above proof, we used instances with $p_{i,i+1} = 1$ for all $i = 1, n-1$ such that all passengers are delayed until a certain station k and after that, they are on time. Depending on k, the optimal strategy is to wait at station 1 or to wait at station n. In fact, we can show that the algorithm that chooses between these options randomly, has a competitive ratio of 2:

ALG 4: With probability 1/2 wait at station 1 and with probability 1/2 wait at station n.

Theorem 3.4. *The competitive ratio of ALG 4 is* $\frac{2\alpha}{1+\alpha}$.

Proof. For a given instance σ, let OPT wait at station k^*. Then,

$$\begin{aligned}\frac{\text{OPT}}{\text{E[ALG 4}(\sigma)]} &\leq \frac{\alpha(\sum_{i,j} d_{ij} + \sum_{i,j} o_{ij})}{\frac{1}{2}(\alpha \sum_{i,j} d_{ij} + \sum_{i,j} o_{ij}) + \frac{1}{2}(\alpha \sum_{i,j} o_{ij} + \sum_{i,j} d_{ij})} \\ &= \frac{\alpha(\sum_{i,j} d_{ij} + \sum_{i,j} o_{ij})}{\sum_{i,j} d_{ij}(\frac{1}{2}\alpha + \frac{1}{2}) + \sum_{i,j} o_{ij}(\frac{1}{2} + \frac{1}{2}\alpha)} \\ &= \frac{2\alpha}{1+\alpha}.\end{aligned}$$

Note that the inequality is an equality for instances such that, first, all passengers are on time, and, then, all passengers are delayed. □

Using Yao's principle [5] we can prove the following lower bound for randomized algorithms solving the problem.

Theorem 3.5. *No randomized algorithm for the problem can have a competitive ratio smaller than* $\frac{1+3\alpha}{2+2\alpha}$.

Proof. Consider the following scenario: There are $n+1$ stations and n passenger trails such that each consists of one passenger going from station i to $i+1$. The first k passenger trails are delayed and the last $n-k$ trails are on time. Let k be uniformly distributed in $\{0, \ldots, n\}$. In this scenario, the optimal solution is either to wait in station 1 or in $n+1$, depending on whether there are more on time or late passengers.

Let algorithm A_i be the algorithm that does not wait until station i and then, if there already has been an on time passenger, it does not wait at all and else, it waits there. The expected value of A_i is

$$\begin{aligned}E[A_i] &= \sum_{k=0}^{i} \frac{1}{n+1}(\alpha(n-k)+k) + \sum_{k=i+1}^{n} \frac{1}{n+1}(\alpha(k-i)+n-k+i) \\ &= \frac{-2(-1+\alpha)i + n(-1+n+\alpha(3+n))}{2(1+n)}.\end{aligned}$$

The derivative with respect to i is $\frac{1-\alpha}{1+n}$, which is nonpositive since $\alpha \geq 1$. Hence, the term is nonincreasing in i and, therefore, maximal for $i = 0$. Thus, A_0 is the best algorithm for this randomized online

problem and it suffices to consider A_0 when applying Yao's principle. If n is even, we have

$$\begin{aligned} E[\mathrm{OPT}] &= \frac{1}{n+1}(\sum_{k=0}^{\frac{n}{2}} k + (n-k)\alpha + \sum_{k=\frac{n}{2}+1}^{n} k\alpha + n - k) \\ &= \frac{n(n+\alpha(4+3n))}{4(1+n)}. \end{aligned}$$

Hence, we get the quotient

$$\begin{aligned} \frac{E[\mathrm{OPT}]}{E[A_0]} &= \frac{n(n+\alpha(4+3n))2(1+n)}{4(1+n)(-2(-1+\alpha)0+n(-1+n+\alpha(3+n)))} \\ &= \frac{n+\alpha(4+3n)}{2(-1+n+\alpha(3+n))}. \end{aligned}$$

For n going to ∞, this approaches $\frac{1+3\alpha}{2+2\alpha}$.

For odd n, we get

$$\begin{aligned} E[\mathrm{OPT}] &= \frac{1}{n+1}(\sum_{k=0}^{\frac{n}{2}-\frac{1}{2}} k + (n-k)\alpha + \sum_{k=\frac{n}{2}+\frac{1}{2}}^{n} k\alpha + n - k) \\ &= \frac{1}{n+1}((1-\alpha)\sum_{k=0}^{\frac{n}{2}-\frac{1}{2}} k + \sum_{k=0}^{\frac{n}{2}-\frac{1}{2}} n\alpha \\ &+ (\alpha-1)\sum_{k=\frac{n}{2}+\frac{1}{2}}^{n} k + \sum_{k=\frac{n}{2}+\frac{1}{2}}^{n} n) \\ &= \frac{1}{4}(-1+\alpha+n+3\alpha n) \end{aligned}$$

and, therefore, a quotient of

$$\frac{(-1+\alpha+n+3\alpha n)2(1+n)}{4n(-1+n+\alpha(3+n))}.$$

Its derivative with respect to n is

$$\frac{(-1+\alpha)(-1+n)(-1+n+\alpha(3+5n))}{2n^2(-1+n+\alpha(3+n))^2},$$

which is nonnegative, so the quotient is increasing. It approaches $\frac{2+6\alpha}{4+4\alpha}$ as n approaches ∞. □

References

[1] B. Adenso-Díaz, J. Tuya, M. J. Suárez-Cabal, and M. Goitia-Fuertes. DSS for rescheduling of railway services under unplanned events. In Mora et al. [9], pages 72–85.

[2] L. Anderegg, P. Penna, and P. Widmayer. Online train disposition: To wait or not to wait? *Electronic Notes in Theoretical Computer Science*, 66(6):32–41, 2002.

[3] A. Berger, R. Hoffmann, U. Lorenz, and S. Stiller. TOPSU - RDM a simulation platform for online railway delay management. In *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools)*, pages 1–8, 2008.

[4] C. Biederbick and L. Suhl. Decision support tools for customer-oriented dispatching. In *Algorithmic Methods for Railway Optimization*, volume 4359 of *LNCS*, pages 171–183, 2004.

[5] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[6] M. Gatto, B. Glaus, R. Jacob, L. Peeters, and P. Widmayer. Railway delay management: Exploring its algorithmic complexity. In *Proceedings of the 9th Scandinavian Workshop on Algorithm Theory (SWAT)*, volume 3111 of *LNCS*, pages 199–211, 2004.

[7] M. Gatto, R. Jacob, L. Peeters, and A. Schöbel. The computational complexity of delay management. In *Proceedings of the 31st International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 3787 of *LNCS*, pages 227–238, 2005.

[8] M. Gatto, R. Jacob, L. Peeters, and P. Widmayer. Online delay management on a single train line. In *Proceedings of the 4th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS)*, volume 4359 of *LNCS*, pages 306–320, 2004.

[9] M. Mora, G. A. Forgionne, and J. N. D. Gupta, editors. *Decision Making Support Systems: Achievements, Trends and Challenges for the New Decade*. IGI Global, 2003.

[10] M. J. Osborne. *An Introduction to Game Theory*. Oxford University Press, 2004.

[11] A. Schöbel. A model for the delay management problem based on mixed-integer-programming. *Electronic Notes in Theoretical Computer Science*, 50(1):1–10, 2001.

[12] A. Schöbel. *Optimization in Public Transportation*. Springer, 2006.