Tight Analysis of the Smartstart Algorithm for Online Dial-a-Ride on the Line

Alexander Birx
Institute of Mathematics and Graduate School CE, TU Darmstadt, Germany
birx@gsc.tu-darmstadt.de

Yann Disser
Institute of Mathematics and Graduate School CE, TU Darmstadt, Germany
disser@mathematik.tu-darmstadt.de

Abstract

The online Dial-a-Ride problem is a fundamental online problem in a metric space, where transportation requests appear over time and may be served in any order by a single server with unit speed. Restricted to the real line, online Dial-a-Ride captures natural problems like controlling a personal elevator. Tight results in terms of competitive ratios are known for the general setting and for online TSP on the line (where source and target of each request coincide). In contrast, online Dial-a-Ride on the line has resisted tight analysis so far, even though it is a very natural online problem.

We conduct a tight competitive analysis of the Smartstart algorithm that gave the best known results for the general, metric case. In particular, our analysis yields a new upper bound of 2.94 for open, non-preemptive online Dial-a-Ride on the line, which improves the previous bound of 3.41 [Krumke'00]. The best known lower bound remains 2.04 [SODA'17]. We also show that the known upper bound of 2 [STACS'00] regarding Smartstart's competitive ratio for closed, non-preemptive online Dial-a-Ride is tight on the line.

2012 ACM Subject Classification Theory of computation → Online algorithms; Mathematics of computing → Combinatorial optimization

Keywords and phrases dial-a-ride on the line, elevator problem, online algorithms, competitive analysis, smartstart, competitive ratio

Digital Object Identifier 10.4230/LIPIcs.STACS.2019.15

Related Version `http://arxiv.org/abs/1901.04272`

Funding This work was supported by the "Excellence Initiative" of the German Federal and State Governments and the Graduate School CE at TU Darmstadt.

1 Introduction

Online optimization deals with settings where algorithmic decisions have to be made over time without knowledge of the future. A typical introductory example is the problem of controlling an elevator/conveyor system, where requests to transport passengers/goods arrive over time and the elevator needs to decide online how to adapt its trajectory along the real line. In terms of competitive analysis, the central question in this context is how much longer our trajectory will be in the worst-case, relative to an optimum offline solution that knows all requests ahead of time, i.e., we ask for solutions with good *competitive ratio*.

While the elevator problem is a natural online problem, even simplified versions of it have long resisted tight analysis. *Online* TSP *on the line* is such a simplification, where a single server on the real line needs to serve requests that appear over time at arbitrary positions by visiting their location, i.e., requests do not need to be transported. We distinguish the *closed* and *open* variants of this problem, depending on whether the server needs to eventually return to the origin or not. Determining the exact competitive ratios for either variant had been an open problem for more than two decades [3, 5, 12, 13, 15, 16], when Bjelde et al. [4] were finally able to conduct a tight analysis that established competitive ratios of roughly 1.64 for the closed case and 2.04 for the open case.

The next step towards formally capturing the intuitive elevator problem is to allow transportation requests that appear over time; and to fix a capacity $c \in \mathbb{N} \cup \{\infty\}$ of the server that limits the number of transportation requests that can be served simultaneously. The resulting *online* Dial-a-Ride *problem on the line* has received considerable attention in the past [1, 4, 8, 13, 14, 16], but still resists tight analysis. The best known (non-preemptive) bounds put the competitive ratio in the range $[1.75, 2]$ for the closed variant (see [4, 1]). For the open variant the best known (non-preemptive) bounds put the competitiv ratio in the range $[2.04, 3]$ for $c = 1$ and in the range $[2.04, 3.41]$ for $c > 1$ (see [4, 13]). In this paper, we show an improved upper bound of (roughly) 2.94 for open online Dial-a-Ride on the line for arbitrary capacity $c \in \mathbb{N} \cup \{\infty\}$.

A straight-forward algorithm for online Dial-a-Ride on the line is the algorithm Ignore [1]: Whenever the server is idle and unserved requests R_t are present at the current time t, compute an optimum schedule to serve these requests from the current location, and follow this schedule while *ignoring* newly incoming requests. Ignore has a competitive ratio of exactly 4.[†] This competitive ratio can be improved by potentially waiting before starting the optimum schedule, in order to protect against requests that come in right after we decide to start. Ascheuer et al. [1] proposed the algorithm Smartstart (see Algorithm 1) that delays the execution of the optimum schedule until a certain time t relative to the length $L(t, p, R_t)$ of this schedule (formal definitions below).

Smartstart is parameterized by a factor $\Theta > 1$ that scales this waiting period. In this paper, we conduct a tight analysis of the best competitive ratio of Smartstart for open/closed online Dial-a-Ride on the line, over all parameter values $\Theta > 1$.

Results and techniques. The Smartstart algorithm is of particular importance for online Dial-a-Ride, since, on arbirary metric spaces, it achieves the best possible competitive ratio of 2 for the closed variant [1, 3], and the best known competitive ratio of $2 + \sqrt{2} \approx 3.41$ for the open variant [13]. We provide a conclusive treatment of this algorithm for online Dial-a-Ride on the line in terms of competitive analysis, both for the open and the closed variant.

Regarding the open case, we show that Smartstart attains a competitive ratio of $\rho^* \approx 2.94$ for parameter value $\Theta^* \approx 2.05$ (Section 3). To show this, we derive two separate upper bounds depending on Θ (cf. Figure 1): an upper bound $f_1(\Theta)$ for the case that Smartstart has a waiting period before starting its last schedule (Proposition 3.3), and an upper bound $f_2(\Theta)$ for the case that Smartstart begins its final schedule immediately (Proposition 3.4). The resulting general upper bound of $\max\{f_1(\Theta), f_2(\Theta)\}$ has its minimum precisely at the intersection point (Θ^*, ρ^*) of f_1 and f_2.

On the other hand, we show that for $\Theta \in (2, 3)$ there are instances where Smartstart waits before starting its final schedule and has competitive ratio at least $f_1(\Theta)$ (Proposition 4.2). Similarly, we show that for $\Theta \in [2, 2.303]$ there are instances where Smartstart does not wait before starting its final schedule and has competitive ratio at least $f_2(\Theta)$ (Proposition 4.3). Together, this implies that the general upper bound of $\max\{f_1(\Theta), f_2(\Theta)\}$ is tight for $\Theta \in (2, 2.303]$, and thus for $\Theta = \Theta^*$ (cf. Figure 1).

[†] The full proof can be found at `http://arxiv.org/abs/1901.04272`.

To complete our analysis of SMARTSTART, we give lower bound constructions for different domains of Θ (g_1 through g_4 in Figure 1) that establish that Θ^* is indeed the best parameter choice for SMARTSTART in the worst-case (Lemma 4.4). The key ingredient to all our lower bounds is a way to *lure* SMARTSTART away from the origin (Lemma 4.1).

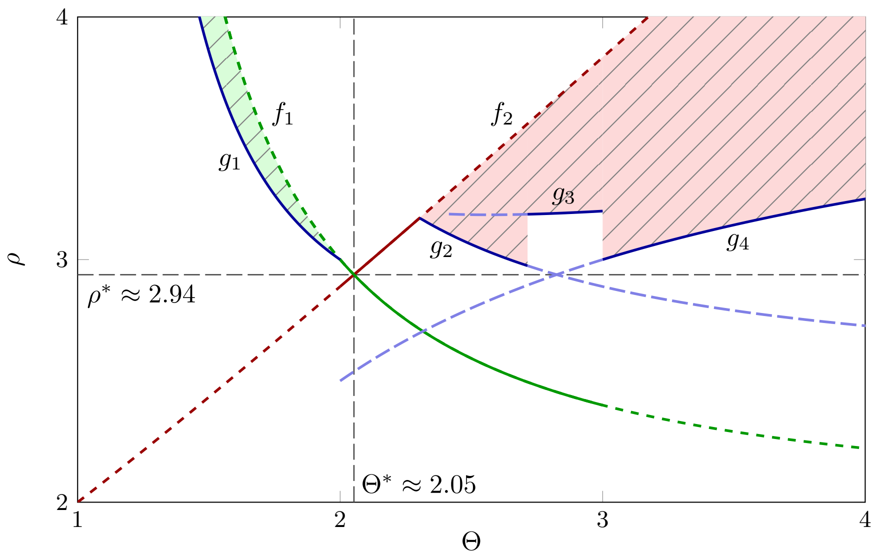

Figure 1 Overview over our bounds for SMARTSTART. The functions f_1 (green) / f_2 (red) are upper bounds for the cases where SMARTSTART waits / does not wait before starting the final schedule, respectively. The upper bounds are drawn solid in the domains where they are tight for their corresponding case. The functions g_1 through g_4 (blue) are general lower bounds; dashed continuations indicate how far these bounds could be extended.

Finally, for the closed variant of the problem, we provide a lower bound of 2 on the best-possible competitive ratio of SMARTSTART over all possible choices of the parameter $\Theta > 1$ (Section 5). This tightly matches the known upper bound for general metric spaces [1].

Significance. The main contribution of this paper is a conclusive treatment of the algorithm SMARTSTART for online DIAL-A-RIDE on the line in terms of competitive analysis. Additionally, our analysis yields an improved upper bound of (roughly) 2.94 for non-preemptive, open online DIAL-A-RIDE on the line. This is the first bound below 3 and narrows the gap for the competitive ratio to $[2.04, 2.94]$. Our work is likely to serve as a starting point towards devising better algorithms (preemptive or non-preemptive) that narrow the gaps for both the open and closed setting by avoiding critical "mistakes" of SMARTSTART, as evidenced by our lower bound constructions

Further related work. In this paper, we focus on the non-preemptive variant of online DIAL-A-RIDE on the line, where requests cannot be unloaded on the way in reaction to the arrival of new requests. For the case where preemption is allowed, the best known bounds for the closed version are $[1.64, 2]$ (see [3, 1]), which is slightly worse than the gap of $[1.75, 2]$ in the non-preemptive case. On the other hand, the best bounds for the open, preemptive variant are $[2.04, 2.41]$ (see [4]), which is better than the gap of $[2.04, 2.94]$ in the non-preemptive case. In particular, the preemptive and non-preemptive cases can currently not be separated in terms of competitive ratios.

A variant of the online Dial-a-Ride problem where the objective is to minimize the maximal flow time, instead of the makespan, has been studied by Krumke et al. [14, 15]. They established that in many metric spaces no online algorithm can be competitive with respect to this objective. Hauptmeier et al. [11] showed that a competitive algorithm is possible if we restrict ourselves to instances with "reasonable" load, which roughly means that requests that appear over a sufficiently large time period T can always be served in time at most T.

Lipmann et al. [17] studied a natural variant of closed, online Dial-a-Ride where the destinations of requests are only revealed upon collection by the server. For general metric spaces and server capacity c, they showed a tight competitive ratio of 3 in the preemptive setting, and lower/upper bounds of $\max\{3.12, c\}$ and $2c + 2$, respectively, in the non-preemptive setting.

Yi and Tian [18] considered the online Dial-a-Ride problem with deadlines, with the objective of serving the maximum number of requests. They provided bounds on the competitive ratio depending on the diameter of the metric space. In [19] they further studied this setting when the destination of requests are only revealed upon collection by the server.

The offline version of Dial-a-Ride on the line has been studied in various settings, for an overview see [7]. For the closed, non-preemptive case without release times, Gilmore and Gomory [9] and Atallah and Kosaraju [2] gave a polynomial time algorithm for a server with unit capacity $c = 1$, and Guan [10] showed that the problem is hard for $c = 2$. Bjelde et al. [4] extended this result to any finite $c \geq 2$ and both the open and closed case. They further showed that with release times the problem is already hard for finite $c \geq 1$. On the other hand, the complexity of the case $c = \infty$ has not yet been established. The closed, preemptive case without release times was shown to be polynomial time solvable for $c = 1$ by Atallah and Kosaraju [2], and for $c \geq 2$ by Guan [10].

For the closed, non-preemptive case with finite capacity, Krumke [13] provided a 3-approximation algorithm. Finally, Charikar and Raghavachari [6] gave approximation algorithms for the closed case without release times, both preemptive and non-preemptive, on general metric spaces. They also claimed to have a 2-approximation for the line, but this result appears to be incorrect (personal communication).

2 Preliminaries

Formally, an instance of Dial-a-Ride on the line is given by a set of requests denoted by $\sigma = \{(a_1, b_1; r_1), (a_2, b_2; r_2), \dots, (a_n, b_n; r_n)\}$ that need to be served by a single server with capacity $c \in \mathbb{N} \cup \{\infty\}$, travelling with unit speed and starting at the origin on the real line. Request σ_i appears at time $r_i > 0$ at position $a_i \in \mathbb{R}$ of the real line and needs to be transported to position $b_i \in \mathbb{R}$. The objective of the Dial-a-Ride problem on the line is to find a shortest schedule for the server to transport all requests without carrying more than c requests at once, where the length of a schedule is the length of the resulting trajectory. In the *closed* version of the problem, the server eventually needs to return to the origin, in the *open* version it does not. In the *online* Dial-a-Ride problem on the line, each request σ_i is revealed only at time r_i, and n is only revealed implicitly by the fact that no more requests appear. In contrast, in the *offline* problem, all requests are known ahead of time (but release times still need to be respected).

We define $L(t, p, R)$ to be the length of a shortest schedule that starts at position p at time t and serves all requests in $R \subseteq \sigma$ after they appeared (i.e., the schedule must respect release times). Observe that, for all $0 \le t \le t'$, $p, p' \in \mathbb{R}$, and $R \subseteq \sigma$, we have

$$L(t, p, R) \ge L(t', p, R), \tag{1}$$
$$L(t, p, R) \le |p - p'| + L(t, p', R). \tag{2}$$

By $x_- := \min\{0, \min_{i=1,\dots,n}\{a_i\}, \min_{i=1,\dots,n}\{b_i\}\}$ we denote the leftmost and by $x_+ := \max\{0, \max_{i=1,\dots,n}\{a_i\}, \max_{i=1,\dots,n}\{b_i\}\}$ the rightmost position that needs to be visited by the server. Here and throughout, we orient the real line from left to right. Obviously, there is an optimum trajectory that only visits points in $[x_-, x_+]$, and we let Opt be such a trajectory and $\text{Opt}(\sigma) := L(0, 0, \sigma)$ be its length.

Algorithm 1: SmartStart.

```
p_1 ← 0
for j = 1, 2, ... do
    while t ≤ L(t, p_j, R_t)/(Θ − 1) do
        wait
    t_j ← t
    S_j ← optimal offline schedule serving R_t starting from p_j
    execute S_j
    p_{j+1} ← current position
```

For the description of online algorithms, we denote by t the current time and by R_t the set of requests that have appeared until time t but have not been served yet. The algorithm SmartStart is given in Algorithm 1. Essentially, SmartStart waits before starting an optimal schedule to serve all available requests at time

$$\min_{t' \ge t} \left\{ t' \ge \frac{L(t', p, R_{t'})}{\Theta - 1} \right\}, \tag{3}$$

where p is the current position of the server and $\Theta > 1$ is a parameter of the algorithm that scales the waiting time. Importantly, SmartStart ignores incoming requests while executing a schedule. Whenever we need to distinguish the behavior of SmartStart for different values of $\Theta > 1$, we write SmartStart_Θ to make the choice of Θ explicit. The length of SmartStart's trajectory is denoted by $\text{SmartStart}(\sigma)$. Note that the schedules used by SmartStart are NP-hard to compute for $1 < c < \infty$, see [4].

We let $N \in \mathbb{N}$ be the number of schedules needed by SmartStart to serve σ. The j-th schedule is denoted by S_j, its starting time by t_j, its starting point by p_j, its ending point by p_{j+1} (cf. Algorithm 1), and the set of requests served in S_j by σ_{S_j}. For convenience, we set $t_0 = p_0 = 0$. Finally, we denote by $y_-^{S_j}$ the leftmost and by $y_+^{S_j}$ the rightmost position that occurs in the requests σ_{S_j}. Note that $y_-^{S_j}$ and $y_+^{S_j}$ need not lie on different sides of the origin, in contrast to $x_{-/+}$.

3 Upper Bound for the Open Version

In this section, we give an upper bound on the completion time

$$\text{SmartStart}(\sigma) = t_N + L(t_N, p_N, \sigma_{S_N}) \tag{4}$$

of SmartStart, relative to $\text{Opt}(\sigma)$. To do this, we consider two cases, depending on whether or not SmartStart waits after finishing schedule S_{N-1} and before starting the final schedule S_N. If SMARTSTART waits, the starting time of schedule S_N is given by

$$t_N = \frac{1}{\Theta - 1} L(t_N, p_N, \sigma_{S_N}), \tag{5}$$

otherwise, we have

$$t_N = t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{S_{N-1}}). \tag{6}$$

We start by giving a lower bound on the starting time of a schedule.†

▶ **Lemma 3.1.** *Algorithm* SMARTSTART *does not start schedule S_j earlier than time $\frac{|p_{j+1}|}{\Theta}$, i.e., we have $t_j \geq \frac{|p_{j+1}|}{\Theta}$.*

Proof sketch. Since SMARTSTART at least has to move from p_j to p_{j+1}, we have

$$L(t_j, p_j, \sigma_{S_j}) \geq |p_j - p_{j+1}|.$$

Note however that SMARTSTART needs at least time $|p_j|$ to reach p_j. Therefore, we have

$$\begin{aligned} t_j &\geq \min\{t \geq |p_j| \colon t + |p_j - p_{j+1}| \leq \Theta t\} \\ &= \min\left\{t \geq |p_j| \colon \frac{|p_j - p_{j+1}|}{\Theta - 1} \leq t\right\} \\ &= \max\left\{|p_j|, \frac{|p_j - p_{j+1}|}{\Theta - 1}\right\}. \end{aligned}$$

The claims now follows by showing $\max\left\{|p_j|, \frac{|p_j - p_{j+1}|}{\Theta-1}\right\} \geq \frac{|p_{j+1}|}{\Theta}$. ◀

The following bound on the length of SMARTSTART's schedules is an essential ingredient in our upper bounds.

▶ **Lemma 3.2.** *For every schedule S_j of* SMARTSTART*, we have*

$$L(t_j, p_j, \sigma_{S_j}) \leq \left(1 + \frac{\Theta}{\Theta + 2}\right) \mathrm{OPT}(\sigma).$$

Proof. First, we notice that by the triangle inequality we have

$$L(t_j, p_j, \sigma_{S_j}) \leq |p_j| + L(t_j, 0, \sigma_{S_j}) \leq \mathrm{OPT}(\sigma) + |p_j|. \tag{7}$$

Now, let $\sigma_{S_j}^{\mathrm{OPT}}$ be the first request of σ_{S_j} that is picked up by OPT and let a_j^{OPT} be its starting point and r_j^{OPT} be its release time. We have

$$L(t_j, p_j, \sigma_{S_j}) \leq |a_j^{\mathrm{OPT}} - p_j| + L(t_j, a_j^{\mathrm{OPT}}, \sigma_{S_j}), \tag{8}$$

again by the triangle inequality. Since OPT serves all requests of σ_{S_j} starting at position a_j^{OPT} no earlier than time r_j^{OPT}, we have

$$L(t_j, a_j^{\mathrm{OPT}}, \sigma_{S_j}) \overset{r_j^{\mathrm{OPT}} \leq t_j}{\leq} L(r_j^{\mathrm{OPT}}, a_j^{\mathrm{OPT}}, \sigma_{S_j}) \leq \mathrm{OPT}(\sigma) - r_j^{\mathrm{OPT}}, \tag{9}$$

which yields

$$\begin{aligned} L(t_j, p_j, \sigma_{S_j}) &\overset{(8)}{\leq} |a_j^{\mathrm{OPT}} - p_j| + L(t_j, a_j^{\mathrm{OPT}}, \sigma_{S_j}) \\ &\overset{(9)}{\leq} \mathrm{OPT}(\sigma) + |a_j^{\mathrm{OPT}} - p_j| - r_j^{\mathrm{OPT}} \\ &\overset{t_{j-1} < r_j^{\mathrm{OPT}}}{<} \mathrm{OPT}(\sigma) + |a_j^{\mathrm{OPT}} - p_j| - t_{j-1}. \end{aligned} \tag{10}$$

Since p_j is the destination of a request, OPT needs to visit it. In the case that OPT visits p_j before collecting $\sigma_{S_j}^{\text{OPT}}$, OPT still has to collect and serve every request of σ_{S_j} after it has visited position p_j the first time, which directly implies

$$\left(1+\frac{\Theta}{\Theta+2}\right)\text{OPT}(\sigma) > \text{OPT}(\sigma) \geq L(|p_j|, p_j, \sigma_{S_j}) \overset{|p_j|\leq t_j}{\geq} L(t_j, p_j, \sigma_{S_j}).$$

On the other hand, if OPT collects $\sigma_{S_j}^{\text{OPT}}$ before visiting the position p_j, we have

$$t_{j-1} + |a_j^{\text{OPT}} - p_j| \overset{t_{j-1}<r_j^{\text{OPT}}}{<} r_j^{\text{OPT}} + |a_j^{\text{OPT}} - p_j| \leq \text{OPT}(\sigma), \tag{11}$$

since OPT cannot collect $\sigma_{S_j}^{\text{OPT}}$ before time r_j^{OPT} and then still has to visit position p_j. Thus, we have

$$\begin{aligned} L(t_j, p_j, \sigma_{S_j}) &\overset{(10)}{<} \text{OPT}(\sigma) + |a_j^{\text{OPT}} - p_j| - t_{j-1} \\ &\overset{(11)}{\leq} 2\text{OPT}(\sigma) - 2t_{j-1} \\ &\overset{\text{Lem 3.1}}{\leq} 2\text{OPT}(\sigma) - 2\frac{|p_j|}{\Theta}. \end{aligned} \tag{12}$$

This implies

$$L(t_j, p_j, \sigma_{S_j}) \overset{(7),(12)}{\leq} \min\left\{\text{OPT}(\sigma) + |p_j|, 2\text{OPT}(\sigma) - \frac{2}{\Theta}|p_j|\right\} \leq \left(1+\frac{\Theta}{\Theta+2}\right)\text{OPT}(\sigma),$$

since the minimum above is largest for $|p_j| = \frac{\Theta}{\Theta+2}\text{OPT}(\sigma)$. ◀

The following proposition uses Lemma 3.2 to provide an upper bound for the competitive ratio of SMARTSTART, in the case, where SMARTSTART does have a waiting period before starting the final schedule.

▶ **Proposition 3.3.** *In the case that* SMARTSTART *waits before executing S_N, we have*

$$\frac{\text{SMARTSTART}(\sigma)}{\text{OPT}(\sigma)} \leq f_1(\Theta) := \frac{2\Theta^2 + 2\Theta}{\Theta^2 + \Theta - 2}.$$

Proof. Assume SMARTSTART waits before starting the final schedule. Then we have

$$t_N + L(t_N, p_N, \sigma_{S_N}) = \Theta t_N \tag{13}$$

by definition of SMARTSTART. This implies

$$\text{SMARTSTART}(\sigma) \overset{(4)}{=} t_N + L(t_N, p_N, \sigma_{S_N}) \overset{(13)}{=} \Theta t_N \overset{(5)}{=} \frac{\Theta}{\Theta-1} L(t_N, p_N, \sigma_{S_N}).$$

Lemma 3.2 thus yields the claimed bound:

$$\begin{aligned} \text{SMARTSTART}(\sigma) &= \frac{\Theta}{\Theta-1} L(t_N, p_N, \sigma_{S_N}) \\ &\overset{\text{Lem 3.2}}{\leq} \frac{\Theta}{\Theta-1}\left(1+\frac{\Theta}{\Theta+2}\right)\text{OPT}(\sigma) \\ &= \frac{2\Theta^2 + 2\Theta}{\Theta^2 + \Theta - 2}\text{OPT}(\sigma). \end{aligned}$$ ◀

It remains to examine the case, where the algorithm SMARTSTART has no waiting period before starting the final schedule.†

▶ **Proposition 3.4.** *If* SMARTSTART *does not wait before executing S_N, we have*

$$\frac{\text{SMARTSTART}(\sigma)}{\text{OPT}(\sigma)} \leq f_2(\Theta) := \left(\Theta + 1 - \frac{\Theta - 1}{3\Theta + 3}\right).$$

Proof sketch. If SMARTSTART starts S_N without waiting, its completion time is given by

$$\text{SMARTSTART}(\sigma) \overset{(6)}{=} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{S_{N-1}}) + L(t_N, p_N, \sigma_{S_N}). \tag{14}$$

Let $\sigma_{S_N}^{\text{OPT}}$ be the first request of σ_{S_N} that is picked up by OPT and let a_N^{OPT} be its starting point and r_N^{OPT} be its release time. Then we have

$$\text{OPT}(\sigma) \geq r_N^{\text{OPT}} + L(r_N^{\text{OPT}}, a_N^{\text{OPT}}, \sigma_{S_N}). \tag{15}$$

Using the triangle inequality as well as the definition of SMARTSTART, we obtain

$$\begin{aligned}
\text{SMARTSTART}(\sigma) &\overset{(14)}{=} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{S_{N-1}}) + L(t_N^{\text{OPT}}, p_N, \sigma_{S_N})\\
&\overset{(3)}{\leq} \Theta t_{N-1} + L(t_N^{\text{OPT}}, p_N, \sigma_{S_N})\\
&\overset{(1)}{\leq} \Theta t_{N-1} + |p_N - a_N^{\text{OPT}}| + L(t_N^{\text{OPT}}, a_N^{\text{OPT}}, \sigma_{S_N})\\
&\overset{(15)}{\leq} \Theta t_{N-1} + |p_N - a_N^{\text{OPT}}| + \text{OPT}(\sigma) - r_N^{\text{OPT}}\\
&\overset{r_N^{\text{OPT}} > t_{N-1}}{<} (\Theta - 1) r_N^{\text{OPT}} + |p_N - a_N^{\text{OPT}}| + \text{OPT}(\sigma).
\end{aligned}$$

Clearly, $\text{OPT}(\sigma) \geq r_N^{\text{OPT}}$ since $\sigma_{S_N}^{\text{OPT}}$ cannot be served before this time, and $\text{OPT}(\sigma) \geq |p_N - a_N^{\text{OPT}}|$ since p_N must be the source or destination of a request (or the origin if $N = 1$) and must thus be visited by OPT. It follows from the above that $\text{SMARTSTART}(\sigma) \leq (\Theta + 1)\text{OPT}(\sigma)$. To get a better bound, we use that not both inequalities for $\text{OPT}(\sigma)$ can be tight simultaneously: From $\text{OPT}(\sigma) = r_N^{\text{OPT}}$ it follows that OPT finishes at position a_N^{OPT}. Assume that $\text{OPT}(\sigma) = |p_N - a_N^{\text{OPT}}|$ holds as well. Since OPT finishes at position a_N^{OPT}, this is only possible if $p_N = 0$ and $\text{OPT}(\sigma) = |a_N^{\text{OPT}}|$. Without loss of generality, there is no request $(0, 0; 0)$, hence SMARTSTART always waits before starting its first schedule, and thus a schedule S_{N-1} must exist. Because of $p_N = 0$, this schedule must end in the origin, which implies that there is some request that needs to be delivered to the origin after time 0. But this contradicts $\text{OPT}(\sigma) = |a_N^{\text{OPT}}|$, since OPT needs to deliver this request, too. The bound of the proposition is now obtained by carefully balancing r_N^{OPT} and $|p_N - a_N^{\text{OPT}}|$. ◀

We combine the results of Proposition 3.3 and Proposition 3.4 to obtain the main result of this section.

▶ **Theorem 3.5.** *Let Θ^* be the only positive, real solution of $f_1(\Theta) = f_2(\Theta)$, i.e.,*

$$\Theta^* + 1 - \frac{\Theta^* - 1}{3\Theta^* + 3} = \frac{2\Theta^{*2} + 2\Theta^*}{\Theta^{*2} + \Theta^* - 2}.$$

Then, $\text{SMARTSTART}_{\Theta^*}$ *is ρ^*-competitive with $\rho^* := f_1(\Theta^*) = f_2(\Theta^*) \approx 2.93768$.*

Proof. For the case, where SmartStart does wait before starting the final schedule, we have established the upper bound

$$\frac{\textsc{SmartStart}(\sigma)}{\textsc{Opt}(\sigma)} \le \frac{2\Theta^2+2\Theta}{\Theta^2+\Theta-2} = f_1(\Theta)$$

in Proposition 3.3 and for the case, where SmartStart starts the final schedule immediately after the second to final one, we have established the upper bound

$$\frac{\textsc{SmartStart}(\sigma)}{\textsc{Opt}(\sigma)} \le \Theta + 1 - \frac{\Theta-1}{3\Theta+3} = f_2(\Theta)$$

in Proposition 3.4. Therefore the parameter for SmartStart with the smallest upper bound is

$$\Theta^* = \underset{\Theta>1}{\operatorname{argmin}} \left\{\max\{f_1(\Theta), f_2(\Theta)\}\right\}.$$

We note that f_1 is strictly decreasing for $\Theta > 1$ and that f_2 is strictly increasing for $\Theta > 1$. Therefore the minimum above lies at the intersection point of f_1 and f_2 that is larger than 1, i.e., Θ^* is the only positive, real solution of

$$\Theta + 1 - \frac{\Theta-1}{3\Theta+3} = \frac{2\Theta^2+2\Theta}{\Theta^2+\Theta-2}.$$

The resulting upper bound for the competitive ratio is

$$\rho^* = f_1(\Theta^*) = f_2(\Theta^*) \approx 2.93768.$$ ◀

4 Lower Bound for the Open Version

In this section, we explicitly construct instances that demonstrate that the upper bounds given in the previous section are tight for certain ranges of $\Theta > 1$, in particular for $\Theta = \Theta^*$ (as in Theorem 3.5). Further, we show that choices of $\Theta > 1$ different from Θ^* yield competitive ratios worse than $\rho^* \approx 2.94$. Together, this implies that ρ^* is exactly the best possible competitive ratio for SmartStart.

All our lower bounds rely on the following lemma that gives a way to lure SmartStart away from the origin, with almost no time overhead. More specifically, the lemma provides a way to make SmartStart move to any position $p > 0$ within time $p + \mu$, where $\mu > 0$ is arbitrarily small.

▶ **Lemma 4.1.** *Let the capacity $c \in \mathbb{N} \cup \{\infty\}$ of the server be arbitrary but fixed, $p > 0$ be any position on the real line and $\mu > 0$ be any positive number. Furthermore, let $\delta > 0$ be such that $\frac{p}{\delta\Theta} = n \in \mathbb{N}$ and $\delta < (\Theta - 1)\mu$. Algorithm* SmartStart *finishes serving the set of requests $\sigma = \{\sigma_1, \ldots, \sigma_{n+1}\}$ with*

$$\begin{aligned}
\sigma_1 &= (\delta, \delta; 0),\\
\sigma_i &= \left(i\delta, i\delta; \frac{1}{\Theta-1}\delta + (i-1)\delta\right) \quad \text{for } i \in \{2, \ldots, n\}\\
\sigma_{n+1} &= (p, p; \mu + n\delta)
\end{aligned}$$

and reaches the position p at time $p + \mu$, provided that no additional requests appear until time $\frac{p}{\Theta} + \mu$.

Proof. We show via induction that every request σ_i with $i \in \{1, \ldots, n\}$ is served in a separate schedule S_i with starting position $p_i = (i-1)\delta$ and starting time

$$t_i = \frac{1}{\Theta - 1}\delta + (i-1)\delta.$$

This is clear for $i = 1$: By definition, SMARTSTART starts from $p_1 = 0$. The schedule S_1 to serve σ_1 is started at time

$$t_1 = \min\left\{ t \geq 0 \;\middle|\; \frac{L(t, 0, \{\sigma_1\})}{\Theta - 1} \leq t \right\} = \frac{1}{\Theta - 1}\delta,$$

and reaches position δ at time $\frac{1}{\Theta-1}\delta + \delta = \frac{\Theta}{\Theta-1}\delta$. Note that the release time of every request σ_i is larger than t_1, ensuring that S_1 indeed only serves σ_1.

We assume the claim is true for some $k \in \{1, \ldots, n-1\}$. Consider $i = k+1$. By reduction, the server finishes schedule S_k at position $p_{k+1} = k\delta$ at time $\frac{1}{\Theta-1}\delta + k\delta$. Therefore, we have

$$t_{k+1} \geq \frac{1}{\Theta - 1}\delta + k\delta.$$

On the other hand, we have

$$\frac{L\left(\frac{\delta}{\Theta-1} + k\delta, k\delta, \{\sigma_{k+1}\}\right)}{\Theta - 1} = \frac{\delta}{\Theta - 1} < \frac{1}{\Theta - 1}\delta + k\delta.$$

Since there are no other unserved requests at time $\frac{1}{\Theta-1}\delta + k\delta$, the schedule S_{k+1} is started at time $t_{k+1} = \frac{1}{\Theta-1}\delta + k\delta$ and only serves σ_{k+1} as claimed. It remains to examine the final request σ_{n+1}. The above shows that in the schedule S_n is finished at time

$$t_n + L(t_n, p_n, \{\sigma_n\}) = \frac{1}{\Theta - 1}\delta + (n-1)\delta + \delta = \frac{1}{\Theta - 1}\delta + n\delta < \mu + n\delta$$

at position $n\delta = \frac{p}{\Theta}$, i.e., before the request σ_{n+1} is released at time $\mu + n\delta$. On the other hand, we have

$$\frac{L\left(\mu + n\delta, \frac{p}{\Theta}, \{\sigma_{n+1}\}\right)}{\Theta - 1} = \frac{\frac{\Theta-1}{\Theta}p}{\Theta - 1} = \frac{p}{\Theta} = n\delta < \mu + n\delta.$$

Therefore the final schedule S_{n+1} is started at time $t_{n+1} = \mu + n\delta = \mu + \frac{p}{\Theta}$, and we get

$$\begin{aligned} \text{SMARTSTART}((\sigma_i)_{i\in\{1,\ldots,n+1\}}) &= t_{n+1} + L(t_{n+1}, p_{n+1}, \{\sigma_{n+1}\}) \\ &= \mu + \frac{p}{\Theta} + \frac{\Theta - 1}{\Theta}p \\ &= \mu + p. \end{aligned}$$

Note that for every request the starting point is identical to the ending point. Thus, our construction remains valid for every capacity $c \in \mathbb{N} \cup \{\infty\}$. Furthermore, there is no interference with requests that are released after time $t_{n+1} = \mu + \frac{p}{\Theta}$. ◀

Equipped with this strategy to lure SMARTSTART away from the origin, we now move on to establish lower bounds matching Propositions 3.3 and 3.4.[†]

▶ **Proposition 4.2.** *Let the capacity $c \in \mathbb{N} \cup \{\infty\}$ of the server be arbitrary but fixed and let $2 < \Theta < 3$. For every sufficiently small $\varepsilon > 0$, there is a set of requests σ such that* SMARTSTART *waits before starting the final schedule and such that the inequality*

$$\frac{\text{SMARTSTART}(\sigma)}{\text{OPT}(\sigma)} \geq \frac{2\Theta^2 + 2\Theta}{\Theta^2 + \Theta - 2} - \varepsilon$$

holds, i.e., the upper bound established in Proposition 3.3 is tight for $\Theta \in (2, 3)$.

Proof sketch. We start by luring SmartStart to position 1 via Lemma 4.1. This can be done such that the schedule ending in 1 starts at time $\mu + \frac{1}{\Theta}$ for some sufficiently small $\mu > 0$ Immediately after the start of this schedule, we add a series of non-overlapping requests that require the server to move to position $-\frac{1}{\Theta}$ and afterwards to position 1. We can show that Opt serves the resulting set of requests simply by moving to $-\frac{1}{\Theta}$ and then straight to 1. On the other hand, independent of the capacity, SmartStart needs to cross the space between the origin and point 1 two more times. A quantitative analysis of this setting yields the claimed bound. ◀

▶ **Proposition 4.3.** *Let the capacity $c \in \mathbb{N} \cup \{\infty\}$ of the server be arbitrary but fixed and let $2 \le \Theta \le \frac{1}{2}(1+\sqrt{13})$. For every sufficiently small $\varepsilon > 0$ there is a set of requests σ such that* SmartStart *immediately starts S_N after S_{N-1} and such that*

$$\frac{\textsc{SmartStart}(\sigma)}{\textsc{Opt}(\sigma)} \ge \Theta + 1 - \frac{\Theta - 1}{3\Theta + 3} - \varepsilon,$$

i.e., the upper bound established in Proposition 3.4 is tight for $\Theta \in [2, \frac{1}{2}(1+\sqrt{13})] \approx [2, 2.303]$.

Proof. Let $\varepsilon > 0$ with $\varepsilon < \frac{1}{5\Theta}\frac{3\Theta^2-\Theta}{3\Theta+3}$ and $\varepsilon' = \frac{3\Theta+3}{3\Theta^2-\Theta}\varepsilon$. We apply Lemma 4.1 with $p = 1$ and $\mu = \frac{\varepsilon'}{2}$. For convenience, we start the enumeration of the schedules with the first schedule after the application of Lemma 4.1. Algorithm SmartStart reaches position $p_1 = 1$ at time $1 + \frac{\varepsilon'}{2}$. Now let the requests

$$\sigma_1^{(1)} = \left(2 + \frac{1}{\Theta} - \varepsilon', 2 + \frac{1}{\Theta} - \varepsilon'; \frac{1}{\Theta} + \varepsilon'\right),$$
$$\sigma_1^{(2)} = \left(-\frac{1}{\Theta}, -\frac{1}{\Theta}; \frac{1}{\Theta} + \varepsilon'\right)$$

appear. Note that both requests are released after time $\frac{1}{\Theta} + \frac{\varepsilon'}{2}$ and, therefore, do not interfere with the application of Lemma 4.1. If SmartStart serves $\sigma_1^{(2)}$ before serving $\sigma_1^{(1)}$ the time it needs is at least

$$\left|1 - \left(-\frac{1}{\Theta}\right)\right| + \left|\left(-\frac{1}{\Theta}\right) - \left(2 + \frac{1}{\Theta} - \varepsilon'\right)\right| = 1 + \frac{1}{\Theta} + 2 + \frac{2}{\Theta} - \varepsilon' = 3 + \frac{3}{\Theta} - \varepsilon'.$$

The best schedule that serves $\sigma_1^{(2)}$ after serving $\sigma_1^{(1)}$ needs time

$$\left|1 - \left(2 + \frac{1}{\Theta} - \varepsilon'\right)\right| + \left|\left(2 + \frac{1}{\Theta} - \varepsilon'\right) - \left(-\frac{1}{\Theta}\right)\right| = 1 + \frac{1}{\Theta} - \varepsilon' + 2 + \frac{2}{\Theta} - \varepsilon' = 3 + \frac{3}{\Theta} - 2\varepsilon'.$$

Thus, SmartStart serves $\sigma_1^{(2)}$ after serving $\sigma_1^{(1)}$, and, for all $t \ge 1 + \frac{\varepsilon'}{2}$, we obtain

$$L\left(t, p_1, \{\sigma_1^{(1)}, \sigma_1^{(2)}\}\right) = L\left(t, 1, \{\sigma_1^{(1)}, \sigma_1^{(2)}\}\right) = 3 + \frac{3}{\Theta} - 2\varepsilon'.$$

By assumption, we have $\Theta \le \frac{1}{2}(1+\sqrt{13})$ and $\varepsilon < \frac{1}{5\Theta}\frac{3\Theta^2-\Theta}{3\Theta+3}$, i.e., $\varepsilon' < \frac{1}{5\Theta} < 1$, which implies that for the time $1+\frac{\varepsilon'}{2}$, when SMARTSTART reaches position $p_1 = 1$, the inequality

$$\begin{aligned}
\frac{L\left(1+\frac{\varepsilon'}{2}, p_1, \{\sigma_1^{(1)}, \sigma_1^{(2)}\}\right)}{\Theta-1} &= \frac{3+\frac{3}{\Theta}-2\varepsilon'}{\Theta-1} \\
&= \frac{3-2\varepsilon'}{\Theta-1} + \frac{3}{\Theta(\Theta-1)} \\
&\overset{1<\Theta\leq\frac{1}{2}(1+\sqrt{13})}{\geq} \frac{3-2\varepsilon'}{\frac{1}{2}(\sqrt{13}-1)} + \frac{3}{\frac{1}{4}(\sqrt{13}-1)(1+\sqrt{13})} \\
&= \frac{3-2\varepsilon'}{\frac{1}{2}(\sqrt{13}-1)} + 1 \\
&\overset{\frac{1}{2}(\sqrt{13}-1)<2}{>} 1+\frac{\varepsilon'}{2}
\end{aligned}$$

holds. Thus, SMARTSTART has a waiting period and starts schedule S_1 at time

$$\begin{aligned}
t_1 &= \min\left\{ t \geq 1+\frac{\varepsilon'}{2} \;\middle|\; \frac{L(t, p_1, \{\sigma_1^{(1)}, \sigma_1^{(2)}\})}{\Theta-1} \leq t \right\} \\
&= \min\left\{ t \geq 1+\frac{\varepsilon'}{2} \;\middle|\; \frac{3+\frac{3}{\Theta}-2\varepsilon'}{\Theta-1} \leq t \right\} \\
&= \frac{3+\frac{3}{\Theta}-2\varepsilon'}{\Theta-1} \\
&= \frac{3\Theta+3}{\Theta(\Theta-1)} - \frac{2\varepsilon'}{\Theta-1}.
\end{aligned}$$

Next, we let the final request

$$\sigma_2 = \left(\frac{3\Theta+3}{\Theta(\Theta-1)} - \frac{2}{\Theta} - \varepsilon', \frac{3\Theta+3}{\Theta(\Theta-1)} - \frac{2}{\Theta} - \varepsilon'; \frac{3\Theta+3}{\Theta(\Theta-1)} \right)$$

appear. SMARTSTART finishes schedule S_1 at time

$$t_1 + L(t_1, p_1, \{\sigma_1^{(1)}, \sigma_1^{(2)}\}) = \frac{3\Theta+3}{\Theta(\Theta-1)} - \frac{2\varepsilon'}{\Theta-1} + 3 + \frac{3}{\Theta} - 2\varepsilon' = \frac{3\Theta+3}{\Theta-1} - \frac{2\Theta\varepsilon'}{\Theta-1}$$

at position $p_2 = -\frac{1}{\Theta}$. For all $t \geq \frac{3\Theta+3}{\Theta-1} - \frac{2\Theta}{\Theta-1}\varepsilon'$, we obtain

$$L\left(t, -\frac{1}{\Theta}, \{\sigma_2\}\right) = \frac{3\Theta+3}{\Theta(\Theta-1)} - \frac{1}{\Theta} - \varepsilon'.$$

By assumption, we have $2 \leq \Theta \leq \frac{1}{2}(1+\sqrt{13}) < 3$ and $\varepsilon < \frac{1}{5\Theta}\frac{3\Theta^2-\Theta}{3\Theta+3}$, i.e., $\varepsilon' < \frac{1}{5\Theta}$, which implies that, for the finishing time $\frac{3\Theta+3}{\Theta-1} - \frac{2\Theta\varepsilon'}{\Theta-1}$ of schedule S_1, the inequality

$$\begin{aligned}
\frac{L\left(\frac{3\Theta+3}{\Theta-1} - \frac{2\Theta\varepsilon'}{\Theta-1}, -\frac{1}{\Theta}, \{\sigma_2\}\right)}{\Theta-1} &= \frac{3\Theta+3}{\Theta(\Theta-1)^2} - \frac{1+\Theta\varepsilon'}{\Theta(\Theta-1)} \\
&\overset{\Theta\geq 2}{<} \frac{3\Theta+3}{\Theta-1} - \frac{1+\Theta\varepsilon'}{\Theta(\Theta-1)} \\
&\overset{1>5\Theta\varepsilon'}{<} \frac{3\Theta+3}{\Theta-1} - \frac{6\varepsilon'}{\Theta-1} \\
&\overset{\Theta<3}{<} \frac{3\Theta+3}{\Theta-1} - \frac{2\Theta\varepsilon'}{\Theta-1}
\end{aligned} \qquad (16)$$

holds. (Note that inequality (16) still holds for slightly smaller Θ if we let $\varepsilon \to 0$.) Because of inequality (16), the final schedule S_2 is started at time

$$t_2 = \frac{3\Theta+3}{\Theta-1} - \frac{2\Theta\varepsilon'}{\Theta-1}$$

without waiting. To sum it up, we have

$$\begin{aligned}\textsc{SmartStart}(\sigma) &= t_2 + L(t_2, p_2, \{\sigma_2\})\\ &= \frac{3\Theta+3}{\Theta-1} - \frac{2\Theta\varepsilon'}{\Theta-1} + \frac{3\Theta+3}{\Theta(\Theta-1)} - \frac{1}{\Theta} - \varepsilon'\\ &= \frac{3\Theta+3}{\Theta-1} + \frac{3\Theta+3}{\Theta(\Theta-1)} - \frac{1}{\Theta} - \frac{3\Theta-1}{\Theta-1}\varepsilon'.\end{aligned}$$

On the other hand, $\textsc{Opt}$ goes from the origin straight to position $-\frac{1}{\Theta}$ serving request $\sigma_1^{(2)}$ at time $\frac{1}{\Theta} + \varepsilon'$ (i.e., it has to wait for ε' units of time after it reaches position $-\frac{1}{\Theta}$) and returns to the origin at time $\frac{2}{\Theta} + \varepsilon'$. Let $q > 0$ be the position of a request that has occurred by the application of Lemma 4.1 at the beginning of this proof. Then this request is released earlier than time $q + \frac{\varepsilon'}{2}$. Since $\textsc{Opt}$ reaches position q not earlier than time $\frac{2}{\Theta} + \varepsilon' + q > q + \frac{\varepsilon'}{2}$, $\textsc{Opt}$ can go straight from the origin to the right and can serve all remaining requests without waiting. Note that the position $\frac{3\Theta+3}{\Theta(\Theta-1)} - \frac{2}{\Theta} - \varepsilon'$ of σ_2 is equal to or to right of the position $2 + \frac{1}{\Theta} - \varepsilon'$ of $\sigma_1^{(2)}$ because of $\Theta \le \frac{1}{2}(1+\sqrt{13})$. Thus, $\textsc{Opt}$ finishes at position $\frac{3\Theta+3}{\Theta(\Theta-1)} - \frac{2}{\Theta} - \varepsilon'$ and we have

$$\begin{aligned}\textsc{Opt}(\sigma) &= \left|0 - \left(-\frac{1}{\Theta}\right)\right| + \varepsilon' + \left|-\frac{1}{\Theta} - \left(\frac{3\Theta+3}{\Theta(\Theta-1)} - \frac{2}{\Theta} - \varepsilon'\right)\right|\\ &= \frac{1}{\Theta} + \varepsilon' + \frac{1}{\Theta} + \frac{3\Theta+3}{\Theta(\Theta-1)} - \frac{2}{\Theta} - \varepsilon'\\ &= \frac{3\Theta+3}{\Theta(\Theta-1)}.\end{aligned}$$

Note that $\textsc{Opt}$ can do this even if $c = 1$ since for all requests the starting point is equal to the destination. Since we have $\varepsilon' = \frac{3\Theta+3}{3\Theta^2-\Theta}\varepsilon$, we finally obtain

$$\begin{aligned}\frac{\textsc{SmartStart}(\sigma)}{\textsc{Opt}(\sigma)} &= \frac{\frac{3\Theta+3}{\Theta-1} + \frac{3\Theta+3}{\Theta(\Theta-1)} - \frac{1}{\Theta} - \frac{3\Theta-1}{\Theta-1}\varepsilon'}{\frac{3\Theta+3}{\Theta(\Theta-1)}}\\ &= \Theta + 1 - \frac{\Theta-1}{3\Theta+3} - \frac{3\Theta^2-\Theta}{3\Theta+3}\varepsilon'\\ &= \Theta + 1 - \frac{\Theta-1}{3\Theta+3} - \varepsilon,\end{aligned}$$

as claimed. ◀

Recall that the optimal parameter Θ^* established in Theorem 3.5 is the only positive, real solution of the equation

$$\Theta + 1 - \frac{\Theta-1}{3\Theta+3} = \frac{2\Theta^2+2\Theta}{\Theta^2+\Theta-2},$$

which is $\Theta^* \approx 2.0526$. Therefore, according to Proposition 4.2 and Proposition 4.3 the parameter Θ^* lies in the ranges where the upper bounds of Propositions 3.3 and 3.4 are both tight. It remains to make sure that for all Θ that lie outside of this range the competitive ratio of $\textsc{SmartStart}_\Theta$ is larger than $\rho^* \approx 2.93768$.[†]

▶ **Lemma 4.4.** *Let*

$$I_1 = (1, 2], \quad I_2 = (\tfrac{1}{2}(1 + \sqrt{13}), 1 + \sqrt{2}], \quad I_3 = (1 + \sqrt{2}, 3), \quad I_4 = [3, \infty)$$

be intervals. For every $i \in \{1, 2, 3, 4\}$ there is a set of requests σ, such that, for all $\Theta \in I_i$,

$$\frac{\textsc{Smartstart}(\sigma)}{\textsc{Opt}(\sigma)} > \rho^* \approx 2.93768.$$

Our main theorem now follows from Theorem 3.5 combined with Propositions 4.2 and 4.3, as well as Lemma 4.4.

▶ **Theorem 4.5.** *The competitive ratio of* $\textsc{Smartstart}_{\Theta^*}$ *is exactly*

$$\rho^* = f_1(\Theta^*) = f_2(\Theta^*) \approx 2.93768.$$

For every other $\Theta > 1$ with $\Theta \neq \Theta^*$ the competitive ratio of $\textsc{Smartstart}_{\Theta}$ *is larger than ρ^*.*

Proof. We have shown in Proposition 4.2 that the upper bound

$$\frac{\textsc{Smartstart}(\sigma)}{\textsc{Opt}(\sigma)} \leq f_1(\Theta) = \frac{2\Theta^2 + 2\Theta}{\Theta^2 + \Theta - 2}$$

established in Proposition 3.3 for the case, where Smartstart waits before starting the final schedule, is tight for all $\Theta \in (2, 3)$. Furthermore, we have shown in Proposition 4.3 that the upper bound

$$\frac{\textsc{Smartstart}(\sigma)}{\textsc{Opt}(\sigma)} \leq f_2(\Theta) = \left(\Theta + 1 - \frac{\Theta - 1}{3\Theta + 3}\right)$$

established in Proposition 3.4 for the case, where Smartstart does not wait before starting the final schedule, is tight for all $\Theta \in (2, \frac{1}{2}(1 + \sqrt{13})]$. Since $\Theta^* \approx 2.0526$ lies in those ranges, the competitive ratio of $\textsc{Smartstart}_{\Theta^*}$ is indeed exactly ρ^*.

It remains to show that for every $\Theta > 1$ with $\Theta \neq \Theta^*$ the competitive ratio is larger. First, according to Lemma 4.4, the competitive ratio of Smartstart with parameter $\Theta \in (1, 2]$ or $\Theta \in (\frac{1}{2}(1 + \sqrt{13}), \infty)$ is larger than ρ^*. By monotonicity of f_1, every function value in $(2, \Theta^*)$ is larger than $f_1(\Theta^*) = \rho^*$. Thus, the competitive ratio of Smartstart with parameter $\Theta \in (2, \Theta^*)$ is larger than ρ^*, since f_1 is tight on $(2, \Theta^*)$ by Proposition 4.2. Similarly, by monotonicity of f_2, every function value in $(\Theta^*, \frac{1}{2}(1 + \sqrt{13})]$ is larger than $f_2(\Theta^*) = \rho^*$. Thus, the competitive ratio of Smartstart with parameter $\Theta \in (\Theta^*, \frac{1}{2}(1 + \sqrt{13})]$ is larger than ρ^*, since f_1 is tight on $(\Theta^*, \frac{1}{2}(1 + \sqrt{13})]$ by Proposition 4.3. ◀

5 Lower Bound for the Closed Version

We provide a lower bound for Smartstart for closed online Dial-a-Ride on the line that matches the upper bound given in [1] for arbitrary metric spaces. Note that in this setting, by definition, every schedule of Smartstart is a closed walk that returns to the origin.

▶ **Theorem 5.1.** *The competitive ratio of* Smartstart *for closed online* Dial-a-Ride *on the line with $\Theta = 2$ is exactly 2. For every other $\Theta > 1$ with $\Theta \neq 2$ the competitive ratio of* $\textsc{Smartstart}_{\Theta}$ *is larger than 2.*

Proof. We show that the competitive ratio of $\textsc{SmartStart}_2$ is at least 2 and that the competitive ratio of $\textsc{SmartStart}_\Theta$ is larger than 2 for all $\Theta \neq 2$. From the fact that SMARTSTART is 2-competitive even for general metric spaces [1, Thm. 6], it follows that $\textsc{SmartStart}_2$ has competitive ratio exactly 2 on the line.

Let $\Theta \leq 2$ and consider the set of requests $\{\sigma_1\}$ with $\sigma_1 = (0.5, 0.5; 0)$. Obviously, OPT can serve this request and return to the origin in time $\textsc{Opt}(\{\sigma_1\}) = 1$. Thus, for all $t \geq 0$, we have $L(t, 0, \{\sigma_1\}) = 1$. On the other hand, SMARTSTART waits until time

$$t_1 = \frac{L(t_1, 0, \{\sigma_1\})}{\Theta - 1} = \frac{1}{\Theta - 1}$$

to start its only schedule and finishes at time $\frac{\Theta}{\Theta-1}$. To sum it up, we have

$$\frac{\textsc{SmartStart}(\{\sigma_1\})}{\textsc{Opt}(\{\sigma_1\})} = \frac{\Theta}{\Theta - 1}$$

with $\frac{\Theta}{\Theta-1} > 2$ for all $\Theta < 2$ and $\frac{\Theta}{\Theta-1} = 2$ for $\Theta = 2$. Now let $2 < \Theta \leq 3$ and $\varepsilon \in (0, \min\{1 - \frac{1}{\Theta-1}, \frac{\Theta-2}{2(\Theta-1)}\})$, and consider the set of requests $\{\sigma_1, \sigma_2\}$ with

$$\sigma_1 = (0.5, 0.5; 0) \qquad \text{and} \qquad \sigma_2 = \left(1 - \frac{1}{\Theta - 1} - \varepsilon, 1 - \frac{1}{\Theta - 1} - \varepsilon; \frac{1}{\Theta - 1} + \varepsilon\right).$$

By assumption, we have $\Theta > 2$ and $\varepsilon < 1 - \frac{1}{\Theta-1}$, which implies

$$0 \overset{\varepsilon < 1 - \frac{1}{\Theta-1}}{<} 1 - \frac{1}{\Theta - 1} - \varepsilon \overset{\Theta \leq 3}{<} 0.5,$$

i.e., the position of request σ_2 lies between 0 and 0.5. If OPT moves to position 0.5 and then returns to the origin, it is at position

$$a_2 = 0.5 - \left| \underbrace{\left(\frac{1}{\Theta - 1} + \varepsilon\right)}_{>0.5} - 0.5 \right| = 1 - \frac{1}{\Theta - 1} - \varepsilon$$

at time $r_2 = \frac{1}{\Theta-1} + \varepsilon$. Thus, OPT can serve σ_2 on the way and we have $\textsc{Opt}(\{\sigma_1, \sigma_2\}) = 1$. For all $t \geq 0$, we have $L(t, 0, \{\sigma_1\}) = 1$. Therefore, SMARTSTART waits until time

$$t_1 = \frac{L(t_1, 0, \{\sigma_1\})}{\Theta - 1} = \frac{1}{\Theta - 1}.$$

before starting its first schedule. Since we have $\frac{1}{\Theta-1} < \frac{1}{\Theta-1} + \varepsilon$, SMARTSTART starts to serve σ_1 at time t_1 and returns to the origin at time $\frac{\Theta}{\Theta-1}$. For all $t \geq 0$, we have

$$L(t, 0, \{\sigma_2\}) = 2 - \frac{2}{\Theta - 1} - 2\varepsilon,$$

thus SMARTSTART does not start the second and final schedule before time $\frac{2 - \frac{2}{\Theta-1} - 2\varepsilon}{\Theta-1}$. By assumption, we have $\Theta > 2$, which implies $\frac{\Theta}{\Theta-1} > \frac{2 - \frac{2}{\Theta-1} - 2\varepsilon}{\Theta-1}$. Thus, the second schedule is started at time $t_2 = \frac{\Theta}{\Theta-1}$ and finished at time

$$\textsc{SmartStart}(\{\sigma_1, \sigma_2\}) = \frac{\Theta}{\Theta - 1} + 2 - \frac{2}{\Theta - 1} - 2\varepsilon.$$

To sum it up, we have

$$\begin{aligned}\frac{\textsc{Smartstart}(\{\sigma_1,\sigma_2\})}{\textsc{Opt}(\{\sigma_1,\sigma_2\})} &= \frac{\Theta}{\Theta-1}+2-\frac{2}{\Theta-1}-2\varepsilon \\ &\overset{\varepsilon<\frac{\Theta-2}{2(\Theta-1)}}{>} \frac{3\Theta-4}{\Theta-1}-2\frac{\Theta-2}{2(\Theta-1)} \\ &= 2.\end{aligned}$$

Now let $\Theta > 3$ and $\varepsilon \in (0, 0.5 - \frac{1}{\Theta-1})$, and consider the set of requests $\{\sigma_1, \sigma_2\}$ with

$$\sigma_1 = (0.5, 0.5; 0) \qquad \text{and} \qquad \sigma_2 = \left(0.5, 0.5; \frac{1}{\Theta-1}+\varepsilon\right).$$

By assumption, we have $\varepsilon < 0.5 - \frac{1}{\Theta-1}$, which implies $\frac{1}{\Theta-1} + \varepsilon < 0.5$, i.e., σ_2 is released before position 0.5 is reachable. If Opt moves to position 0.5 and then returns to the origin, it can serve both requests without additional waiting time and we have $\textsc{Opt}(\{\sigma_1, \sigma_2\}) = 1$. For all $t \geq 0$, we have $L(t, 0, \{\sigma_1\}) = 1$. Therefore, Smartstart waits until time

$$t_1 = \frac{L(t_1, 0, \{\sigma_1\})}{\Theta-1} = \frac{1}{\Theta-1}.$$

before starting its first schedule. Since we have $\frac{1}{\Theta-1} < \frac{1}{\Theta-1} + \varepsilon$, Smartstart starts to serve σ_1 at time t_1 and returns to the origin at time $\frac{\Theta}{\Theta-1}$. For all $t \geq 0$, we have

$$L(t, 0, \{\sigma_2\}) = 1,$$

thus Smartstart does not start the second and final schedule before time $\frac{1}{\Theta-1}$. By assumption, we have $\Theta > 3$, which implies $\frac{\Theta}{\Theta-1} > \frac{1}{\Theta-1}$. Thus, the second schedule is started at time $t_2 = \frac{\Theta}{\Theta-1}$ and finished at time

$$\textsc{Smartstart}(\{\sigma_1,\sigma_2\}) = \frac{\Theta}{\Theta-1}+1.$$

To sum it up, we have

$$\frac{\textsc{Smartstart}(\{\sigma_1,\sigma_2\})}{\textsc{Opt}(\{\sigma_1,\sigma_2\})} = \frac{\Theta}{\Theta-1}+1 > 2. \qquad \blacktriangleleft$$

References

1 Norbert Ascheuer, Sven Oliver Krumke, and Jörg Rambau. Online Dial-a-Ride Problems: Minimizing the Completion Time. In *Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 639–650, 2000.

2 Mikhail J. Atallah and S. Rao Kosaraju. Efficient Solutions to Some Transportation Problems with Applications to Minimizing Robot Arm Travel. *SIAM Journal on Computing*, 17(5), 1988.

3 G. Ausiello, E. Feuerstein, S. Leonardi, L. Stougie, and M. Talamo. Algorithms for the On-Line Travelling Salesman. *Algorithmica*, 29(4):560–581, 2001.

4 Antje Bjelde, Yann Disser, Jan Hackfeld, Christoph Hansknecht, Maarten Lipmann, Julie Meißner, Kevin Schewior, Miriam Schlöter, and Leen Stougie. Tight Bounds for Online TSP on the Line. In *Proceedings of the 28th Annual Symposium on Discrete Algorithms (SODA)*, pages 994–1005, 2017.

5 Michiel Blom, Sven O. Krumke, Willem E. de Paepe, and Leen Stougie. The Online TSP Against Fair Adversaries. *INFORMS Journal on Computing*, 13(2):138–148, 2001.

6 Moses Charikar and Balaji Raghavachari. The Finite Capacity Dial-A-Ride Problem. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 458–467, 1998.

7 Willem E. de Paepe, Jan Karel Lenstra, Jiri Sgall, René A. Sitters, and Leen Stougie. Computer-Aided Complexity Classification of Dial-a-Ride Problems. *INFORMS Journal on Computing*, 16(2):120–132, 2004.

8 Esteban Feuerstein and Leen Stougie. On-line Single-server Dial-a-ride Problems. *Theoretical Computer Science*, 268(1):91–105, 2001.

9 Paul C. Gilmore and Ralph E. Gomory. Sequencing a One State-Variable Machine: A Solvable Case of the Traveling Salesman Problem. *Operations Research*, 12(5):655–679, 1964.

10 D. J. Guan. Routing a Vehicle of Capacity Greater Than One. *Discrete Applied Mathematics*, 81(1-3):41–57, 1998.

11 Dietrich Hauptmeier, Sven Oliver Krumke, and Jörg Rambau. The Online Dial-a-Ride Problem Under Reasonable Load. In *Proceedings of the 4th Italian Conference on Algorithms and Complexity (CIAC)*, pages 125–136, 2000.

12 Patrick Jaillet and Michael R. Wagner. Generalized Online Routing: New Competitive Ratios, Resource Augmentation, and Asymptotic Analyses. *Operations Research*, 56(3):745–757, 2008.

13 Sven O. Krumke. Online Optimization Competitive Analysis and Beyond, 2001. Habilitation thesis.

14 Sven O. Krumke, Willem E. de Paepe, Diana Poensgen, Maarten Lipmann, Alberto Marchetti-Spaccamela, and Leen Stougie. On Minimizing the Maximum Flow Time in the Online Dial-a-ride Problem. In *Proceedings of the Third International Conference on Approximation and Online Algorithms (WAOA)*, pages 258–269, 2006.

15 Sven O. Krumke, Luigi Laura, Maarten Lipmann, Alberto Marchetti-Spaccamela, Willem de Paepe, Diana Poensgen, and Leen Stougie. Non-abusiveness Helps: An O(1)-Competitive Algorithm for Minimizing the Maximum Flow Time in the Online Traveling Salesman Problem. In *Proceedings of the 5th International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX)*, pages 200–214, 2002.

16 Maarten Lipmann. *On-Line Routing*. PhD thesis, Technical University Eindhoven, 2003.

17 Maarten Lipmann, Xiwen Lu, Willem E. de Paepe, Rene A. Sitters, and Leen Stougie. On-Line Dial-a-Ride Problems Under a Restricted Information Model. *Algorithmica*, 40(4):319–329, 2004.

18 Fanglei Yi and Lei Tian. On the Online Dial-a-ride Problem with Time-windows. In *Proceedings of the 1st International Conference on Algorithmic Applications in Management (AAIM)*, pages 85–94, 2005.

19 Fanglei Yi, Yinfeng Xu, and Chunlin Xin. Online Dial-a-ride Problem with Time-windows Under a Restricted Information Model. In *Proceedings of the 2nd International Conference on Algorithmic Aspects in Information and Management (AAIM)*, pages 22–31, 2006.