Online train disposition: to wait or not to wait? *

Luzi Anderegg† Paolo Penna‡ Peter Widmayer†

June 15, 2009

Abstract

We deal with an online problem arising from bus/tram/train disposition problems. In particular, we look at the case in which the delay is *unknown* and the vehicle can only wait in a station so as to minimize the passengers' waiting time.

We present deterministic polynomial-time optimal algorithms and matching lower bounds for several problem versions. In addition, all lower bounds also apply to randomized algorithms, thus implying that using randomization does not help.

1 The setting

While many of the optimization problems encountered in transportation have already been studied in the early days of operations research [Dantzig and Fulkerson, 1954], [Bertossi, Carraresi, and Gallo, 1987], [Brucker, Hurink, and Rolfes, 1999] and have even stimulated the development of the field, this is not the case for *disposition* problems. Disposition (also known as operations control) deals with the *real time reaction* against the negative effects of *unexpected events*. For railways, the goal is to maintain high service quality in spite of events such as delays due to disturbances. Problems of this sort have been attacked mostly by computer simulations [Heimburger, Herzenberg, and Wilson, 1999], [Zhu and Schnieder, 2001] (see also [Mansilla, 2001] for a survey). In this paper, we pursue a different approach: we aim at an understanding of the fundamental algorithmic nature of these problems. In particular, we will look at *worst case* analysis of algorithms that must work with

*A preliminary version of this paper appeared in [Anderegg, Penna, and Widmayer, 2002]

†Institute for Theoretical Computer Science, ETH Zentrum, CH-8092 Zürich, Switzerland, E-mail: {lastname}@inf.ethz.ch.

‡Dipartimento di Informatica ed Applicazioni "R.M. Capocelli", Università di Salerno, via S. Allende 2, I-84081 Baronissi (SA), Italy. E-mail: penna@dia.unisa.it.

partial information (e.g., we know that a vehicle has been delayed, but we do not exactly know by how many time units). To this aim, we will show how these questions can be treated as an *online problem* [Borodin and El-Yaniv, 1998], [Fiat and Woeginger, 1998]. Then, we will characterize the performance of algorithms depending on several factors like (i) number of vehicles, (ii) whether the algorithm has some estimation of the delays, (iii) whether it can use randomization, etc. This problem is closely related to the *delay management problem*: a schedule and a delay for one or more vehicles is given and good decisions (e.g., to wait or to depart) for the consecutive trains must be taken. In contrast to our problem, most delay management studies assume that the exact delay is known to the algorithm. [Suhl, Biederbick, and Kliewer, 2001] and [Adenso-Díaz, Gonzáles, and Gonzáles-Torre, 1999] use simulation systems for analyzing the delay management problem. Other authors, such as [Schöbel, 2001] (see also [Ginkel, 2001] for a survey), formulate an optimization problem in which the goal is to minimize the total waiting time subject to different constraints (slack times available at stations and tracks, connections can be dropped, etc.). For the same objective function, [Gatto, Glaus, Jacob, Peeters, and Widmayer, 2004] described polynomial time algorithms for special cases, such as a limited number of transfers, or a railway network with a path topology. In a follow-up paper [Gatto, Jacob, Peeters, and Schöbel, 2005], a more general variant of the delay management problem was shown to be NP-complete both with and without slack times (or buffer times) in the timetable. More work on the delay management problem has been done in [Ginkel and Schöbel, 2007], [Cicerone, D'Angelo, Di Stefano, Frigioni, and Navarra, 2008]. The online delay management problem has been addressed further in [Gatto, Jacob, Peeters, and Widmayer, 2007] where the authors again considered also the case of a single line. The main difference between the model in that paper and the one here is on the delay incurred by the passengers. In [Gatto, Jacob, Peeters, and Widmayer, 2007], passengers' delay is exclusively due to the fact that a train at a station does not wait and thus a connection is missed. In our model, instead, passengers arrive at the station and we consider how much each of them should wait before their train leaves.

1.1 The disposition problem

Consider the following scenario from high-frequency bus (or tram or train) systems: we are given a station with $r > 0$ passengers arriving at each time unit on average (i.e., r is the *arrival rate of passengers at the station*). Buses reach the station regularly every t time units if no delay occurs. Whenever a bus reaches the station, it picks up all waiting passengers (i.e., the seating capacities

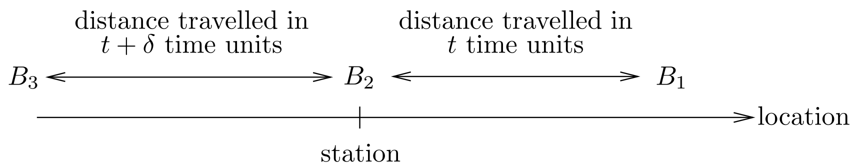

Figure 1: The case of three buses.

of the bus are not our concern). We assume that picking up the passengers is instantaneous, i. e., requires no time. This implies that the overall passenger *waiting time at the station* (sum of all individual waiting times) is $r \cdot t^2/2$ per t time units between any two consecutive buses. Now consider the case in which one bus is currently at some station and the next bus is delayed by some amount of $\delta > 0$ time units. Assume the only action we can take is to make a bus wait in a station (this is sometimes referred to as *holding* [O'Dell and Wilson, 1999]). The problem now is to decide how long the bus in the station should wait, if at all.

A convenient way of looking at this problem is to consider a snapshot of only three buses B_3, B_2 and B_1 as in Figure 1 that are travelling from left to right (we will drop the limitation to three buses later and consider more buses). Then, from the point of view of the passengers in the station, making B_2 wait for w time units is equivalent to "shift" B_2 leftwards by the distance that a bus travels in w time units. The overall waiting time (denoted as cost) can be computed according to which bus passengers get in (Figure 2 shows the case $w = 0$), as

$$cost = \underbrace{r(t+w)^2/2}_{B_2} + \underbrace{r(t+\delta-w)^2/2}_{B_3}. \tag{1}$$

Clearly, *knowing* δ, the best choice (i.e., the choice that minimizes the value in Equation 1) is $w = \delta/2$. However, for our problem of interest, we only know that B_3 is delayed (e. g., because of a traffic jam), but we do not exactly know δ. In this case, should B_2 leave immediately or wait for a while? In the latter case, how much should it wait for?

1.2 An algorithmic perspective

We view this question as an *online* problem in which we have to choose a good w *without knowing* δ (ideally, w should be good for *all* possible delays δ). Because the value of t is purely a matter of scaling time units, we will assume from now on that $t = 1$. Then, the actual waiting time, denoted as cost, is $r(1+w)^2/2 + r(1+\delta-w)^2/2$, and the optimum waiting time is $r(1+\delta/2)^2$. For the

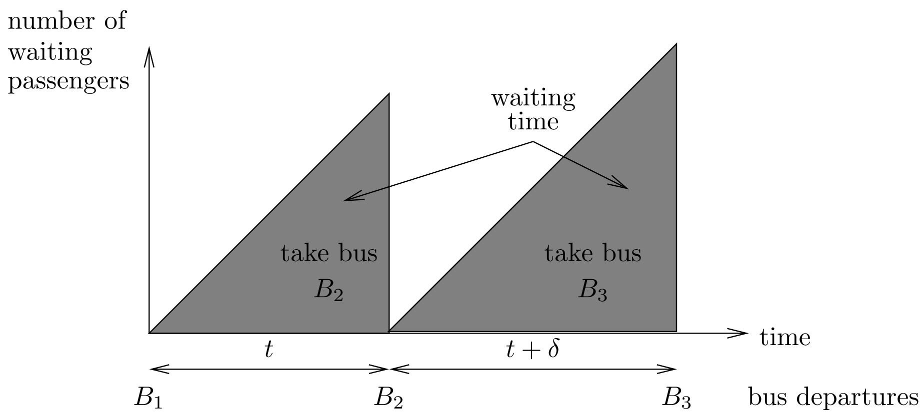

Figure 2: Passengers waiting time when bus B_3 has a delay δ.

competitive ratio, that is, the ratio between the actual and the optimum waiting times, the arrival rate r cancels, and therefore we assume for simplicity of the presentation from now on that $r = 2$. That is, we get the actual waiting time $\mathsf{cost}(w, \delta) = (1 + w)^2 + (1 + \delta - w)^2$, and the competitive ratio within the interval of arrival times from bus B_1's departure to B_3's arrival is

$$\rho(w, \delta) = \frac{\mathsf{cost}(w, \delta)}{\mathsf{opt}(\delta)} = \frac{(1 + w)^2 + (1 + \delta - w)^2}{2(1 + \delta/2)^2}. \tag{2}$$

We are interested in online algorithms that minimize the above ratio without knowing δ, that is, algorithms that decide w in such a way that $\max_{\delta \geq 0} \rho(w, \delta)$ is as small as possible. This is clearly equivalent to find $\min_{w \geq 0} \max_{\delta \geq 0} \rho(w, \delta)$. Note that if we choose a waiting time $w > 1$ and the adversary chooses not to delay bus B_3 then the situation is as if there is no bus B_2. The competitive ratio becomes strictly greater than two. As choosing not to wait gives a competitive ratio of less than two (compare also Section 2 and the proof of Theorem 5) we only need to consider w in $[0, 1]$ respectively find $\min_{w \in [0,1]} \max_{\delta \geq 0} \rho(w, \delta)$.

We consider two versions of this problem: (a) the *unbounded* case in which δ can be any positive integer; (b) the *bounded* case in which $\delta \leq \Delta$, where Δ is a positive integer *known to the algorithm*, that is, an upper bound on the maximum delay that can occur.

Remark 1 (Competitive measures) Notice that we are adopting the definition of *strictly* c-competitive algorithms. However, for our problem(s) this is equivalent to that of c-competitiveness. Indeed, we have assumed $t = 1$ only for the sake of simplicity, but *we do not consider t constant* (or as a parameter of the problem), since we want to derive algorithms that perform well *for any t*

known to the algorithm. Under this assumption, we can always construct an instance whose cost is arbitrarily large by increasing t. This allows to apply any lower bound on the strict competitiveness also to the (weaker) definition of c-competitiveness. On the other hand, if t is a *constant of the problem*, then the definition of c-competitiveness is meaningless: the worst solution we can get has cost at most $\mathsf{opt} + \Delta r = \mathsf{opt} + O(1)$. This would imply a 1-competitive algorithm, *regardless of what we do*, while the (strictly) competitive ratio tells us whether the strategy is good or not.

1.3 Our contribution

We consider the above mentioned online problem and its natural extension in which a set of $n+2$ buses (instead of three) is given: bus B_1 already left the station, bus B_{n+2} has a delay δ and we have to decide the waiting time w_i for each B_i, for $i = 2, \ldots, n+1$. This provides a family of basic disposition problems that capture some fundamental aspects of the real situations. For these problems, we completely characterize the competitive ratio of both deterministic and randomized algorithms, depending on n and Δ. In particular, we prove the *tight* bounds shown in Figure 3.

Problem version	**Lower bound**	**Upper bound**
Unbounded delays	$n+1$	$n+1$
Bounded delays ($\delta \leq \Delta$)	$1 + n\left(\frac{\Delta}{2+2n+\Delta}\right)^2$	$1 + n\left(\frac{\Delta}{2+2n+\Delta}\right)^2$

Figure 3: Our results on the competitive ratio of online algorithms. All upper bounds are obtained via deterministic algorithms, while lower bounds also apply to randomized ones.

Interestingly, all the upper bounds are given via *deterministic algorithms*, while the lower bounds also apply to *randomized* ones. Indeed, we show that the competitive ratio attained by our deterministic algorithms cannot be improved even when considering randomized algorithms against an *oblivious* adversary [Borodin and El-Yaniv, 1998], [Fiat and Woeginger, 1998]. In other words, randomization is useless for our disposition problems.

Paper organization. For the sake of clarity, we first present our results for the case $n = 1$. In particular, Sections 2 and 3 deal with the unbounded and the bounded case, respectively. We then extend the results to the case $n > 1$ in Section 4. Finally, in Section 5 we discuss further extensions and open questions.

2 Unbounded delays

We first observe that two strategies are always possible:

No wait. In this case $w = 0$ and $\rho(w, \delta) = \frac{1+(1+\delta)^2}{2(1+\delta/2)^2}$. For $\delta \to \infty$, this ratio tends to 2 from below.

Wait "forever". This means that B_2 waits until B_3 arrives in the station. Then, $\rho(w, \delta) = 2$.

The above two strategies seem quite inefficient. Indeed, a better choice might be a compromise of them (i.e. wait, but not too much). Unfortunately, the following result shows that finding such a compromise is impossible:

Theorem 2 *No (randomized) algorithm can be better than 2-competitive in the case of unbounded delays.*

Proof. Every (randomized) algorithm ALG chooses an upper bound $W \in \mathcal{R}^+ \cup \{\infty\}$ on the waiting time according to some probability distribution independent of δ (this value is chosen by the adversary and is not known to the algorithm). For every δ, the waiting time is $\min(W, 1 + \delta)$ because we never wait more than the time the delayed bus arrives at the station (at that point δ is disclosed to the algorithm and the optimal decision is to have the bus to leave). In particular $W = \infty$ corresponds to the "wait forever" strategy meaning that, for every δ, the waiting time w is equal to $1 + \delta$. Observe that the "no wait" strategy ($W = 0$) strictly dominates the "wait forever" strategy ($W = \infty$) because $\rho(0, \delta) = \frac{1+(1+\delta)^2}{2(1+\delta/2)^2} = \frac{2+\delta^2+2\delta}{2+\delta^2/2+2\delta} \leq 2 = \rho(1 + \delta, \delta)$, for every $\delta \geq 0$ (see Equation 2). Therefore, for every ALG that chooses $W = \infty$ (the "wait forever" strategy) with nonzero probability, there is another algorithm ALG$'$ which has the same or a better competitive ratio and that chooses $W = \infty$ with probability zero.

We can thus focus on algorithms that choose always a *finite* upper bound W on the waiting time according to some probability distribution. This implies that, for any $p \in (0, 1]$, there exists $\overline{w}$ such that $Pr[W \leq \overline{w}] \geq 1 - p$. Since $\rho(w, \delta)$ is decreasing for $w \in [0, \delta/2]$ (see Equation 2) we have that $\rho(W, \delta) \geq \rho(\overline{w}, \delta)$ for all $W \leq \overline{w}$ and $\delta \geq 2\overline{w}$. Therefore, the competitive ratio is at least $(1 - p) \cdot \rho(\overline{w}, \delta)$ for every $\delta \geq 2\overline{w}$. Since p can be arbitrarily small and since the adversary can choose δ arbitrarily large to make $\rho(\overline{w}, \delta)$ close to 2 (note that, for any fixed $\overline{w}$, $\rho(\overline{w}, \cdot)$ tends to 2 for $\delta \to \infty$), the lower bound $(1 - p) \cdot \rho(\overline{w}, \delta)$ on the competitive ratio can be made arbitrarily close to 2. Hence the theorem follows. □

Although the above theorem implies that both strategies above are optimal for large delays, it is clear that "no wait" is always better than "wait forever". Moreover, the former performs quite well whenever δ is small. In the subsequent section we investigate this version of the problem.

3 Bounded delays

In this section we consider the version of the problem in which $\delta \leq \Delta$, where Δ is a positive constant *known to the algorithm*. The purpose of this is twofold: on the one hand we want to study whether this additional information allows for improved competitive ratios; on the other hand, we are interested in finding tight bounds that show how fast the competitive ratio tends to 2 as Δ increases. The "no wait" strategy provides a first upper bound. However, the reader can easily check that choosing $w = \Delta/2$ gives already an improvement. In the next section we give a tight bound for deterministic algorithms.

3.1 Deterministic algorithms

Our algorithm DET should choose a good value of w based solely on the information that $\delta \leq \Delta$. To this aim, we first restrict ourselves to a *weaker adversary* that chooses only $\delta = 0$ or $\delta = \Delta$. Therefore, our goal will become

$$\min_{w} \max\{\rho(w,0), \rho(w,\Delta)\}. \tag{3}$$

In order to determine the best value for w according to Equation 3, we look for which values of w the adversary would give us $\delta = 0$, that is $\rho(w,0) \geq \rho(w,\Delta)$. The latter condition is equivalent to

$$\frac{(1+w)^2 + (1-w)^2}{2} \geq \frac{(1+w)^2 + (1+\Delta-w)^2}{2(1+\Delta/2)^2},$$

which corresponds to $w \geq \Delta/(4+\Delta) =: w_0(\Delta)$. Since $\rho(w,\Delta)$ is monotonically decreasing in $[0, w_0(\Delta)]$ and $\rho(w,0)$ is monotonically increasing in $[w_0(\Delta), \Delta]$, we have (see also Figure 4)

$$\min_{w} \max\{\rho(w,0), \rho(w,\Delta)\} = \rho(w_0(\Delta), 0) = 1 + \left(\frac{\Delta}{4+\Delta}\right)^2. \tag{4}$$

The following lemma is used to show that DET performs well also against an adversary choosing *any* $\delta \in [0, \Delta]$.

Lemma 3 *For any* $w \geq 0$, $\max_{0 \leq \delta \leq \Delta} \rho(w, \delta) \leq \max_{\delta \in \{0, \Delta\}} \rho(w, \delta)$.

Proof. We distinguish the two cases $w \geq w_0(\Delta)$ and $w < w_0(\Delta)$. For $w \geq w_0(\Delta)$, we show that $\rho(w, \delta) \leq \rho(w, 0)$ whereas for $w < w_0(\Delta)$, we show that $\rho(w, \delta) \leq \rho(w, \Delta)$. So, let us assume that $w \geq w_0(\Delta)$. Then

$$\begin{aligned}
\rho(w,\delta) - \rho(w,0) &= -\left(\frac{\delta(1+w)(w(\delta+4)-\delta))}{(2+\delta)^2}\right) \\
&\leq -\left(\frac{\delta(1+w)(w_0(\Delta)(\delta+4)-\delta)}{(2+\delta)^2}\right) \\
&= -\left(\frac{\delta(1+w)(\delta(w_0(\Delta)-1)+4w_0(\Delta))}{(2+\delta)^2}\right) \\
&\leq -\left(\frac{\delta(1+w)(\Delta(\frac{\Delta}{\Delta+4}-1)+4\frac{\Delta}{\Delta+4})}{(2+\delta)^2}\right) = 0
\end{aligned}$$

Hence, in the first case $\rho(w, \delta)$ is maximized for δ equal zero.

For the case $w < w_0(\Delta)$, we consider the difference $\rho(w, \delta) - \rho(w, \Delta)$. Since

$$\frac{d^2}{dw^2}[\rho(w,\delta) - \rho(w,\Delta)] = \frac{8}{(2+\delta)^2} - \frac{8}{(2+\Delta)^2} \geq 0 \qquad (\text{for } \delta \leq \Delta),$$

the difference is convex between 0 and Δ with respect to w. Hence over the region $w \in [0, w_0(\Delta)]$ the maximum of $\rho(w, \delta) - \rho(w, \Delta)$ must be either at $w = 0$ or at $w = w_0(\Delta)$. But

$$\rho(0,\delta) - \rho(0,\Delta) = \frac{4(\delta-\Delta)(\delta+\Delta+\delta\Delta)}{(2+\delta)^2(2+\Delta)^2} \leq 0$$

and

$$\rho(w_0(\Delta),\delta) - \rho(w_0(\Delta),\Delta) = \frac{8\delta(\delta-\Delta)(2+\Delta)}{(2+\delta)^2(4+\Delta)^2} \leq 0.$$

Hence, in the second case $\rho(w, \delta)$ is maximized for δ equal Δ. □

Because of the above lemma and the definition of $w_0(\Delta)$, we obtain the following:

Theorem 4 *No deterministic algorithm can be strictly better than* $1 + w_0(\Delta)^2$ *competitive, where* $w_0(\Delta) = \Delta/(4 + \Delta)$. *Therefore,* Det *is optimal for any* $\Delta \geq 0$.

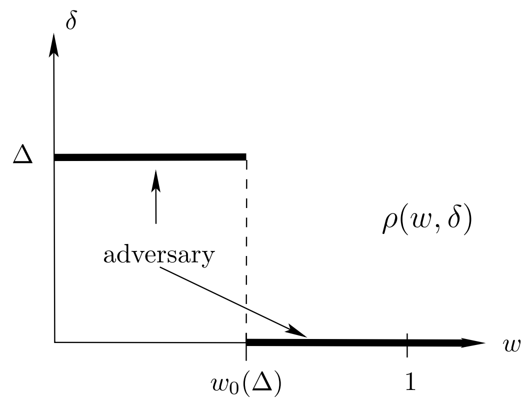

Figure 4: The worst cases for the deterministic algorithm.

As expected, this bound tends to 2 when Δ goes to infinity (which corresponds to the case of unbounded delays).

3.2 Lower bound for randomized algorithms

In this section, we show that no randomized algorithm RAND can achieve an expected competitive ratio smaller than the competitive ratio of the deterministic algorithm DET.

Theorem 5 *For any $\Delta > 0$, no randomized algorithm* RAND *can be better than* DET.

Proof. Suppose w is chosen randomly with corresponding random variable W. Then, for any W, the adversary chooses δ such that $E[\rho(W, \delta)]$ is maximized. Because $\rho(w, \delta)$ is a convex function in w (see Equation 2), we can apply Jensen's inequality [Jensen, 1906], and reduce the randomized case to the deterministic one. In particular, Jensen's inequality implies

$$\max_{\delta} E[\rho(W, \delta)] \geq \max_{\delta} \rho(E[W], \delta).$$

The latter quantity is the competitive ratio of the deterministic algorithm choosing $w = E[W]$. Hence, the result is implied by Theorem 4. $\square$

4 Many buses

Consider a set of $n+2$ buses $\{B_1, \ldots, B_{n+2}\}$ such that: (i) B_1 already left the station, (ii) B_{n+2} has been delayed by δ, and (iii) the set $\{B_2, \ldots, B_{n+1}\}$ corresponds to the *control set* of n buses that we can delay in order to minimize the overall waiting time. Let $\vec{w} = (w_2, w_3, \ldots, w_{n+1})$ represent such waiting times, i.e., bus B_i is delayed by w_i, $i = 2, \ldots, n+1$. The cost is clearly

$$\mathsf{cost}(\vec{w}, \delta, n) = \underbrace{(1+w_2)^2}_{B_2} + \sum_{i=2}^{n} \underbrace{(1+w_{i+1}-w_i)^2}_{B_{i+1}} + \underbrace{(1+\delta-w_{n+1})^2}_{B_{n+2}}. \tag{5}$$

As we do assume that B_i always precedes B_{i+1}, for $i = 1, \ldots, n+1$, we restrict ourselves to those $\vec{w}$ for which $w_i \leq w_{i+1}$, $i = 2, \ldots, n$. In fact, we next show that without loss of generality it is enough to consider certain "balanced" waiting times:

Definition 6 (Balanced vector) *Let the* balanced vector $\vec{u}(x)$ *be the vector assigning waiting time* $w_i = (i-1)x/n$ *to bus* B_i *for* $i = 2, \ldots, n+1$.

Observe that for a balanced vector $\vec{u}(x)$ the distance between two any consecutive buses is identical, namely equal to $1 + x/n$ (except between buses B_{n+1} and B_{n+2} where it is $1 + \delta - x$). The following fact then follows from Equation 5 and Definition 6.

Fact 7 *For every* $\vec{w}$ *that is not balanced it holds that*

$$\forall \delta : cost(\vec{w}, \delta, n) > cost(\vec{u}(w_{n+1}), \delta, n).$$

Because of the above fact, we have to choose an optimal waiting time for bus B_{n+1} to compensate the delay δ. The buses B_2 to B_n are then evenly distributed. That is, we have to set $w_{n+1} = x$ so that

$$\max_{\delta} \rho(\vec{u}(x), \delta) = \max_{\delta} \mathsf{cost}(\vec{u}(x), \delta)/\mathsf{opt}(\delta)$$

is as small as possible. In the sequel, we let

$$\rho(x, \delta, n) := \rho(\vec{u}(x), \delta) = \frac{n(1+x/n)^2 + (1+\delta-x)^2}{(n+1)(1+\delta/(n+1))^2}. \tag{6}$$

Observe that, for all $x > 1$, $\rho(x, 0, n) \geq \max_{\delta} \rho(0, \delta, n)$. Hence, we will restrict ourselves to $x \in [0, 1]$. The following result is a simple generalization of Theorem 2.

Theorem 8 *For the case of* $n+2$ *buses and unbounded delays, no (randomized) algorithm can be better than* $(n+1)$*-competitive.*

4.1 Bounded delays

We first consider an adversary that always picks $\delta \in \{0, \Delta\}$, as in the case $n = 1$. Then, we observe that $\rho(w, 0, n)$ is equal to $\rho(w, \Delta, n)$ for $w = n\Delta/(2 + 2n + \Delta) =: w_0(\Delta)$. Further, $\rho(w, \Delta, n)$ is greater than $\rho(w, 0, n)$ and monotonically decreasing in $[0, w_0(\Delta)]$. In $[w_0(\Delta), \Delta]$, $\rho(w, 0, n)$ is greater than $\rho(w, \Delta, n)$ and monontonically increasing. Therefore, the best deterministic algorithm DET against the restricted adversary is given by the value $w_0(\Delta)$ with competitive ratio equal to $\rho(w_0(\Delta), 0, n)$. From Equation 6, it follows that

$$\forall w, \quad \rho(w, 0, n) = 1 + w^2/n,$$

thus implying a competitive ratio of $1 + n\left(\frac{\Delta}{2+2n+\Delta}\right)^2$.

The next lemma shows that the adversary cannot profit from choosing δ in $[0, \Delta]$.

Lemma 9 *For any $w \geq 0$, $\max_{0\leq\delta\leq\Delta} \rho(w, \delta, n) \leq \max_{\delta\in 0,\Delta} \rho(w, \delta, n)$.*

Proof. The proof follows the same steps as Lemma 3.

For $w \geq w_0(\Delta)$:

$$\begin{aligned}
\rho(w, \delta, n) - \rho(w, 0, n) &= \frac{\delta(n + w)(\delta n - (\delta + 2n + 2)w)}{n(\delta + n + 1)^2} \\
&\leq \frac{\delta(n + w)(\delta n - (\delta + 2n + 2)w_0(\Delta))}{n(\delta + n + 1)^2} \\
&= \frac{\delta(n + w)(\delta(n - w_0(\Delta)) - (2n + 2)w_0(\Delta))}{n(\delta + n + 1)^2} \\
&\leq \frac{\delta(n + w)(\Delta(n - w_0(\Delta)) - (2n + 2)w_0(\Delta))}{n(\delta + n + 1)^2} = 0
\end{aligned}$$

For $w < w_0(\Delta)$: Since for $\delta \leq \Delta$

$$\frac{d^2}{dw^2}[\rho(w, \delta, n) - \rho(w, \Delta, n)] = (n + 1)\left(\frac{2 + \frac{2}{n}}{(\delta + n + 1)^2} - \frac{2 + \frac{2}{n}}{(\Delta + n + 1)^2}\right)$$

is greater equal zero, the maximum of this difference is either at $w = 0$ or at $w = w_0(\Delta)$. But

$$\rho(0, \delta, n) - \rho(0, \Delta, n) = (n + 1)\left(\frac{n(\delta - \Delta)(n\delta + 2\Delta\delta + \delta + n\Delta + \Delta)}{(\delta + n + 1)^2(\Delta + n + 1)^2}\right) \leq 0$$

and

$$\rho(w_0(\Delta), \delta, n) - \rho(w_0(\Delta), \Delta, n) = \frac{4\delta n(n+1)(\delta - \Delta)(n + \Delta + 1)}{(\delta + n + 1)^2(2n + \Delta + 2)^2} \leq 0.$$

□

Lemma 9 together with the definition of $w_0(\Delta)$ implies the following:

Theorem 10 *For any $n \geq 1$ and for any $\Delta > 0$, no deterministic online algorithm can have competitive ratio better than $1 + w_0(\Delta)^2/n$, where $w_0(\Delta) = n\Delta/(2 + 2n + \Delta)$. Therefore,* Det *is optimal.*

Similar to Section 3 we can extend this result to any randomized algorithm Rand.

Theorem 11 *For any $n \geq 1$ and for any $\Delta > 0$, no randomized algorithm* Rand *can be better than* Det.

Proof. We first observe that the function $\rho(\cdot, \delta, n)$ is convex, that is, for every $\delta \geq 0$ and for any two vectors $\vec{w}$, $\vec{z}$ it holds that

$$\lambda \cdot \rho(\vec{w}, \delta, n) + (1 - \lambda) \cdot \rho(\vec{z}, \delta, n) \geq \rho(\lambda\vec{w} + (1 - \lambda)\vec{z}, \delta, n), \tag{7}$$

for $\lambda \in [0, 1]$. This follows from Equation 5 and from the fact that the function $(1 + x)^2$ is convex. In particular, for $\vec{e} = (E[W_2], \ldots, E[W_{n+1}])$ being the expected vector of $\vec{W} = (W_2, \ldots, W_{n+1})$, we obtain

$$\begin{aligned} \max_{\delta} E[\rho(\vec{W}, \delta)] &\geq \max_{\delta} \rho(\vec{e}, \delta) && \text{(Jensen's inequality)} \\ &\geq \max_{\delta} \rho(\vec{u}(e_{n+1}), \delta) && \text{(apply Fact 7 with } e_{n+1} = E[W_{n+1}]) \end{aligned}$$

The latter quantity is the competitive ratio of the deterministic algorithm corresponding to the vector $\vec{u}(e_{n+1})$. Theorem 10 thus implies the desired result. □

5 Conclusion

Disposition in a transportation system is critical for customer satisfaction. In this paper we look at a disposition problem arising in high-frequency bus (or tram or train) systems from an algorithmic point of view. We formulate the problem as an online problem and prove tight bounds on the competitive ratio of the problem.

This work provides a basis for competitive analysis of disposition in more complex high-frequency transportation systems. We prove our results for a basic setting which captures some of the main aspects of more complex situations. In particular, our model focuses on the waiting time experienced by passengers waiting at a station. Our results characterize the loss of efficiency (w.r.t. the total waiting time) depending on the amount of resources available (i.e., the number of buses) and on the amount of information about the delay we have (i.e., an upper bound Δ on the delay δ).

We have shown that optimal solutions can be obtained from very simple deterministic algorithms which fix the waiting time of the bus B_{n+1} preceding the delayed bus to a value $w_0(\Delta) \cdot t$, where t is the time two consecutive buses reach the station if not delayed. The optimal waiting time for the other buses is always uniquely determined by the waiting time chosen for B_{n+1}. This simple strategy outperforms any possible randomized choice of buses waiting times.

As future research, our setting can be extended in several ways to reflect real world aspects. For instance, several stations with different arrival rates, several bus lines sharing some stations, or other cost functions can be considered.

Moreover, the formulation as an online problem can be applied to other disposition problems in transportation systems. In this context, the model should be adapted to the different types of possible reactions in the case of an unexpected event and the appropriate cost function.

Acknowledgments. The authors wish to thank an anonymous reviewer for suggesting the use of Jensen's inequality argument in the analysis of randomized algorithms and for suggesting a simpler proof of Lemma 3. This work has been partially supported by the Swiss National Science Foundation under Project no. 200021-107685 (Algorithmic Methods for Delay Management). Most of this work was done while the second author was working at ETH Zürich.

References

B. Adenso-Díaz, M.O. Gonzáles, and P. Gonzáles-Torre. On-line timetable re-scheduling in regional train services. *Transportation Research*, 33B:287–398, 1999.

L. Anderegg, P. Penna, and P. Widmayer. Online train disposition: to wait or not to wait? *Proc. ATMOS 2002, also available in the Electronic Notes in Theoretical Computer Science*, 66(6), 2002.

A. Bertossi, P. Carraresi, and G. Gallo. On some matching problems arising in vehicle scheduling models. *Networks*, 17:271–281, 1987.

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

P. Brucker, J.L. Hurink, and T. Rolfes. Routing of railway carriages: A case study. In *Memorandum No. 1498, Fac. of Mathematical Sciences*. Univ. of Twente, Fac. of Math. Sciences, 1999.

S. Cicerone, G. D'Angelo, G. Di Stefano, D. Frigioni, and A. Navarra. Delay management problem: Complexity results and robust algorithms. In B. Yang, D.-Z. Du, and C. A. Wang, editors, *COCOA*, volume 5165 of *Lecture Notes in Computer Science*, pages 458–468. Springer, 2008. ISBN 978-3-540-85096-0.

G. Dantzig and D. Fulkerson. Minimizing the number of tankers to meet a fixed schedule. *Nav. Res. Logistics Q.*, 1:217–222, 1954.

A. Fiat and G. Woeginger, editors. *Online Algorithms: The State of the Art*. Springer, 1998.

M. Gatto, B. Glaus, R. Jacob, L. Peeters, and P. Widmayer. Railway delay management: Exploring its algorithmic complexity. In *Algorithm Theory - Proceedings SWAT 2004*, pages 199–211. Springer-Verlag LNCS 3111, 2004.

M. Gatto, R. Jacob, L. Peeters, and A. Schöbel. The computational complexity of delay management. In D. Kratsch, editor, *WG*, volume 3787 of *Lecture Notes in Computer Science*, pages 227–238. Springer, 2005. ISBN 3-540-31000-2.

M. Gatto, R. Jacob, L. Peeters, and P. Widmayer. Online delay management on a single train line. In F. Geraets, L. G. Kroon, A. Schöbel, D. Wagner, and C. D. Zaroliagis, editors, *ATMOS*, volume 4359 of *Lecture Notes in Computer Science*, pages 306–320. Springer, 2007. ISBN 978-3-540-74245-6.

A. Ginkel. Event-activity networks in delay management. Master's thesis, University of Kaiserslautern, 2001.

A. Ginkel and A. Schöbel. To Wait or Not to Wait? The Bicriteria Delay Management Problem in Public Transportation. *Transportation Science*, 41(4):527, 2007.

D.E. Heimburger, A.J. Herzenberg, and N.H.M. Wilson. Using simple simulation models in operational analysis of rail transit lines: Case of study of boston's red line. *Transportation Research Record*, 1677:21–30, 1999.

J. L. W. V. Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1):175–193, 1906.

S. Mansilla. Report on disposition of trains. Technical report, ETH Zürich, 2001.

S.W. O'Dell and N.H.M. Wilson. Optimal real-time control strategies for rail transit operations during disruptions. *Lecture Notes in Economics and Math. Sys., Computer-Aided Transit Scheduling*, pages 299–323, 1999.

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

L. Suhl, C. Biederbick, and N. Kliewer. Design of customer-oriented dispatching support for railways. In *Computer-Aided Scheduling of Public Transport*, volume 505 of *Lecture Notes in Economics and Mathematical Systems*, pages 365–386. Springer-Verlag, 2001.

P. Zhu and E. Schnieder. Determining traffic delays through simulation. *Lecture Notes in Economics and Math. Sys., Computer-Aided Scheduling of Public Transport*, pages 387–398, 2001.