



Improved Bounds for Open Online Dial-a-Ride on the Line

Alexander Birx¹ · Yann Disser²  · Kevin Schewior³ 

Received: 28 January 2020 / Accepted: 17 September 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

We consider the open, non-preemptive online DIAL-A-RIDE problem on the real line, where transportation requests appear over time and need to be served by a single server. We give a lower bound of 2.0585 on the competitive ratio, which is the first bound that strictly separates open online DIAL-A-RIDE on the line from open online TSP on the line in terms of competitive analysis, and is the best currently known lower bound even for general metric spaces. On the other hand, we present an algorithm that improves the best known upper bound from 2.9377 to 2.6662. The analysis of our algorithm is tight.

Keywords Dial-a-ride on the line · Elevator problem · Online algorithms · Competitive analysis · Smartstart · Competitive ratio

1 Introduction

We consider the online DIAL-A-RIDE problem on the line, where transportation requests appear over time and need to be transported to their respective destinations by a single

In part supported by the ‘Excellence Initiative’ of the German Federal and State Governments and the Graduate School CE at TU Darmstadt. Also in part supported by the Independent Research Fund Denmark, Natural Sciences, Grant DFF-0135-00018B.

A preliminary version of this article appeared in [8].

✉ Kevin Schewior
kevs@sdu.dk

Alexander Birx
alexander.birx@gmail.com

Yann Disser
disser@mathematik.tu-darmstadt.de

¹ Darmstadt, Germany

² Institute of Mathematics and Graduate School CE, TU Darmstadt, Darmstadt, Germany

³ Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark

server. More precisely, each request is of the form $\sigma_i = (a_i, b_i; r_i)$ and appears in position $a_i \in \mathbb{R}$ along the real line at time $r_i \geq 0$ and needs to be transported to position $b_i \in \mathbb{R}$. The server starts at the origin, can move at unit speed, and has a capacity $c \in \mathbb{N} \cup \{\infty\}$ that bounds the number of requests it can carry simultaneously. The objective is to minimize the completion time, i.e., the time until all requests have been served. In this paper, we focus on the *non-preemptive* and *open* setting, where the former means that requests can only be unloaded at their destinations, and the latter means that we do not require the server to return to the origin after serving all requests. Note that, in accordance with the literature (e.g., [9], we think of the real line as a horizontal object, with $-\infty$ on the very “left” and $+\infty$ on the very “right”. Our pictures, however, are turned sideways for a more compact representation.

We aim to bound the *competitive ratio* of the problem, i.e., the smallest ratio any online algorithm can guarantee between the completion time of its solution compared to an (offline) optimum solution that knows all requests ahead of time. To date, the best known lower bound of 2.0346 on this ratio was shown by Bjelde et al. [9], already for online TSP, where $a_i = b_i$ for all requests (i.e., requests only need to be visited). The best known upper bound of 2.9377 was achieved by the SMARTSTART algorithm [7]. *Our results.* Our first result is an improved lower bound for online DIAL-A-RIDE on the line. Importantly, since the bound of roughly 2.0346 was shown to be tight for open online TSP [9], our new bound is the first time that open DIAL-A-RIDE on the line can be strictly separated from open online TSP in terms of competitive analysis. In addition, our bound is the currently best known lower bound even for general metric spaces. Specifically, we show the following.

Theorem 1 *Let $\rho \approx 2.0585$ be the second largest root of the polynomial $4\rho^3 - 26\rho^2 + 39\rho - 5$. There is no $(\rho - \varepsilon)$ -competitive algorithm for open, non-preemptive ($c < \infty$) online DIAL-A-RIDE on the line for any $\varepsilon > 0$.*

Our construction is a non-trivial variation of the construction achieving roughly 2.0346 for online TSP [9]. This construction is comprised of an initial request, a first stage consisting in turn of different iterations, and a second stage. We show that, by using a proper transportation request as initial request, we can adapt a single iteration of the first stage as well as the second stage to achieve the bound of roughly 2.0585 in the DIAL-A-RIDE setting.

Our second result is an improved algorithm SMARTERSTART for online DIAL-A-RIDE on the line. This algorithm improves the waiting strategy of the SMARTSTART algorithm, which was identified as a weakness in [7]. We show that this modification improves the competitive ratio of the algorithm and give a tight analysis. Specifically, we show the following.

Theorem 2 *The competitive ratio of SMARTERSTART is (roughly) 2.6662.*

The general idea of SMARTERSTART is to improve the tradeoff between the case when the algorithm waits before starting its final schedule and the case when it starts the final schedule immediately. Our modification of SMARTSTART significantly improves the performance in the former case, while only moderately degrading the performance in the latter case. Overall, this results in an improved worst-case performance.

Table 1 Overview of the best known bounds for online DIAL-A-RIDE on the line (top), and online DIAL-A-RIDE on general metric spaces (bottom)

	Open		Closed	
	Lower bound	Upper bound	Lower bound	Upper bound
<i>Line</i>				
Non-preemptive	2.0585 (Thm. 1)	2.6662 (Thm. 2)	1.75 [9]	2
Preemptive	2.04	2.41 [9]	1.64	2
TSP	2.04 [9]	2.04 [9]	1.64 [3]	1.64 [9]
<i>General</i>				
Non-preemptive	2.0585 (Thm. 1)	3.41 [18]	2	2 [1, 13]
Preemptive	2.04	3.41	2	2
TSP	2.04	2.5 [3]	2 [3]	2 [3]

Results are split into the non-preemptive case (with $c < \infty$), the preemptive case, and the TSP-case, where source and destination of each request coincide. Bold results are original, all other results follow immediately

Related work. The online DIAL-A-RIDE problem has received considerable attention in the past (e.g. [1, 7, 9, 10, 13, 17]). Table 1 gives an overview of the currently best known bounds on the line for open online DIAL-A-RIDE and its special case open online TSP. Note that we only consider the case of bounded capacity $c < \infty$. For $c = \infty$, the best known lower bound is 2.04 (via open online TSP) and the best known upper bound is 2.41 [9, Theorem 6.3].

The following results are known for *closed* online DIAL-A-RIDE: For general metric spaces, the competitive ratio is exactly 2, both for online DIAL-A-RIDE as well as online TSP [1, 3, 13]. On the line, a better upper bound is known only for online TSP, where the competitive ratio is exactly $(9 + \sqrt{17})/8 \approx 1.6404$ [3, 9]. The best known lower bound for closed, non-preemptive DIAL-A-RIDE on the line is 1.75 [9].

When the objective is to minimize the maximum (or average) flow time, on many metric spaces no online algorithm can be competitive [19, 21]. Hauptmeier et al. [16] showed that a competitive algorithm is possible if we restrict ourselves to instances with “reasonable” load. Krumke et al. [20] and Bienkowski et al. [4, 5] minimized the sum of completion times. Yi and Tian [23] considered online DIAL-A-RIDE with deadlines, where the objective is to maximize the number of requests that are served in time. Other interesting variants of online DIAL-A-RIDE where destinations of requests are only revealed upon their collection were studied by Lipmann et al. [22] as well as Yi and Tian [24].

For an overview of results for the offline version of DIAL-A-RIDE on the line, see [12]. Without release times, Gilmore and Gomory [14] and Atallah and Kosaraju [2] gave a polynomial time algorithm for closed, non-preemptive DIAL-A-RIDE on the line with capacity $c = 1$. Guan [15] showed that the closed, non-preemptive problem is hard for $c = 2$, and Bjelde et al. [9] extended this result for any finite capacity $c \geq 2$ in both the open and the closed variant. Bjelde et al. [9] also showed that the problem with release times is already hard for finite $c \geq 1$ in both variants, and Krumke [18] gave a 3-approximation algorithm for the closed variant. The complexity for the case $c = \infty$

remains open. For closed, preemptive DIAL-A-RIDE on the line without release times, Atallah and Kosaraju [2] gave a polynomial time algorithm for $c = 1$ and Guan [15] for $c \geq 2$. Charikar and Raghavachari [11] presented approximation algorithms for the closed case without release times on general metric spaces.

2 General Lower Bound

In this section, we prove Theorem 1. Let $c < \infty$ and ALG be a deterministic online algorithm for open online DIAL-A-RIDE. Let $\rho \approx 2.0585$, be the second largest root of the polynomial $4\rho^3 - 26\rho^2 + 39\rho - 5$. We describe a request sequence σ_ρ such that $\text{ALG}(\sigma_\rho) \geq \rho \text{OPT}(\sigma_\rho)$.

We first give a high-level description of our construction disregarding many technical details. Our construction is based on that in [9] for the TSP version of the problem. That construction consists of two *stages*: After an initial request $(1, 1; 1)$ (assuming w.l.o.g. ALG's position at time 1 is at most 0), the first stage starts. This stage consists of a loop, which ends as soon as two so-called critical requests are established. The second stage consists of augmenting the critical requests by suitable additional ones to show the desired competitive ratio. A single iteration of the loop only yields a lower bound of roughly 2.0298, but as the number of iterations approaches infinity one can show the tight bound of roughly 2.0346 in the limit.

In the DIAL-A-RIDE setting, we show a lower bound of roughly 2.0585 using the same general structure but only a single iteration. Our additional leeway stems from replacing the initial request $(1, 1; 1)$ with c initial requests of the form $(1, \delta; 1)$ where $\delta > 1$: At the time when an initial request is loaded, we show that w.l.o.g. all c requests are loaded and then proceed as we did when $(1, 1; 1)$ was served. In the new situation, the algorithm has to first deliver the c initial requests to be able to serve additional requests. For the optimum, the two situations however do not differ, because in the new situation there will be an additional request to the right of δ later anyway. Interestingly, this leeway turns out to be sufficient not only to create critical requests (w.r.t. a slightly varied notion of criticality) for a competitive ratio of larger than 2.0298 but even strictly larger than 2.0346. The second stage has to be slightly adapted to match the new notion of criticality. It remains unclear how to use multiple iterations in our setting.

We start by making observations that will simplify the exposition. Consider a situation in which the server is fully loaded.¹ First note that it is essentially irrelevant whether we assume that the server, without delivering any of the loaded requests, can still serve requests $(a_i, b_i; t_i)$ for which $a_i = b_i$: If it can, we simply move a_i and b_i by $\varepsilon > 0$ apart, forbidding the server to serve it before delivering one of the loaded requests first. Therefore, we assume for simplicity that, when fully loaded, the server has to first deliver a request before it can serve any other one. We note that, in our construction, the above idea can be implemented without loss, not even in terms of ε since we are applying the above idea to $O(1)$ requests.

¹ Note that this crucially needs that the capacity is bounded.

The latter discussion also motivates restricting the space of considered algorithms: We call ALG *eager* if it, when fully loaded with requests with identical destinations, immediately delivers these requests without detour. It is clear that we can transform every algorithm ALG' into an eager algorithm ALG'_{eager} by letting it deliver the requests right away, waiting until ALG' would have delivered them, and then letting it continue like ALG' . Since ALG' cannot collect or serve other requests while being fully loaded, we have $ALG'_{eager}(\sigma) \leq ALG'(\sigma)$ for every request sequence σ .

Observation 1 *Every algorithm for online DIAL-A-RIDE can be turned into an eager algorithm with the same competitive ratio.*

Thus, we may assume that ALG is eager. We now give some intuition on critical requests. Suppose that we have two requests $\sigma^R = (t^R, t^R; t^R)$ and $\sigma^L = (-t^L, -t^L, t^L)$ with $t^L \leq t^R$ to the right and to the left of the origin, respectively, and that ALG serves σ^R first at some time $t^* \geq (2\rho - 2)t^L + (\rho - 2)t^R$. Now suppose we could force ALG to serve σ^L directly after σ^R , even if additional requests are released. Then we could just release the request $\sigma_*^R = (t^R, t^R, 2t^L + t^R)$ and we would have

$$ALG(\sigma_\rho) = t^* + 2t^L + 2t^R \geq 2\rho t^L + \rho t^R = \rho OPT(\sigma_\rho),$$

since OPT can serve the three requests in time $2t^L + t^R$ by serving σ^L first. In fact, we will show that we can force ALG into this situation (or a comparably bad situation) if the requests $\sigma^R = (t^R, t^R; t^R)$ and $\sigma^L = (-t^L, -t^L, t^L)$ satisfy the following properties. To describe the trajectory of a server, we use the notation “move(a)” for the tour that moves the server from its current position with unit speed to the point $a \in \mathbb{R}$.

Definition 1 We call the last two requests $\sigma^R = (t^R, t^R; t^R)$ and $\sigma^L = (-t^L, -t^L; t^L)$ of a request sequence with $0 < t^L \leq t^R$ *critical* for ALG if the following conditions hold:

- (i) Both tours $\text{move}(-t^L) \oplus \text{move}(t^R)$ and $\text{move}(t^R) \oplus \text{move}(-t^L)$ (started at time 0) serve all requests presented until time t^R .
- (ii) ALG serves both σ^R and σ^L after time t^R and ALG’s position at time t^R lies between t^R and $-t^L$.
- (iii) If ALG serves σ^R before σ^L , it does so no earlier than $t_*^R := (2\rho - 2)t^L + (\rho - 2)t^R$.
- (iv) If ALG serves σ^L before σ^R , it does so no earlier than $t_*^L := (2\rho - 2)t^R + (\rho - 2)t^L$.
- (v) It holds that $\frac{t_*^R}{t^L} \leq \frac{4\rho^2 - 30\rho + 50}{-8\rho^2 + 50\rho - 66} \approx 1.717$.

Lemma 1 *If there is a request sequence with two critical requests for ALG, we can release additional requests such that ALG is not $(\rho - \varepsilon)$ -competitive on the resulting instance for any $\varepsilon > 0$.*

Definition 1 differs from [9, Definition 4.2] only in the right-hand side of property (v), which is 2 in the original paper. Lemma 1 has been proved in [9, Lemma 4.3] for

slightly smaller values of ρ but request sequences that satisfy the weaker properties of [9, Definition 4.2], however, a careful inspection of the proof of [9, Lemma 4.3] shows that the statement of Lemma 1 also holds in our setting.

2.1 Analysis of the First Stage

In this subsection, we describe the first stage. Recall that our goal in the first stage is to construct a request sequence σ_ρ that satisfies all properties of Definition 1.

Here and throughout, we let $\text{pos}(t)$ denote the position of ALG’s server at time t . We assume w.l.o.g. that $\text{pos}(1) \leq 0$ (the other case is symmetric). Now, let

$$\delta := \frac{3\rho^2 - 11}{-3\rho^3 + 15\rho - 4} \approx 2.414$$

and let c initial requests $\sigma_{(j)}^R = (1, \delta; 1)$ with $j \in \{1, \dots, c\}$ appear. These are the only requests appearing in the entire construction with a starting point differing from the destination. We make a basic observation on how ALG has to serve these requests.

Lemma 2 *ALG cannot collect any of the requests $\sigma_{(j)}^R$ before time 2. If ALG collects the requests after time $\rho\delta - (\delta - 1)$ or serves $c' < c$ requests before loading the remaining $c - c'$, it is not $(\rho - \varepsilon)$ -competitive.*

Proof ALG cannot collect any $\sigma_{(j)}^R$ before time 2 since its position at time 1 is $\text{pos}(1) \leq 0$. Moreover, ALG is not $(\rho - \varepsilon)$ -competitive if it collects one of the requests after time $\rho\delta - (\delta - 1)$, since it cannot finish before time $\rho\delta$ and we have

$$\text{ALG}(\{\sigma_{(j)}^R\}_{j \in \{1, \dots, c\}}) \geq \rho\delta = \rho \text{OPT}(\{\sigma_{(j)}^R\}_{j \in \{1, \dots, c\}}).$$

Assume ALG serves $c' < c$ requests before loading the remaining $c - c'$. Then, because of

$$\delta = \frac{3\rho^2 - 11}{-3\rho^3 + 15\rho - 4} \stackrel{\rho > 2.032}{>} \frac{1}{3 - \rho}, \tag{1}$$

we have

$$\text{ALG}(\{\sigma_R^0\}) \geq 1 + \delta + 2(\delta - 1) \stackrel{(1)}{>} \rho\delta = \rho \text{OPT}(\{\sigma_R^0\}).$$

This completes the proof. □

We hence may assume that ALG loads all c requests $\sigma_{(j)}^R$ at the same time. Let $t^L \in [2, \rho\delta - (\delta - 1))$ be the time ALG loads the c requests $\sigma_{(j)}^R$. We start the first stage and present a variant of a single iteration of the construction in [9]: We let the request $\sigma^L = (-t^L, -t^L; t^L)$ appear and define the function

$$\ell(t) = (4 - \rho) \cdot t - (2\rho - 2) \cdot t^L,$$

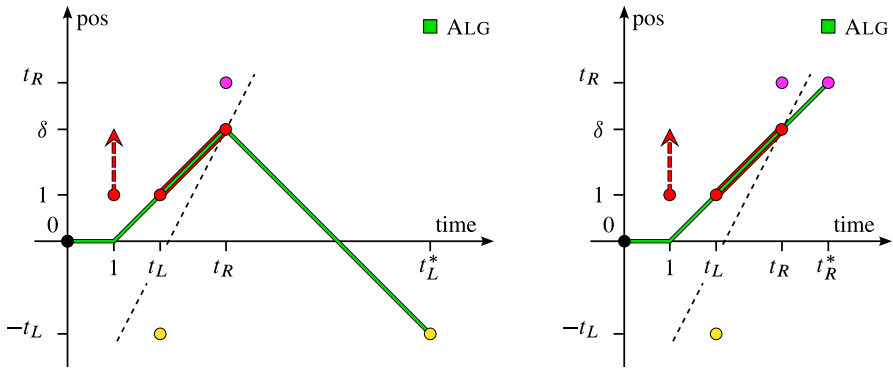


Fig. 1 Left: ALG (green) serves σ_L (yellow) before σ_R (violet) at time t_L^* . Right: ALG serves σ_R before the request σ_L at time t_R^* . Both: The requests σ_0^j are red and the line ℓ is the dashed black line (Color figure online)

which can be viewed as a line in the path-time diagram. Because of $\rho > 2$, we have $\ell(t^L) = (6 - 3\rho)t^L < 0 < \text{pos}(t^L)$, i.e., ALG’s position at time t^L is to the right of the line ℓ . Thus, ALG crosses the line ℓ before it serves σ^L . Let t^R be the time ALG crosses ℓ for the first time and let the request $\sigma^R = (t^R, t^R; t^R)$ appear. Assume ALG crosses the line ℓ and serves σ^R before σ^L . Then it does not serve σ^R before time

$$t^R + |\ell(t^R) - t^R| = (2\rho - 2)t^L + (\rho - 2)t^R = t_*^R, \tag{2}$$

where we use that $\ell(t^R) = \text{pos}(t^R) \leq t^R$ for the first equality. Now assume ALG crosses ℓ at time $t^R \geq \frac{3\rho-5}{7-3\rho}t^L$ and serves σ^L before σ^R . Then it does not serve σ^L before time

$$\begin{aligned} t^R + |\ell(t^R) - (-t^L)| &= (5 - \rho)t^R - (2\rho - 3)t^L \\ &\geq (2\rho - 2)t^R + (7 - 3\rho)\frac{3\rho - 5}{7 - 3\rho}t^L - (2\rho - 3)t^L \\ &= (2\rho - 2)t^R + (\rho - 2)t^L = t_*^L. \end{aligned} \tag{3}$$

See Fig. 1 for an illustration. The following lemma shows that the two requests cannot be served before these respective times by establishing that indeed $t^R \geq \frac{3\rho-5}{7-3\rho}t^L$.

Lemma 3 ALG can neither serve σ^L before time t_*^L nor can it serve σ^R before time t_*^R .

Proof Since ALG is eager, it delivers the c requests $\sigma_{(j)}^R$ without waiting or detour, i.e., we have $\text{pos}(t^L + (\delta - 1)) = \delta$. Furthermore, we have

$$\begin{aligned} \ell(t^L + (\delta - 1)) &= (4 - \rho)(t^L + (\delta - 1)) - (2\rho - 2)t^L \\ &= (6 - 3\rho)t^L + (4 - \rho)(\delta - 1) \\ &\leq (6 - 3\rho)(\rho\delta - (\delta - 1)) + (4 - \rho)(\delta - 1) \end{aligned}$$

$$= \frac{3\rho^4 - 18\rho^3 + 3\rho^2 + 50\rho - 14}{3\rho^3 - 15\rho + 4}$$

$$\stackrel{\rho < 2.06}{<} \delta = \text{pos} \left(t^L + (\delta - 1) \right),$$

i.e., ALG’s position at time $t^L + (\delta - 1)$ is to the right of ℓ . The earliest possible time ALG crosses ℓ is the solution of

$$\ell(t^R) = (4 - \rho)t^R - (2\rho - 2)t^L = \text{pos} \left(t^L + (\delta - 1) \right) + t^L + (\delta - 1) - t^R,$$

which is $t^R = \frac{2\rho-1}{5-\rho}t^L + \frac{2\delta-1}{5-\rho}$. The inequality

$$\begin{aligned} \left(\frac{3\rho - 5}{7 - 3\rho} - \frac{2\rho - 1}{5 - \rho} \right) t^L &= \frac{3\rho^2 + 3\rho - 18}{3\rho^2 - 22\rho + 35} t^L \\ &\leq \frac{3\rho^2 + 3\rho - 18}{3\rho^2 - 22\rho + 35} (\rho\delta - (\delta - 1)) \\ &= \frac{3\rho^3 + 6\rho^2 - 15\rho - 18}{3\rho^4 - 15\rho^3 - 15\rho^2 + 79\rho - 20} \\ &= \frac{2\delta - 1}{5 - \rho}, \end{aligned}$$

implies that we have

$$t^R \geq \frac{3\rho - 5}{7 - 3\rho} t^L. \tag{4}$$

Because of inequality (2) ALG does not serve σ^R before t_*^R and because of the inequalities (4) and (3) it does not serve σ^L before time t_*^L . \square

In fact, also the other properties of critical requests are satisfied.

Lemma 4 *The requests σ^R and σ^L of the request sequence σ_ρ are critical.*

Proof We have to show that the requests σ^R and σ^L of the request sequence σ_ρ satisfy the properties (i) to (v) of Definition 1. The release time of every request is equal to its starting position, thus every request can be served/loaded immediately once its starting position is visited and property (i) of Definition 1 is satisfied. At time t^R ALG has not served σ^R , because for that it would have needed to go right from time 0 on; it has not served σ^L either, because during the period of time $[t_L, t_R]$ ALG and σ^L were on different sides of ℓ . This establishes the first part of (ii) of Definition 1. Furthermore at time t^R ALG is at position $\text{pos} (t^R) = (4 - \rho)t^R - (2\rho - 2)t^L$ with

$$-t^L \leq (4 - \rho)t^R - (2\rho - 2)t^L \leq t^R.$$

Therefore, the second part of (ii) of Definition 1 is satisfied as well.

Lemma 3 shows that (iii) and (iv) of Definition 1 are satisfied. It remains to show that property (v) is satisfied. For this we need to examine the release time t^R of σ^R . The time t^R is largest if ALG tries to avoid crossing the line ℓ for as long as possible, i.e., it continues to move right after serving the requests $\sigma_{(j)}^R$. Then, we have $\text{pos}(t) = 1 - t^L + t$ for $t \in [t^L, t^R]$ and t^R is the solution of

$$1 - t^L + t^R = (4 - \rho)t^R - (2\rho - 2)t^L.$$

Thus, in general, we have $t^R \leq \frac{2\rho-3}{3-\rho}t^L + \frac{1}{3-\rho}$, i.e.,

$$\frac{t^R}{t^L} \leq \frac{2\rho - 3}{3 - \rho} + \frac{1}{(3 - \rho)t^L} \stackrel{t^L \geq 2}{\leq} \frac{4\rho - 5}{6 - 2\rho}. \tag{5}$$

For property (v), we need $\frac{t^R}{t^L} \leq \frac{4\rho^2-30\rho+50}{-8\rho^2+50\rho-66}$. This is satisfied if

$$\frac{4\rho - 5}{6 - 2\rho} \leq \frac{4\rho^2 - 30\rho + 50}{-8\rho^2 + 50\rho - 66},$$

which, for our value of $\rho \approx 2.0585$, is equivalent to

$$4\rho^3 - 26\rho^2 + 39\rho - 5 \geq 0,$$

which is true by definition of ρ . □

This concludes the description of the first stage.

2.2 Analysis of the Second Stage

In this subsection, we prove Lemma 1 by describing the second stage and showing that it fulfills its purpose. Let the requests σ^L and σ^R be critical. Furthermore, let $p_0 \in \{-t^L, t^R\}$ be the starting position of the request $\sigma_0 \in \{\sigma^L, \sigma^R\}$ that is served first by ALG and let $p_1 \in \{-t^L, t^R\}$ be the starting position of the request $\sigma_1 \in \{\sigma^L, \sigma^R\}$ that is not served first by ALG. By properties (iii) and (iv) of Definition 1, ALG cannot serve σ_0 before time $(2\rho - 2)|p_1| + (\rho - 2)|p_0|$. Thus, we have

$$\text{ALG}(\sigma_\rho) \geq (2\rho - 2)|p_1| + (\rho - 2)|p_0| + |p_0 - p_1| = (2\rho - 1)|p_1| + (\rho - 1)|p_0|. \tag{6}$$

We have equality in inequality (6) if ALG serves σ_0 at the earliest possible time and then moves directly to position p_1 . However, in general ALG does not need to do this and instead can wait. At time $t \geq \max\{|p_0|, |p_1|\}$, we have

$$\text{ALG}(\sigma_\rho) \geq t + |\text{pos}(t) - p_0| + |p_0 - p_1|$$

if ALG still has to serve σ_0 and

$$\text{ALG}(\sigma_\rho) \geq t + |\text{pos}(t) - p_1|$$

if σ_0 is served and only σ_1 is left to be served. We want to measure the delay of ALG at a time $t \geq \max\{|p_0|, |p_1|\}$, i.e., the difference between the time ALG needs at least to serve both requests σ_0 and σ_1 and the time $(2\rho - 1)|p_1| + (\rho - 1)|p_0|$. We define for $t \geq \max\{|p_0|, |p_1|\}$ the function

$$\text{delay}(t) := \begin{cases} t + |\text{pos}(t) - p_0| - (\rho - 2)|p_0| - (2\rho - 2)|p_1| & \text{if } \sigma_0 \text{ is not} \\ & \text{served at } t, \\ t + |\text{pos}(t) - p_1| - (\rho - 1)|p_0| - (2\rho - 1)|p_1| & \text{if } \sigma_0 \text{ is served} \\ & \text{at } t, \text{ but } \sigma_1 \text{ not,} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

We make the following observation about delay.

Observation 2 *Let $t \geq \max\{|p_0|, |p_1|\}$ be a time at which σ_1 is not served yet. The earliest time ALG can serve σ_1 is $(2\rho - 1)|p_1| + (\rho - 1)|p_0| + \text{delay}(t)$.*

Lemma 5 *There is a $W \geq 0$ with*

$$\text{delay}\left(2|p_1| + |p_0| + \frac{W}{\rho - 1}\right) = W$$

Proof Because of property (ii) of Definition 1, at time $\max\{|p_0|, |p_1|\}$ neither σ_0 nor σ_1 has been served by ALG yet. Since ALG serves σ_1 after σ_0 , and since $\max\{|p_0|, |p_1|\} = t^R$, the request σ_1 is not served before time

$$\max\{|p_0|, |p_1|\} + |p_0| + |p_1| \geq 2|p_1| + |p_0|,$$

i.e. $\text{delay}(2|p_1| + |p_0|)$ is defined. Because of properties (iii) and (iv) of Definition 1, σ_0 is not served before time $(2\rho - 2)|p_1| + (\rho - 2)|p_0|$. Thus, for $t \geq (2\rho - 2)|p_1| + (\rho - 2)|p_0|$, we have $\text{delay}(t) \geq 0$. We have

$$\begin{aligned} 2|p_1| + |p_0| &\stackrel{\text{Def 1 (v)}}{\geq} 2|p_1| + (3 - \rho) \frac{-8\rho^2 + 50\rho - 66}{4\rho^2 - 30\rho + 50} |p_1| + (\rho - 2)|p_0| \\ &\stackrel{2 < \rho < 2.5}{>} (2\rho - 2)|p_1| + (\rho - 2)|p_0|, \end{aligned} \tag{7}$$

i.e. $\text{delay}(2|p_1| + |p_0|) \geq 0$. If $\text{delay}(2|p_1| + |p_0|) = 0$, we have $W = 0$ and are done. Otherwise, by inequality (7), we have $\text{delay}(2|p_1| + |p_0|) > 0$. Note that ALG needs to serve σ_1 at some point to be $(\rho - \varepsilon)$ -competitive. Let $W^* \geq 0$ be chosen such that ALG serves σ_1 at time $2|p_1| + |p_0| + W^*/(\rho - 1)$; such a value exists again because σ_1

has not been served by ALG at time $2|p_1| + |p_0|$. Therefore

$$\text{delay} \left(2|p_1| + |p_0| + \frac{W^*}{\rho - 1} - \varepsilon' \right)$$

is defined for some sufficiently small $\varepsilon' \leq |p_1|$. Define the function

$$f(W) := \text{delay} \left(2|p_1| + |p_0| + \frac{W}{\rho - 1} \right) - W.$$

Note that f is continuous and we have $f(0) > 0$. If

$$\text{delay} \left(2|p_1| + |p_0| + \frac{W^*}{\rho - 1} - \varepsilon' \right) \leq \frac{W^*}{\rho - 1} - \varepsilon' \stackrel{\rho > 2}{\leq} W^* - (\rho - 1)\varepsilon',$$

we have $f(W^* - (\rho - 1)\varepsilon') < 0$ and we find W in the interval $(0, W^* - (\rho - 1)\varepsilon']$. Otherwise, we have

$$\text{delay} \left(2|p_1| + |p_0| + \frac{W^*}{\rho - 1} - \varepsilon' \right) > \frac{W^*}{\rho - 1} - \varepsilon'.$$

By Observation 2, ALG has not served σ_1 at time

$$(2\rho - 1)|p_1| + (\rho - 1)|p_0| + \frac{W^*}{\rho - 1} - \varepsilon' \stackrel{\rho > 2, \varepsilon' \leq |p_1|}{>} 2|p_1| + |p_0| + \frac{W^*}{\rho - 1}.$$

This is a contradiction to the fact that W^* was chosen such that ALG serves σ_1 at time $2|p_1| + |p_0| + \frac{W^*}{\rho - 1}$. □

Lemma 6 *Let $W \geq 0$ with*

$$\text{delay} \left(2|p_1| + |p_0| + \frac{W}{\rho - 1} \right) = W.$$

Then ALG serves σ_0 no later than time $2|p_1| + |p_0| + \frac{W}{\rho - 1}$.

Proof Assume we have

$$2|p_1| + |p_0| + \frac{W}{\rho - 1} \geq (2\rho - 2)|p_1| + (\rho - 2)|p_0| + W. \tag{8}$$

Then, by definition of W and Observation 2, ALG can serve σ_1 at time

$$\begin{aligned} &(2\rho - 1)|p_1| + (\rho - 1)|p_0| + \text{delay} \left(2|p_1| + |p_0| + \frac{W}{\rho - 1} \right) \\ &= (2\rho - 1)|p_1| + (\rho - 1)|p_0| + W. \end{aligned} \tag{9}$$

Because of inequality (8), this can only be the case if ALG serves σ_0 no later than time

$$(2\rho - 1)|p_1| + (\rho - 1)|p_0| + W - |p_1| - |p_0| = (2\rho - 2)|p_1| + (\rho - 2)|p_0| + W \stackrel{(8)}{\leq} 2|p_1| + |p_0| + \frac{W}{\rho - 1}.$$

Thus, it remains to show inequality (8). Because of property (i) of Definition 1 all requests can be served by the tours $\text{move}(p_0) \oplus \text{move}(p_1)$ and $\text{move}(p_1) \oplus \text{move}(p_0)$. By Observation 2 and by (9), we have $\text{ALG}(\sigma_\rho) \geq (2\rho - 1)|p_1| + (\rho - 1)|p_0| + W$. Thus, if we have

$$\begin{aligned} \text{ALG}(\sigma_\rho) &\geq (2\rho - 1)|p_1| + (\rho - 1)|p_0| + W \\ &> (\rho - \varepsilon)(2|p_1| + |p_0|) \\ &\geq (\rho - \varepsilon)\text{OPT}(\sigma_\rho), \end{aligned}$$

ALG is not $(\rho - \varepsilon)$ -competitive. Therefore, we may assume

$$(2\rho - 1)|p_1| + (\rho - 1)|p_0| + W \leq (\rho - \varepsilon)(2|p_1| + |p_0|),$$

and thus

$$\begin{aligned} W &\leq (\rho - \varepsilon)(2|p_1| + |p_0|) - (2\rho - 1)|p_1| - (\rho - 1)|p_0| \\ &= (1 - 2\varepsilon)|p_1| + (1 - \varepsilon)|p_0| \\ &< |p_1| + |p_0|. \end{aligned} \tag{10}$$

Inequality (8) now is equivalent to the inequality

$$\begin{aligned} &\frac{2|p_1| + |p_0| - ((2\rho - 2)|p_1| + (\rho - 2)|p_0|)}{1 - \frac{1}{\rho - 1}} \\ &= \frac{(\rho - 1)((4 - 2\rho)|p_1| + (3 - \rho)|p_0|)}{\rho - 2} \\ &= \frac{(\rho - 1)(4 - 2\rho)}{\rho - 2}|p_1| + \frac{(\rho - 1)(3 - \rho)}{\rho - 2}|p_0| \\ &\stackrel{\text{Def 1 (v)}}{\geq} |p_0| + (2 - 2\rho)|p_1| \\ &\quad + \frac{(-\rho^2 + 3\rho - 1)(-8\rho^2 + 50\rho - 66)}{(\rho - 2)(4\rho^2 - 30\rho + 50)}|p_1| \\ &\geq |p_0| + \frac{5\rho^3 - 36\rho^2 + 86\rho - 67}{2\rho^3 - 19\rho^2 + 55\rho - 50}|p_1| \\ &\stackrel{2 < \rho < 2.5}{>} |p_0| + |p_1| \\ &\stackrel{(10)}{>} W \end{aligned}$$

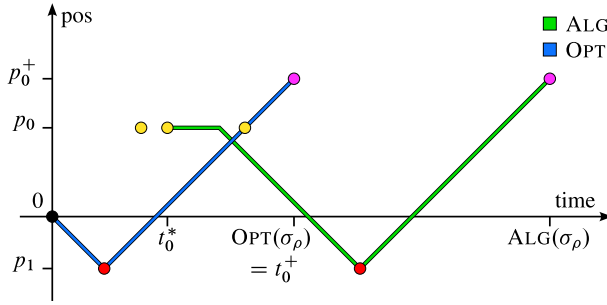


Fig. 2 Case 1: ALG (green) serves σ_1 (red) before σ_0^+ (violet). We assume delay(t) = $W = 2$ for $t \geq t_0^* + W$, where t_0^* is the earliest time that ALG can serve σ_0 . OPT is blue and the request σ_0 is yellow (Color figure online)

if we solve inequality (8) for W . □

Now we have all ingredients to prove Lemma 1.

Proof of Lemma 1 Let $W \geq 0$ with

$$\text{delay} \left(2|p_1| + |p_0| + \frac{W}{\rho - 1} \right) = W.$$

We present the request

$$\begin{aligned} \sigma_0^+ &= (p_0^+, p_0^+; t_0^+) \\ &:= \left(p_0 + \text{sgn}(p_0) \frac{W}{\rho - 1}, p_0 + \text{sgn}(p_0) \frac{W}{\rho - 1}; 2|p_1| + |p_0| + \frac{W}{\rho - 1} \right) \end{aligned}$$

and distinguish two cases.

Case 1: At time t_0^+ , ALG is at least as close to p_1 as to p_0^+ or it serves σ_1 before σ_0^+ . See Fig. 2 for an illustration of this case. In this case, we do not present additional requests. By Lemma 6, ALG has served σ_0 at time t_0^+ or before and by Observation 2 it does not serve σ_1 earlier than time $(2\rho - 1)|p_1| + (\rho - 1)|p_0| + W$. Thus, we have

$$\begin{aligned} \text{ALG}(\sigma_\rho) &\geq (2\rho - 1)|p_1| + (\rho - 1)|p_0| + W + |p_1| + |p_0| + \frac{W}{\rho - 1} \\ &\geq \rho \left(2|p_1| + |p_0| + \frac{W}{\rho - 1} \right) \\ &= \rho \text{OPT}(\sigma_\rho). \end{aligned}$$

Case 2: At time t_0^+ , ALG is closer to p_0^+ than to p_1 and it serves σ_0^+ first. We assume that the offline server continues moving away from the origin after serving σ_0^+ at time $|p_0^+|$. Then, the position of the offline server at time $t \geq |p_1|$ is $\text{sgn}(p_0)t + 2p_1$. We denote by

$$M(t) := \frac{\text{sgn}(p_0)t + 3p_1}{2}$$

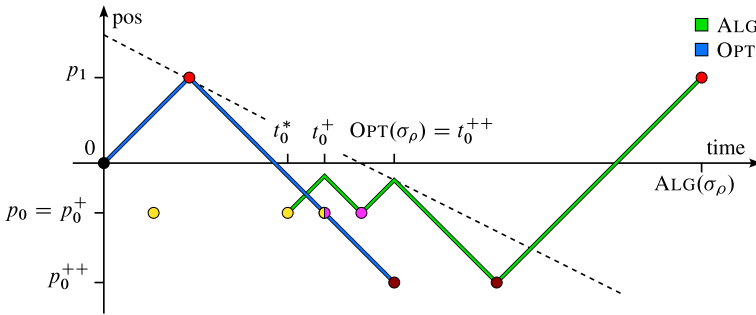


Fig. 3 Case 2.1: The midpoint of OPT’s position and p_1 (dashed line) reaches p_0^+ (violet) the same time as ALG (green). No new requests are released. In this figure, we have $\text{delay}(t) = W = 2$ for $t \geq t_0^* + W$. OPT is blue, the request σ_0 is yellow, and the request σ_1 is red (Color figure online)

the midpoint between the current position of the offline server and the position p_1 . Note that the time $M^{-1}(p)$, when the midpoint is at position p (on the same side of p_1 as p_0), is given by

$$M^{-1}(p) := |2p - 3p_1|.$$

We again distinguish two cases.

Case 2.1: ALG does not serve σ_0^+ until time $M^{-1}(p_0^+)$. See Fig. 3 for an illustration of this case. In this case, we do not present additional requests. Since we are in Case 2, neither σ_0^+ nor σ_1 is served at time $M^{-1}(p_0^+)$. Thus, we have

$$\begin{aligned} \text{ALG}(\sigma_\rho) &\geq M^{-1}(p_0^+) + |p_0^+| + |p_1| \\ &= |2p_0^+ - 3p_1| + |p_0^+| + |p_1| \\ &= |2p_0 + 2\text{sgn}(p_0)\frac{W}{\rho - 1} - 3p_1| + |p_0| + \frac{W}{\rho - 1} + |p_1| \\ &= 3|p_0| + 4|p_1| + 3\frac{W}{\rho - 1} \\ &= \rho|p_0| + (3 - \rho)|p_0| + 4|p_1| + 3\frac{W}{\rho - 1} \\ &\stackrel{\text{Def. 1 (v)}}{\geq} \rho|p_0| + \frac{(3 - \rho)(-8\rho^2 + 50\rho - 66)}{4\rho^2 - 30\rho + 50}|p_1| + 4|p_1| + 3\frac{W}{\rho - 1} \\ &= \rho|p_0| + \frac{4\rho^3 - 29\rho^2 + 48\rho + 1}{2\rho^2 - 15\rho + 25}|p_1| + 3\frac{W}{\rho - 1} \\ &\stackrel{2 < \rho < 2.5}{>}{\rho|p_0| + 2\rho|p_1| + 3\frac{W}{\rho - 1}} \\ &\stackrel{\rho < 3}{>}{\rho \left(|p_0| + 2|p_1| + \frac{W}{\rho - 1} \right)} \\ &= \rho \text{OPT}(\sigma_\rho). \end{aligned}$$

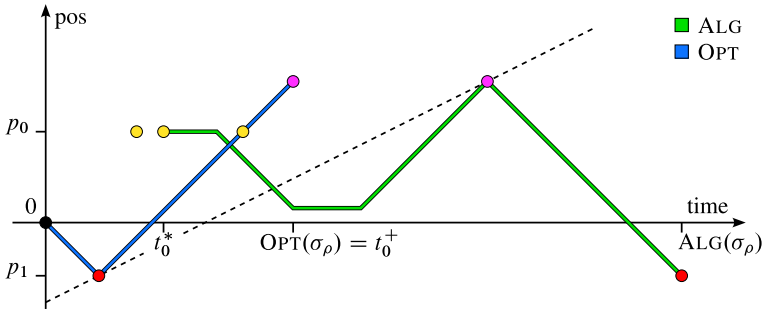


Fig. 4 Case 2.2: ALG (green) serves σ_0^+ (violet) before the midpoint of OPT’s position and p_1 (dashed line) reaches σ_0^+ . Thus, σ_0^+ (brown) is released. We have $\text{delay}(t) = W = 0$ for $t \geq t^* + W$. OPT is shown in blue, σ_0 is shown in yellow, and σ_1 is shown in red (Color figure online)

Case 2.2: ALG serves σ_0^+ before time $M^{-1}(p_0^+)$.

See Fig. 4 for an illustration of this case. By definition of W , the delay function is defined for time t_0^+ , hence ALG has not served σ_1 before time t_0^+ . Note that, at time t_0^+ , ALG is not on the same side of the midpoint $M(t_0^+)$ as this unserved request (using that we are in Case 2) and the midpoint is moving towards the server, so the server and midpoint must eventually meet. Additionally, since ALG serves σ_0^+ before time $M^{-1}(p_0^+)$ (using that we are in Case 2.2), i.e., the time that the midpoint reaches p_0^+ , there is a (first) time $t_{\text{mid}} \geq t_0^+$ at which ALG has served σ_0^+ and $M(t_{\text{mid}}) = \text{pos}(t_{\text{mid}})$. We present the request

$$\sigma_0^{++} = (p_0^{++}, p_0^{++}; t_0^{++}) := (\text{sgn}(p_0)t_{\text{mid}} + 2p_1, \text{sgn}(p_0)t_{\text{mid}} + 2p_1; t_{\text{mid}}).$$

Note that ALG is at the midpoint between p_0^{++} and p_1 and, thus, both tours $\text{move}(p_0^{++}) \oplus \text{move}(p_1)$ and $\text{move}(p_1) \oplus \text{move}(p_0^{++})$ incur identical costs for ALG. We have

$$\text{ALG}(\sigma_\rho) \geq t_{\text{mid}} + 3 \left(\frac{|\text{sgn}(p_0)t_{\text{mid}} + 2p_1 - p_1|}{2} \right) = \frac{5t_{\text{mid}} - 3|p_1|}{2}$$

We have $\text{OPT}(\sigma_\rho) = t_{\text{mid}}$, i.e., we want to show

$$\text{ALG}(\sigma_\rho) \geq \frac{5t_{\text{mid}} - 3|p_1|}{2} \geq \rho t_{\text{mid}} = \rho \text{OPT}(\sigma_\rho). \tag{11}$$

Inequality (11) is equivalent to

$$(5 - 2\rho)t_{\text{mid}} \geq 3|p_1|. \tag{12}$$

Since $\rho < 2.5$, the coefficient $(5 - 2\rho)$ of t_{mid} is positive. Thus we may assume t_{mid} is minimal to show the inequality (12). By definition of t_{mid} , σ_0^+ is already served at time t_{mid} . Hence, t_{mid} is minimum if, starting at time t_0^+ at position $\text{pos}(t_0^+)$, ALG serves σ_0^+ and then moves towards the origin. Then, t_{mid} is the solution of the equation

$$\text{sgn}(p_0)t_0^+ + |\text{pos}(t_0^+) - p_0^+| + p_0^+ - \text{sgn}(p_0)t_{\text{mid}} = \frac{\text{sgn}(p_0)t_{\text{mid}} + 3p_1}{2}. \quad (13)$$

Because of Lemma 6, the request σ_0 is already served at time t_0^+ . Furthermore, since the position of σ_1 has not been visited yet at time t_0^+ , we have $\text{sgn}(p_0)\text{pos}(t_0^+) > \text{sgn}(p_0)p_1$, i.e.,

$$|\text{pos}(t_0^+) - p_1| = \text{sgn}(p_0)(\text{pos}(t_0^+) - p_1) > 0$$

and thus, because of $-\text{sgn}(p_0)p_1 = |p_1|$, we get

$$\begin{aligned} \text{delay}(t_0^+) &= t_0^+ + |\text{pos}(t_0^+) - p_1| - (\rho - 1)|p_0| - (2\rho - 1)|p_1| \\ &= t_0^+ + \text{sgn}(p_0)\text{pos}(t_0^+) - \text{sgn}(p_0)p_1 - (\rho - 1)|p_0| - (2\rho - 1)|p_1| \\ &= t_0^+ + \text{sgn}(p_0)\text{pos}(t_0^+) + |p_1| - (\rho - 1)|p_0| - (2\rho - 1)|p_1|. \end{aligned} \quad (14)$$

Solving equation (14) for $\text{sgn}(p_0)\text{pos}(t_0^+)$ gives

$$\begin{aligned} &\text{sgn}(p_0)\text{pos}(t_0^+) \\ &= \text{delay}\left(2|p_1| + |p_0| + \frac{W}{\rho - 1}\right) - \frac{W}{\rho - 1} \\ &\quad + (\rho - 2)|p_0| + (2\rho - 4)|p_1| \\ &= W - \frac{W}{\rho - 1} + (\rho - 2)|p_0| + (2\rho - 4)|p_1| \\ &= \frac{\rho - 2}{\rho - 1}W + (\rho - 2)|p_0| + (2\rho - 4)|p_1| \\ &\stackrel{\rho < 3}{<} \frac{W}{\rho - 1} + (\rho - 2)|p_0| + (2\rho - 4)|p_1| \\ &\stackrel{\text{Def 1 (v)}}{\leq} \frac{W}{\rho - 1} + \left((\rho - 2) + (2\rho - 4)\frac{4\rho^2 - 30\rho + 50}{-8\rho^2 + 50\rho - 66}\right)|p_0| \\ &\stackrel{1.9 < \rho < 4.3}{<} \frac{W}{\rho - 1} + |p_0| \\ &= |p_0^+| \\ \text{sgn}(p_0^+) &= \text{sgn}(p_0) \\ &= \text{sgn}(p_0)p_0^+. \end{aligned} \quad (15)$$

Thus, we have

$$|\text{pos}(t_0^+) - p_0^+| = \text{sgn}(p_0)(p_0^+ - \text{pos}(t_0^+)) > 0 \quad (16)$$

Using inequality (16) and plugging inequality (15) into inequality (13) gives us

$$\begin{aligned}
 \operatorname{sgn}(p_0)t_{\text{mid}} &= \frac{1}{3}(2\operatorname{sgn}(p_0)t_0^+ + 2|\operatorname{pos}(t_0^+) - 2p_0^+| + 2p_0^+ - 3p_1) \\
 &\stackrel{(16)}{=} \frac{1}{3}(2\operatorname{sgn}(p_0)t_0^+ + 2\operatorname{sgn}(p_0)p_0^+ - 2\operatorname{sgn}(p_0)\operatorname{pos}(t_0^+) + 2p_0^+ - 3p_1) \\
 &= \frac{1}{3}\left(-7p_1 + 6p_0 + \frac{(6\operatorname{sgn}(p_0))W}{\rho - 1} - 2\operatorname{sgn}(p_0)\operatorname{pos}(t_0^+)\right) \\
 &\stackrel{(15)}{=} \frac{1}{3}\left(- (15 - 4\rho)p_1 + (10 - 2\rho)p_0 + \frac{(10 - 2\rho)\operatorname{sgn}(p_0)W}{\rho - 1}\right) \quad (17)
 \end{aligned}$$

Note that we also used $\operatorname{sgn}(p_0) = \operatorname{sgn}(p_0^+) = -\operatorname{sgn}(p_1)$. Multiplying equality (17) with $\operatorname{sgn}(p_0)$ gives us

$$t_{\text{mid}} = \frac{1}{3}\left((15 - 4\rho)|p_1| + (10 - 2\rho)|p_0| + \frac{(10 - 2\rho)W}{\rho - 1}\right). \quad (18)$$

By substituting (18) into (12) and noting that it is hardest to satisfy, when $W = 0$, we get

$$\frac{|p_0|}{|p_1|} \leq \frac{4\rho^2 - 30\rho + 50}{-8\rho^2 + 50\rho - 66},$$

which is true due to Definition 1 (v). □

This completes the proof of Theorem 1.

3 An Improved Algorithm

One of the simplest approaches for an online algorithm to solve DIAL-A-RIDE is the following: Always serve the set of currently unserved requests in an optimum offline schedule and ignore all new incoming requests while doing so. Afterwards, repeat this procedure with all ignored unserved requests until no new requests arrive. This simple algorithm that is often called IGNORE [1] has a competitive ratio of exactly 4 [7, 18]. The main weakness of IGNORE is that it always starts its schedule immediately. Ascheuer et al. showed that it is beneficial if the server waits sometimes before starting a schedule and introduced the SMARTSTART algorithm [1], which has a competitive ratio of roughly 2.94 [7].

We define $L(t, p, R)$ to be the smallest length of a 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). In other words, $L(t, p, R)$ is the earliest possible completion time of a schedule for R that is only allowed to start at time t minus t . 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.

Algorithm 1: SMARTSTART

```

 $p_1 \leftarrow 0$ 
for  $j = 1, 2, \dots$  do
  while current time  $t < L(t, p_j, R_t)/(\Theta - 1)$  do
    | wait
   $t_j \leftarrow t$ 
   $S_j \leftarrow$  optimal offline schedule serving  $R_t$  starting from  $p_j$ 
  execute  $S_j$ 
   $p_{j+1} \leftarrow$  current position

```

The algorithm SMARTSTART is given in Algorithm 1. Essentially, at time t , SMARTSTART waits before starting an optimal schedule to serve all available requests at time

$$\min_{t' \geq t} \left\{ t' \geq \frac{L(t', p, R_{t'})}{\Theta - 1} \right\}, \tag{19}$$

where p is the current position of the server and $\Theta > 1$ is a parameter of the algorithm that scales the waiting time. Importantly, like IGNORE, SMARTSTART ignores incoming requests while executing a schedule.

Birx and Disser identified that SMARTSTART’s waiting routine defined by inequality (19) has a critical weakness [7, Lemma 4.1]. It is possible to lure the server to any position q in time $q + \varepsilon$ for every $\varepsilon > 0$. Roughly speaking, a request $\sigma_1 = ((\Theta - 1)\varepsilon, (\Theta - 1)\varepsilon; (\Theta - 1)\varepsilon)$ is released first and then for every $i \in \{2, \dots, q/\varepsilon\}$ a request $\sigma_i = (i\varepsilon, i\varepsilon; i\varepsilon)$ follows. The schedule to serve the request σ_1 is started at time ε and finished at time 2ε . The schedule to serve the request at position $i\varepsilon$ is not started earlier than time

$$\frac{L(i\varepsilon, (i - 1)\varepsilon, \{\sigma_i\})}{\Theta - 1} = \frac{|(i - 1)\varepsilon - i\varepsilon|}{\Theta - 1} = \frac{\varepsilon}{\Theta - 1}. \tag{20}$$

This time is (depending on the choice of Θ) earlier than the current time $i\varepsilon$ for every $i \geq 2$. Thus there is no waiting time for any schedule except the first one and the server reaches position q at time $q + \varepsilon$. We see that the request sequence to lure the server away heavily uses that inequality (19) relies on SMARTSTART’s current position p , when computing the waiting time. Thus, we modify the waiting routine of SMARTSTART to avoid luring accordingly. Denote by $\sigma_{\leq t}$ the set of requests that have been released until time t .

The improved algorithm SMARTERSTART is given in Algorithm 2. At time t , it waits before starting an optimal schedule to serve all available requests at time

$$\min_{t' \geq t} \left\{ t' \geq \frac{L(t', 0, \sigma_{\leq t'})}{\Theta - 1} \right\}. \tag{21}$$

Again, $\Theta > 1$ is a parameter of the algorithm that scales the waiting time. In contrast to SMARTSTART, the waiting time is dependent on the length of the optimum offline

Algorithm 2: SMARTERSTART

```

 $p_1 \leftarrow 0$ 
for  $j = 1, 2, \dots$  do
  while current time  $t < L(t, 0, \sigma_{\leq t})/(\Theta - 1)$  do
    | wait
   $t_j \leftarrow t$ 
   $S_j \leftarrow$  optimal offline schedule serving  $R_t$  starting from  $p_j$ 
  execute  $S_j$ 
   $p_{j+1} \leftarrow$  current position

```

schedule serving all requests that appeared until the current time and starting from the origin. This guarantees that the server cannot be forced to reach any position q before time $q/(\Theta - 1)$ since we always have $L(t, 0, \sigma_{\leq t}) > q$ if $\sigma_{\leq t}$ contains a request with destination in position q .

We denote by $\text{SMARTERSTART}(\sigma)$ the length of SMARTERSTART’s trajectory. Whenever we need to distinguish the behavior of SMARTERSTART for different values of $\Theta > 1$, we write $\text{SMARTERSTART}_\Theta$ to make the choice of Θ explicit. Note that the schedules used by IGNORE, SMARTSTART and SMARTERSTART are NP-hard to compute for $1 < c < \infty$, see [9].

We let $N \in \mathbb{N}$ be the number of schedules needed by SMARTERSTART 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} , and the set of requests served in S_j by σ_{S_j} . For convenience, we set $t_0 = p_0 = 0$.

3.1 Upper Bound for SMARTERSTART

We show the upper bound of Theorem 2. The completion time of SMARTERSTART is

$$\text{SMARTERSTART}(\sigma) = t_N + L(t_N, p_N, \sigma_{S_N}). \tag{22}$$

First, observe that, for all $0 \leq t \leq t'$, $p, p' \in \mathbb{R}$, and $R \subseteq \sigma$, we have

$$L(t, p, R) \geq L(t', p, R), \tag{23}$$

$$L(t, p, R) \leq |p - p'| + L(t, p', R), \tag{24}$$

$$L(t, 0, \sigma_{\leq t}) \leq L(t, 0, \sigma) \leq L(0, 0, \sigma) \leq \text{OPT}(\sigma). \tag{25}$$

Similarly to [7], we distinguish between two cases, depending on whether or not SMARTERSTART postpones the execution of the final schedule S_N .

If SMARTERSTART postpones the execution of S_N (i.e., it waits even though there are unserved requests), the starting time of schedule S_N is given by

$$t_N = \frac{1}{\Theta - 1} L(t_N, 0, \sigma_{\leq t_N}). \tag{26}$$

Otherwise, we have

$$t_N = t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{S_{N-1}}) \tag{27}$$

or

$$t_N = r_n. \tag{28}$$

The former (27) is the case if the final schedule S_N is executed directly after the second to final schedule S_{N-1} . The latter (28) is the case if there are no unserved requests at the point of time the execution of S_{N-1} is finished and the last request is released at time $r_n > \frac{1}{\Theta-1}L(t_N, 0, \sigma_{\leq t_N})$. We start by giving a lower bound on the starting time of a schedule. It was shown in [7] that the schedule S_j of SMARTSTART is never started earlier than time $\frac{|p_{j+1}|}{\Theta}$. This changes slightly for SMARTERSTART.

Lemma 7 *Algorithm SMARTERSTART does not start schedule S_j earlier than time $\frac{|p_{j+1}|}{\Theta-1}$, i.e., we have $t_j \geq \frac{|p_{j+1}|}{\Theta-1}$.*

Proof Since p_{j+1} is the ending point of schedule S_j , there is a request with destination in p_{j+1} in the set σ_{S_j} . All requests of σ_{S_j} appear before time t_j , which implies that they are part of the set $\sigma_{\leq t_j}$. Thus, we have

$$L(t_j, 0, \sigma_{\leq t_j}) \geq |p_{j+1}| \tag{29}$$

and therefore

$$t_j \stackrel{(21)}{\geq} \frac{L(t_j, 0, \sigma_{\leq t_j})}{\Theta - 1} \stackrel{(29)}{\geq} \frac{|p_{j+1}|}{\Theta - 1}$$

This completes the proof. □

Using Lemma 7, we can give an upper bound on the length of SMARTERSTART’s schedules, which is an essential ingredient in our upper bounds. The following lemma is proved similarly to [7, Lemma 3.2], which yields an upper bound of $(1 + \frac{\Theta}{\Theta+2})\text{OPT}(\sigma)$ for the length of every schedule S_j of SMARTSTART.

Lemma 8 *For every schedule S_j of SMARTERSTART, we have*

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

Proof First, we notice that, by (24), we have

$$L(t_j, p_j, \sigma_{S_j}) \leq |p_j| + L(t_j, 0, \sigma_{S_j}) \leq \text{OPT}(\sigma) + |p_j|. \tag{30}$$

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

$$L(t_j, p_j, \sigma_{S_j}) \leq |a_j^{\text{OPT}} - p_j| + L(t_j, a_j^{\text{OPT}}, \sigma_{S_j}), \tag{31}$$

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

$$L(t_j, a_j^{\text{OPT}}, \sigma_{S_j}) \stackrel{r_j^{\text{OPT}} \leq t_j}{\leq} L(r_j^{\text{OPT}}, a_j^{\text{OPT}}, \sigma_{S_j}) \leq \text{OPT}(\sigma) - r_j^{\text{OPT}}, \tag{32}$$

which yields

$$\begin{aligned} L(t_j, p_j, \sigma_{S_j}) &\stackrel{(31)}{\leq} |a_j^{\text{OPT}} - p_j| + L(t_j, a_j^{\text{OPT}}, \sigma_{S_j}) \\ &\stackrel{(32)}{\leq} \text{OPT}(\sigma) + |a_j^{\text{OPT}} - p_j| - r_j^{\text{OPT}} \\ &\stackrel{t_{j-1} < r_j^{\text{OPT}}}{<} \text{OPT}(\sigma) + |a_j^{\text{OPT}} - p_j| - t_{j-1}. \end{aligned} \tag{33}$$

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

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

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

$$t_{j-1} + |a_j^{\text{OPT}} - p_j| \stackrel{t_{j-1} < r_j^{\text{OPT}}}{<} r_j^{\text{OPT}} + |a_j^{\text{OPT}} - p_j| \leq \text{OPT}(\sigma), \tag{34}$$

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

$$\begin{aligned} L(t_j, p_j, \sigma_{S_j}) &\stackrel{(33)}{<} \text{OPT}(\sigma) + |a_j^{\text{OPT}} - p_j| - t_{j-1} \\ &\stackrel{(34)}{\leq} 2\text{OPT}(\sigma) - 2t_{j-1} \\ &\stackrel{\text{Lem. 7}}{\leq} 2\text{OPT}(\sigma) - 2\frac{|p_j|}{\Theta - 1}. \end{aligned} \tag{35}$$

This implies

$$\begin{aligned} L(t_j, p_j, \sigma_{S_j}) &\stackrel{(30),(35)}{\leq} \min \left\{ \text{OPT}(\sigma) + |p_j|, 2\text{OPT}(\sigma) - \frac{2}{\Theta - 1}|p_j| \right\} \\ &\leq \left(1 + \frac{\Theta - 1}{\Theta + 1}\right) \text{OPT}(\sigma), \end{aligned}$$

since the minimum above is largest if the two terms are equal, which is the case for $|p_j| = \frac{\Theta-1}{\Theta+1} \text{OPT}(\sigma)$. \square

The following proposition uses Lemma 8 to provide an upper bound for the competitive ratio of SMARTERSTART, in the case that SMARTERSTART does have a waiting period before starting the final schedule.

Proposition 1 *In case SMARTERSTART postpones executing S_N , we have*

$$\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} \leq f_1(\Theta) := \frac{2\Theta^2 - \Theta + 1}{\Theta^2 - 1}.$$

Proof Assume SMARTERSTART waits before starting the final schedule. Then Lemma 8 yields the claimed bound:

$$\begin{aligned} \text{SMARTERSTART}(\sigma) &\stackrel{(22)}{=} t_N + L(t_N, p_N, \sigma_{S_N}) \\ &\stackrel{(26)}{=} \frac{1}{\Theta - 1} L(t_N, 0, \sigma_{\leq t_N}) + L(t_N, p_N, \sigma_{S_N}) \\ &\stackrel{(25)}{\leq} \frac{1}{\Theta - 1} \text{OPT}(\sigma) + L(t_N, p_N, \sigma_{S_N}) \\ &\stackrel{\text{Lem. 8}}{\leq} \left(\frac{1}{\Theta - 1} + 1 + \frac{\Theta - 1}{\Theta + 1} \right) \text{OPT}(\sigma) \\ &= \frac{2\Theta^2 - \Theta + 1}{\Theta^2 - 1} \text{OPT}(\sigma). \end{aligned}$$

This completes the proof. \square

In comparison, the upper bound for the competitive ratio of SMARTSTART, in case SMARTSTART has a waiting period before starting the final schedule is $\frac{2\Theta^2+2\Theta}{\Theta^2+\Theta-2} \text{OPT}(\sigma)$ [7, Proposition 3.2]. Note that SMARTERSTART’s bound is better than SMARTSTART’s bound for $\Theta > 1$.

It remains to examine the case that the algorithm SMARTERSTART has no waiting period before starting the final schedule. For this we use two lemmas from [7] originally proved for SMARTSTART, which are still valid for SMARTERSTART since they give bounds on the optimum offline schedules independently of the waiting routine.

By $x_- := \min\{0, \min_{i=1, \dots, n} a_i, \min_{i=1, \dots, n} b_i\}$ we denote the leftmost position 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. 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_-/x_+ .

We mention that SMARTERSTART is a so-called *schedule-based* algorithm, meaning that it alternates between waiting for some time and executing (without interruption) optimal schedules on all currently unserved requests (see [6] for a formal definition). In particular, the following lemmas can be applied (see also Lemmas 3.4 and 3.6 in [7]). The proofs of these lemmas do not depend on the specific waiting condition and

only need that the schedules S_j are optimal schedules for σ_{S_j} and that all requests of σ_{S_j} have already appeared by the starting time t_j of S_j .

Lemma 9 (Lemma 4.7 in [6]) *Let S_j with $j \in \{1, \dots, N\}$ be a schedule of SMARTERSTART. Moreover, let $\text{OPT}(\sigma) = |x_-| + x_+ + y$ for some $y \geq 0$. Then, we have*

$$L(t_j, 0, \sigma_{S_j}) \leq |\min\{0, y_-^{S_j}\}| + \max\{0, y_+^{S_j}\} + y.$$

Lemma 10 (Lemma 4.9 in [6]) *Let S_j with $j \in \{1, \dots, N\}$ be a schedule of SMARTERSTART. Moreover, let $|x_-| \leq x_+$ and $\text{OPT}(\sigma) = |x_-| + x_+ + y$ for some $y \geq 0$. Then, for every point p that is visited by S_j we have*

$$p \leq |p_j| + |p_j - p_{j+1}| + y - |\min\{0, y_-^{S_j}\}|.$$

Using the bounds established by Lemma 9 and Lemma 10, we can give an upper bound for the competitive ratio of SMARTERSTART if the server is not waiting before starting the final schedule.

Proposition 2 *If SMARTERSTART does not postpone executing S_N , we have*

$$\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} \leq f_2(\Theta) := \frac{3\Theta^2 + 3}{2\Theta + 1}.$$

Proof Assume algorithm SMARTERSTART does not postpone the last schedule, i.e., SMARTERSTART starts the final schedule S_N either immediately after finishing S_{N-1} or immediately after the last request is released.

Let the latter be the case. Then, the final schedule is started at the release time r_n of the last request. Since OPT also has to serve the last request, we have

$$\text{OPT}(\sigma) \geq r_n. \tag{36}$$

In total we have

$$\begin{aligned} \text{SMARTERSTART}(\sigma) &\stackrel{(22)}{=} t_N + L(t_N, p_N, \sigma_{S_N}) \\ &\stackrel{(28)}{=} r_n + L(t_N, p_N, \sigma_{S_N}) \\ &\stackrel{(36)}{\leq} \text{OPT}(\sigma) + L(t_N, p_N, \sigma_{S_N}) \\ &\stackrel{\text{Lem. 8}}{\leq} \left(2 + \frac{\Theta - 1}{\Theta + 1}\right) \text{OPT}(\sigma) \\ &\stackrel{\Theta > 1}{<} \frac{3\Theta^2 + 3}{2\Theta + 1} \text{OPT}(\sigma). \end{aligned}$$

Now, consider the case that the last schedule is started immediately after the second to final one. Let $\sigma_{S_N}^{\text{OPT}}$ be the first request of σ_{S_N} that is served by OPT and let a_N^{OPT} be

its starting point and r_N^{OPT} be its release time. We have

$$\begin{aligned}
 \text{SMARTERSTART}(\sigma) &\stackrel{(22)}{=} t_N + L(t_N, p_N, \sigma_{S_N}) \\
 &\stackrel{(27)}{=} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{S_{N-1}}) + L(t_N, p_N, \sigma_{S_N}) \\
 &\stackrel{t_N \geq r_N^{\text{OPT}}}{\leq} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{S_{N-1}}) \\
 &\quad + L(r_N^{\text{OPT}}, p_N, \sigma_{S_N}). \tag{37}
 \end{aligned}$$

Since OPT serves all requests of σ_{S_N} after time r_N^{OPT} , starting with a request with starting point a_N^{OPT} , we also have

$$\text{OPT}(\sigma) \geq r_N^{\text{OPT}} + L(r_N^{\text{OPT}}, a_N^{\text{OPT}}, \sigma_{S_N}). \tag{38}$$

Furthermore, we have

$$r_N^{\text{OPT}} > t_{N-1} \tag{39}$$

since otherwise $\sigma_{S_N}^{\text{OPT}} \in \sigma_{S_{N-1}}$ would hold. This gives us

$$\begin{aligned}
 \text{SMARTERSTART}(\sigma) &\stackrel{(37)}{\leq} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{S_{N-1}}) + L(r_N^{\text{OPT}}, p_N, \sigma_{S_N}) \\
 &\stackrel{(24)}{\leq} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{S_{N-1}}) + |a_N^{\text{OPT}} - p_N| \\
 &\quad + L(r_N^{\text{OPT}}, a_N^{\text{OPT}}, \sigma_{S_N}) \\
 &\stackrel{(38)}{\leq} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{S_{N-1}}) + |a_N^{\text{OPT}} - p_N| \\
 &\quad + \text{OPT}(\sigma) - r_N^{\text{OPT}} \\
 &\stackrel{(39)}{<} L(t_{N-1}, p_{N-1}, \sigma_{S_{N-1}}) + |a_N^{\text{OPT}} - p_N| + \text{OPT}(\sigma) \tag{40} \\
 &\stackrel{(24)}{\leq} |p_{N-1}| + L(t_{N-1}, 0, \sigma_{S_{N-1}}) \\
 &\quad + |a_N^{\text{OPT}} - p_N| + \text{OPT}(\sigma) \\
 &\stackrel{\text{Lem. 7}}{\leq} (\Theta - 1)t_{N-2} + L(t_{N-1}, 0, \sigma_{S_{N-1}}) \\
 &\quad + |a_N^{\text{OPT}} - p_N| + \text{OPT}(\sigma). \tag{41}
 \end{aligned}$$

We have

$$\text{OPT}(\sigma) \geq t_{N-2} + |a_N^{\text{OPT}} - p_N|, \tag{42}$$

because OPT has to visit both a_N^{OPT} and p_N after time t_{N-2} : It has to visit a_N^{OPT} to collect $\sigma_{S_N}^{\text{OPT}}$ and it has to visit p_N to deliver some request of $\sigma_{S_{N-1}}$. Using the above

inequality, we get

$$\begin{aligned} \text{SMARTERSTART}(\sigma) &\stackrel{(41)}{<} (\Theta - 1)t_{N-2} + L(t_{N-1}, 0, \sigma_{S_{N-1}}) \\ &\quad + |a_N^{\text{OPT}} - p_N| + \text{OPT}(\sigma) \\ &\stackrel{(42)}{\leq} 2\text{OPT}(\sigma) + L(t_{N-1}, 0, \sigma_{S_{N-1}}) + (\Theta - 2)t_{N-2}. \end{aligned} \tag{43}$$

In the case $\Theta \geq 2$, we have

$$\begin{aligned} \text{SMARTERSTART}(\sigma) &\stackrel{(43)}{<} 2\text{OPT}(\sigma) + L(t_{N-1}, 0, \sigma_{S_{N-1}}) + (\Theta - 2)t_{N-2} \\ &\stackrel{(25)}{\leq} (\Theta + 1)\text{OPT}(\sigma) \\ &\stackrel{\Theta \geq 2}{\leq} \frac{3\Theta^2 + 3}{2\Theta + 1}\text{OPT}(\sigma). \end{aligned}$$

Thus, we may assume $\Theta < 2$. Similarly as in inequality (43), we get

$$\begin{aligned} \text{SMARTERSTART}(\sigma) &\stackrel{(41)}{<} (\Theta - 1)t_{N-2} + L(t_{N-1}, 0, \sigma_{S_{N-1}}) \\ &\quad + |a_N^{\text{OPT}} - p_N| + \text{OPT}(\sigma) \\ &\stackrel{(42)}{\leq} \Theta\text{OPT}(\sigma) + L(t_{N-1}, 0, \sigma_{S_{N-1}}) + (2 - \Theta)|a_N^{\text{OPT}} - p_N| \\ &\stackrel{(21)}{\leq} \Theta\text{OPT}(\sigma) + (\Theta - 1)t_{N-1} + (2 - \Theta)|a_N^{\text{OPT}} - p_N| \\ &\leq (2\Theta - 1)\text{OPT}(\sigma) + (2 - \Theta)|a_N^{\text{OPT}} - p_N|, \end{aligned} \tag{44}$$

where the last inequality follows, because there exists a request in σ with release date later than t_{N-1} . This means the claim is shown if we have

$$|p_N - a_N^{\text{OPT}}| \leq \text{OPT}(\sigma) - \frac{\Theta - 1}{2\Theta + 1}\text{OPT}(\sigma) \tag{45}$$

since then we have

$$\begin{aligned} \text{SMARTERSTART}(\sigma) &\stackrel{(44)}{<} (2\Theta - 1)\text{OPT}(\sigma) + (2 - \Theta)|a_N^{\text{OPT}} - p_N| \\ &\stackrel{(45)}{\leq} (2\Theta - 1)\text{OPT}(\sigma) + (2 - \Theta)\left(1 - \frac{\Theta - 1}{2\Theta + 1}\right)\text{OPT}(\sigma) \\ &= \frac{3\Theta^2 + 3}{2\Theta + 1}\text{OPT}(\sigma). \end{aligned}$$

Therefore, we may assume in the following that

$$|p_N - a_N^{\text{OPT}}| > \text{OPT}(\sigma) - \frac{\Theta - 1}{2\Theta + 1}\text{OPT}(\sigma). \tag{46}$$

Let $\text{OPT}(\sigma) = |x_-| + x_+ + y$ for some $y \geq 0$. By definition of x_- and x_+ we have

$$|p_N - a_N^{\text{OPT}}| + y \leq \text{OPT}(\sigma). \tag{47}$$

In the case that OPT visits position p_N before it collects $\sigma_{S_N}^{\text{OPT}}$, we have

$$|a_N^{\text{OPT}} - p_N| + |p_N| \leq \text{OPT}(\sigma). \tag{48}$$

Similarly, if OPT collects $\sigma_{S_N}^{\text{OPT}}$ before it visits position p_N for the first time, we have

$$\begin{aligned} \text{OPT}(\sigma) &\geq r_N^{\text{OPT}} + |a_N^{\text{OPT}} - p_N| \\ &\stackrel{(39)}{>} t_{N-1} + |a_N^{\text{OPT}} - p_N| \\ &\stackrel{\text{Lem. 7}}{\geq} \frac{|p_N|}{\Theta - 1} + |a_N^{\text{OPT}} - p_N| \\ &\stackrel{\Theta \leq 2}{\geq} |p_N| + |a_N^{\text{OPT}} - p_N|. \end{aligned}$$

Thus, inequality (48) holds in general. To sum it up, we may assume that

$$\max\{y, |p_N|, t_{N-2}\} \stackrel{(46),(47),(48),(42)}{<} \frac{\Theta - 1}{2\Theta + 1} \text{OPT}(\sigma) \tag{49}$$

holds. In the following, denote by $y_-^{S_{N-1}}$ the leftmost starting or ending point and by $y_+^{S_{N-1}}$ the rightmost starting or ending point of the requests in $\sigma_{S_{N-1}}$. We compute

$$\begin{aligned} \text{SMARTERSTART}(\sigma) &\stackrel{(40)}{<} L(t_{N-1}, p_{N-1}, \sigma_{S_{N-1}}) + |p_N - a_N^{\text{OPT}}| + \text{OPT}(\sigma) \\ &\stackrel{(48)}{<} L(t_{N-1}, p_{N-1}, \sigma_{S_{N-1}}) + 2\text{OPT}(\sigma) - |p_N| \\ &\stackrel{(23)}{\leq} |p_{N-1}| + L(t_{N-1}, 0, \sigma_{S_{N-1}}) + 2\text{OPT}(\sigma) - |p_N| \\ &\stackrel{\text{Lem. 7}}{\leq} (\Theta - 1)t_{N-2} + L(t_{N-1}, 0, \sigma_{S_{N-1}}) \\ &\quad + 2\text{OPT}(\sigma) - |p_N| \\ &\stackrel{\text{Lem. 9}}{\leq} (\Theta - 1)t_{N-2} + \max\{0, |y_-^{S_{N-1}}|\} + \max\{0, y_+^{S_{N-1}}\} \\ &\quad + y + 2\text{OPT}(\sigma) - |p_N|. \end{aligned} \tag{50}$$

Obviously, position $y_+^{S_{N-1}}$ is visited by SMARTERSTART in schedule S_{N-1} . Therefore, $y_+^{S_{N-1}}$ is smaller than or equal to the rightmost point that is visited by SMARTERSTART during schedule S_{N-1} , which gives us

$$y_+^{S_{N-1}} \stackrel{\text{Lem. 10}}{\leq} |p_{N-1}| + |p_{N-1} - p_N| + y - \max\{0, |y_-^{S_{N-1}}|\}. \tag{51}$$

On the other hand, because of $|x_-| \leq x_+$, we have $\text{OPT}(\sigma) \geq 2|x_-| + x_+$, which implies $y \geq |x_-|$. By definition of x_- and $y_-^{S_{N-1}}$, we have $|x_-| \geq \max\{0, |y_-^{S_{N-1}}|\}$. This gives us $y \geq \max\{0, |y_-^{S_{N-1}}|\}$ and

$$0 \leq |p_{N-1}| + |p_{N-1} - p_N| + y - \max\{0, |y_-^{S_{N-1}}|\}. \tag{52}$$

To sum it up, we have

$$\max\{0, y_+^{S_{N-1}}\} \stackrel{(51),(52)}{\leq} |p_{N-1}| + |p_{N-1} - p_N| + y - \max\{0, |y_-^{S_{N-1}}|\}. \tag{53}$$

The inequality above gives us

$$\begin{aligned} \text{SMARTERSTART}(\sigma) &\stackrel{(50)}{<} (\Theta - 1)t_{N-2} + \max\{0, |y_-^{S_{N-1}}|\} + \max\{0, y_+^{S_{N-1}}\} \\ &\quad + y + 2\text{OPT}(\sigma) - |p_N| \\ &\stackrel{53}{\leq} (\Theta - 1)t_{N-2} + |p_{N-1}| + |p_{N-1} - p_N| + 2y \\ &\quad + 2\text{OPT}(\sigma) - |p_N| \\ &\leq (\Theta - 1)t_{N-2} + |p_{N-1}| + |p_{N-1}| + |p_N| + 2y \\ &\quad + 2\text{OPT}(\sigma) - |p_N| \\ &\stackrel{\text{Lem. 7}}{\leq} (\Theta - 1)t_{N-2} + 2(\Theta - 1)t_{N-2} + 2y + 2\text{OPT}(\sigma) \\ &\stackrel{(49)}{\leq} (3\Theta - 3) \frac{\Theta - 1}{2\Theta + 1} \text{OPT}(\sigma) + 2 \frac{\Theta - 1}{2\Theta + 1} \text{OPT}(\sigma) \\ &\quad + 2\text{OPT}(\sigma) \\ &= \frac{3\Theta^2 + 3}{2\Theta + 1} \text{OPT}(\sigma). \end{aligned}$$

This completes the proof. □

In comparison, the upper bound for the competitive ratio of SMARTSTART in case it does not have a waiting period before starting the final schedule is $\Theta + 1 - \frac{\Theta - 1}{3\Theta + 3} \text{OPT}(\sigma)$ [7, Proposition 3.4]. Note that SMARTERSTART’s bound is slightly worse than SMARTSTART’s bound for $\Theta > 1.47$. However, in combination with the bound of Proposition 1, SMARTERSTART has a better worst-case than SMARTSTART.

Theorem 3 *Let Θ^* be the largest solution of $f_1(\Theta) = f_2(\Theta)$, i.e.,*

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

Then, SMARTERSTART $_{\Theta^}$ is ρ^* -competitive with $\rho^* := f_1(\Theta^*) = f_2(\Theta^*) \approx 2.6662$.*

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

$$\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{2\Theta^2 - \Theta + 1}{\Theta^2 - 1} = f_1(\Theta)$$

in Proposition 1 and for the case, where SMARTERSTART starts the final schedule immediately after the second to final one, we have established the upper bound

$$\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{3\Theta^2 + 3}{2\Theta + 1} = f_2(\Theta)$$

in Proposition 2. Therefore, if it exists,

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

is the parameter for SMARTERSTART with the smallest upper bound. We note that f_1 is strictly decreasing for $\Theta > 1$ and that f_2 is strictly increasing for $\Theta > 1$. Therefore, if an intersection point of f_1 and f_2 that is larger than 1 exists, then this is at Θ^* . Indeed, the intersection point exists, which is the largest solution of

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

The resulting upper bound for the competitive ratio is

$$\rho^* = f_1(\Theta^*) = f_2(\Theta^*) \approx 2.6662.$$

This completes the proof. □

3.2 Lower Bound for SMARTERSTART

We show the lower bound of Theorem 2. 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). Further, we show that choices of $\Theta > 1$ different from Θ^* yield competitive ratios worse than $\rho^* \approx 2.67$. Together, this implies that ρ^* is exactly the best possible competitive ratio for SMARTERSTART.

Proposition 3 *Let $1 < \Theta < 2$. For every sufficiently small $\varepsilon > 0$, there is a set of requests σ such that SMARTERSTART waits before starting the final schedule and such that the inequality*

$$\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} \geq \frac{2\Theta^2 - \Theta + 1}{\Theta^2 - 1} - \varepsilon$$

holds, i.e., the upper bound established in Proposition 1 is tight for $\Theta \in (1, 2)$.

Proof Let $\varepsilon > 0$ with $\varepsilon < \frac{\Theta}{\Theta+1}$ and $\varepsilon' = \frac{\Theta+1}{2\Theta}\varepsilon$. Let the request

$$\sigma_1 = (1, 1; 0)$$

appear. For all $t \geq 0$ we have $L(t, 0, \{\sigma_1\}) = 1$. Thus, SMARTERSTART starts its first schedule S_1 at time $t_1 = \frac{1}{\Theta-1}$ and reaches position $p_2 = 1$ at time $\frac{\Theta}{\Theta-1}$. Next, we let the second and final request

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

appear. For $t \geq \frac{\Theta}{\Theta-1}$ we have

$$\begin{aligned} L(t, 0, \{\sigma_1, \sigma_2\}) &= \left|0 - \left(-\frac{1}{\Theta-1} + \varepsilon'\right)\right| + \left|\left(-\frac{1}{\Theta-1} + \varepsilon'\right) - 1\right| \\ &= \frac{2}{\Theta-1} - 2\varepsilon' + 1. \end{aligned}$$

Thus, the second and final schedule S_2 is not started before time

$$\frac{L\left(\frac{\Theta}{\Theta-1}, 0, \{\sigma_1, \sigma_2\}\right)}{\Theta-1} = \frac{2}{(\Theta-1)^2} + \frac{1-2\varepsilon'}{\Theta-1}.$$

By assumption, we have $\Theta < 2$ and $\varepsilon < \frac{\Theta}{\Theta+1}$, i.e., $\varepsilon' < \frac{1}{2}$, which implies that for the time $\frac{\Theta}{\Theta-1}$, when SMARTERSTART reaches position $p_2 = 1$, the inequality

$$\frac{L\left(\frac{\Theta}{\Theta-1}, 0, \{\sigma_1, \sigma_2\}\right)}{\Theta-1} = \frac{2}{(\Theta-1)^2} + \frac{1-2\varepsilon'}{\Theta-1} \stackrel{\varepsilon' < \frac{1}{2}}{>} \frac{2}{(\Theta-1)^2} \stackrel{1 < \Theta < 2}{>} \frac{\Theta}{\Theta-1} \quad (54)$$

holds. (Note that inequality (54) also holds for slightly larger Θ if we let $\varepsilon \rightarrow 0$.) Because of inequality (54), SMARTERSTART has a waiting period and starts the schedule S_2 at time

$$t_2 = \frac{L\left(\frac{\Theta}{\Theta-1}, 0, \{\sigma_1, \sigma_2\}\right)}{\Theta-1} = \frac{2}{(\Theta-1)^2} + \frac{1-2\varepsilon'}{\Theta-1}.$$

Serving σ_2 from position $p_2 = 1$ takes time

$$\begin{aligned} L(t_2, p_2, \{\sigma_2\}) &= \left|1 - \left(-\frac{1}{\Theta-1} + \varepsilon'\right)\right| + \left|\left(-\frac{1}{\Theta-1} + \varepsilon'\right) - 1\right| \\ &= 2 + \frac{2}{\Theta-1} - 2\varepsilon'. \end{aligned}$$

To sum it up, we have

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

On the other hand, OPT goes from the origin to $-\frac{1}{\Theta-1} + \varepsilon'$ to collect σ_2 at time $\frac{1}{\Theta-1} + \varepsilon'$ (i.e., it has to wait for $2\varepsilon'$ units of time after it reaches position $-\frac{1}{\Theta-1} + \varepsilon'$). Then OPT goes straight to position 1 delivering σ_2 and serving σ_1 . Therefore, we have

$$\text{OPT}(\sigma) = \left| 0 - \left(-\frac{1}{\Theta - 1} + \varepsilon' \right) \right| + 2\varepsilon' + \left| \left(-\frac{1}{\Theta - 1} + \varepsilon' \right) - 1 \right| = \frac{\Theta + 1}{\Theta - 1}.$$

Note, that OPT can do this even if the capacity is $c = 1$, since σ_2 does not need to be carried over position 1, where σ_1 appears. Since we have $\varepsilon' = \frac{\Theta+1}{2\Theta}\varepsilon$, we obtain

$$\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} = \frac{2\Theta^2 - \Theta + 1}{\Theta^2 - 1} - \frac{2\varepsilon'\Theta}{\Theta + 1} = \frac{2\Theta^2 - \Theta + 1}{\Theta^2 - 1} - \varepsilon,$$

as claimed. □

Proposition 4 *Let $\frac{1}{2}(1 + \sqrt{5}) \leq \Theta \leq 2$. For every sufficiently small $\varepsilon > 0$ there is a set of requests σ such that SMARTERSTART immediately starts S_N after S_{N-1} and such that*

$$\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} \geq \frac{3\Theta^2 + 3}{2\Theta + 1} - \varepsilon,$$

i.e., the upper bound established in Proposition 2 is tight for $\Theta \in [\frac{1}{2}(1 + \sqrt{5}), 2] \approx [1.6180, 2]$.

Proof Let $\varepsilon > 0$ with $\varepsilon < \frac{1}{4}(\frac{5\Theta^2 - 9\Theta + 4}{2\Theta + 1})$ (note that $\frac{5\Theta^2 - 9\Theta + 4}{2\Theta + 1} > 0$ for $\Theta > 1$) and $\varepsilon' = \frac{2\Theta + 1}{5\Theta^2 - 9\Theta + 4}\varepsilon$. Let the request

$$\sigma_1 = (1, 1; 0)$$

appear. For all $t \geq 0$, we have $L(t, 0, \{\sigma_1\}) = 1$. Thus, SMARTERSTART starts its first schedule S_1 at time $t_1 = \frac{1}{\Theta-1}$ and reaches position $p_2 = 1$ at time $\frac{\Theta}{\Theta-1}$. Next we let two new requests

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

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

appear. For $t \geq \frac{\Theta}{\Theta - 1}$ we have

$$\begin{aligned} L(t, 0, \{\sigma_1, \sigma_2^{(1)}, \sigma_2^{(2)}\}) &= \left| 0 - \left(-\frac{1}{\Theta - 1} \right) \right| + \varepsilon' \\ &\quad + \left| \left(-\frac{1}{\Theta - 1} \right) - \left(2 + \frac{1}{\Theta - 1} - 2\varepsilon' \right) \right| \\ &= \frac{3}{\Theta - 1} + 2 - \varepsilon'. \end{aligned}$$

Thus, the second schedule S_2 is not started before time

$$\frac{L\left(\frac{\Theta}{\Theta - 1}, 0, \{\sigma_1, \sigma_2^{(1)}, \sigma_2^{(2)}\}\right)}{\Theta - 1} = \frac{3}{(\Theta - 1)^2} + \frac{2 - \varepsilon'}{\Theta - 1}.$$

By assumption of the lemma, we have $\Theta < 2$ and $\varepsilon < \frac{1}{4} \left(\frac{5\Theta^2 - 9\Theta + 4}{2\Theta + 1} \right)$, i.e., $\varepsilon' < \frac{1}{4}$, which implies that for the time $\frac{\Theta}{\Theta - 1}$, when SMARTERSTART reaches position $p_2 = 1$, the inequality

$$\begin{aligned} \frac{L\left(\frac{\Theta}{\Theta - 1}, 0, \{\sigma_1, \sigma_2^{(1)}, \sigma_2^{(2)}\}\right)}{\Theta - 1} &= \frac{3}{(\Theta - 1)^2} + \frac{2 - \varepsilon'}{\Theta - 1} \\ &\stackrel{\varepsilon' < 2}{>} \frac{3}{(\Theta - 1)^2} \stackrel{\Theta < 2}{>} \frac{\Theta}{\Theta - 1} \end{aligned} \tag{55}$$

holds. (Note that inequality (55) also holds for slightly larger Θ if we let $\varepsilon \rightarrow 0$.) Because of inequality (55), SMARTERSTART has a waiting period and starts the schedule S_2 at time

$$t_2 = \frac{L\left(\frac{\Theta}{\Theta - 1}, 0, \{\sigma_1, \sigma_2^{(1)}, \sigma_2^{(2)}\}\right)}{\Theta - 1} = \frac{3}{(\Theta - 1)^2} + \frac{2 - \varepsilon'}{\Theta - 1}.$$

If SMARTERSTART serves $\sigma_2^{(2)}$ before serving $\sigma_2^{(1)}$ the time it needs is at least

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

The best schedule that serves $\sigma_2^{(2)}$ after serving $\sigma_2^{(1)}$ needs time

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

$$= 3 + \frac{3}{\Theta - 1} - 4\varepsilon'.$$

Thus, SMARTERSTART serves $\sigma_2^{(2)}$ after serving $\sigma_2^{(1)}$ and finishes S_2 at position $p_3 = -\frac{1}{\Theta-1}$ at time

$$t_2 + L(t_2, p_2, \{\sigma_2^{(1)}, \sigma_2^{(2)}\}) = \frac{3}{(\Theta - 1)^2} + \frac{2 - \varepsilon'}{\Theta - 1} + 3 + \frac{3}{\Theta - 1} - 4\varepsilon'.$$

Now let the final request

$$\sigma_3 = \left(\frac{3}{(\Theta - 1)^2} - \varepsilon', \frac{3}{(\Theta - 1)^2} - \varepsilon'; \frac{3}{(\Theta - 1)^2} + \frac{2}{\Theta - 1} \right)$$

appear. By assumption, we have $\Theta < 2$, which implies

$$2 + \frac{1}{\Theta - 1} - 2\varepsilon' = \frac{2\Theta - 1}{\Theta - 1} - 2\varepsilon' \stackrel{\Theta < 2}{<} \frac{3}{(\Theta - 1)^2} - \varepsilon',$$

i.e., the position of the request σ_3 lies to the right of $\sigma_2^{(1)}$. Thus, for all $t \geq \frac{3}{(\Theta-1)^2} + \frac{2-\varepsilon'}{\Theta-1} + 3 + \frac{3}{\Theta-1} - 4\varepsilon'$, we have the equation

$$\begin{aligned} L(t, 0, \{\sigma_1, \sigma_2^{(1)}, \sigma_2^{(2)}, \sigma_3\}) &= \left| 0 - \left(-\frac{1}{\Theta - 1}\right) \right| + \left| \left(-\frac{1}{\Theta - 1}\right) - \frac{3}{(\Theta - 1)^2} - \varepsilon' \right| \\ &= \frac{2}{\Theta - 1} + \frac{3}{(\Theta - 1)^2} - \varepsilon'. \end{aligned}$$

Therefore, the final schedule is not started before time

$$\frac{L(t, 0, \{\sigma_1, \sigma_2^{(1)}, \sigma_2^{(2)}, \sigma_3\})}{\Theta - 1} = \frac{2}{(\Theta - 1)^2} + \frac{3}{(\Theta - 1)^3} - \frac{\varepsilon'}{\Theta - 1}.$$

However, by assumption, we have $\Theta \geq \frac{1}{2} (1 + \sqrt{5})$ and $\varepsilon < \frac{1}{4} (\frac{5\Theta^2 - 9\Theta + 4}{2\Theta + 1})$, i.e., $\varepsilon' < \frac{1}{4}$, which implies

$$\begin{aligned} t_2 + L(t_2, p_2, \{\sigma_2^{(1)}, \sigma_2^{(2)}\}) &= \frac{3}{(\Theta - 1)^2} + \frac{2 - \varepsilon'}{\Theta - 1} + 3 + \frac{3}{\Theta - 1} - 4\varepsilon' \\ &= \frac{3\Theta}{(\Theta - 1)^2} + \frac{2\Theta}{\Theta - 1} + 1 - 4\varepsilon' - \frac{\varepsilon'}{\Theta - 1} \\ &= \frac{3(\Theta(\Theta - 1))}{(\Theta - 1)^3} + \frac{2(\Theta(\Theta - 1))}{(\Theta - 1)^2} \\ &\quad + 1 - 4\varepsilon' - \frac{\varepsilon'}{\Theta - 1} \end{aligned}$$

$$\begin{aligned} \varepsilon' &< \frac{1}{4} \\ &> \frac{3(\Theta(\Theta - 1))}{(\Theta - 1)^3} + \frac{2(\Theta(\Theta - 1))}{(\Theta - 1)^2} - \frac{\varepsilon'}{\Theta - 1} \\ \Theta &\geq \frac{1}{2}(1 + \sqrt{5}) \\ &\geq \frac{3}{(\Theta - 1)^3} + \frac{2}{(\Theta - 1)^2} - \frac{\varepsilon'}{\Theta - 1} \\ &= \frac{L(t, 0, \{\sigma_1, \sigma_2^{(1)}, \sigma_2^{(2)}, \sigma_3\})}{\Theta - 1}, \end{aligned}$$

i.e., the starting time of the schedule S_3 is the ending time of the schedule S_2 and we have

$$t_3 = \frac{3}{(\Theta - 1)^2} + \frac{2 - \varepsilon'}{\Theta - 1} + 3 + \frac{3}{\Theta - 1} - 4\varepsilon'.$$

The schedule S_3 needs time

$$L(t_3, p_3, \{\sigma_3\}) = \left| \left(-\frac{1}{\Theta - 1} \right) - \left(\frac{3}{(\Theta - 1)^2} - \varepsilon' \right) \right| = \frac{1}{\Theta - 1} + \frac{3}{(\Theta - 1)^2} - \varepsilon'.$$

To sum it up, we have

$$\begin{aligned} \text{SMARTERSTART}(\sigma) &= t_3 + L(t_3, p_3, \{\sigma_3\}) \\ &= \frac{3}{(\Theta - 1)^2} + \frac{2 - \varepsilon'}{\Theta - 1} + 3 + \frac{3}{\Theta - 1} - 4\varepsilon' \\ &\quad + \frac{1}{\Theta - 1} + \frac{3}{(\Theta - 1)^2} - \varepsilon' \\ &= \frac{6}{(\Theta - 1)^2} + \frac{6}{\Theta - 1} + 3 - \frac{5\Theta - 4}{\Theta - 1} \varepsilon'. \end{aligned}$$

On the other hand, OPT goes from the origin straight to position $-\frac{1}{\Theta - 1}$ serving request $\sigma_2^{(2)}$ at time $\frac{1}{\Theta - 1} + \varepsilon'$ (i.e., it has to wait for ε' units of time after it reaches position $-\frac{1}{\Theta - 1}$). Then OPT walks straight from the origin to position $\frac{3}{(\Theta - 1)^2} - \varepsilon'$ serving all remaining requests. Thus, we have

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

Note that OPT can do this even if $c = 1$ since for all requests the starting point is equal to the ending point. Since we have $\varepsilon' = \frac{2\Theta + 1}{5\Theta^2 - 9\Theta + 4} \varepsilon$, we finally obtain

$$\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} = \frac{\frac{6}{(\Theta - 1)^2} + \frac{6}{\Theta - 1} + 3 - \frac{5\Theta - 4}{\Theta - 1} \varepsilon'}{\frac{2}{\Theta - 1} + \frac{3}{(\Theta - 1)^2}}$$

$$\begin{aligned}
 &= \frac{3\Theta^2 + 3}{2\Theta + 1} - \frac{5\Theta^2 - 9\Theta + 4}{2\Theta + 1} \varepsilon' \\
 &= \frac{3\Theta^2 + 3}{2\Theta + 1} - \varepsilon,
 \end{aligned}$$

as claimed. □

Recall that the optimal parameter Θ^* established in Theorem 3 is the only positive, real solution of the equation

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

which is $\Theta^* \approx 1.7125$. Therefore, according to Proposition 3 and Proposition 4 the parameter Θ^* lies in the range where the upper bounds of Propositions 1 and 2 are both tight. It remains to make sure that for all Θ that lie outside of this range the competitive ratio of $\text{SMARTERSTART}_\Theta$ is larger than $\rho^* \approx 2.6662$. Let $\varepsilon > 0$ with $\varepsilon < \frac{4\Theta+4}{\Theta-1} \cdot \min\{\frac{\Theta}{2\Theta-2}, \frac{\Theta^2-\Theta-2}{(\Theta-1)^2}, \frac{1}{\Theta-1}\}$ (note that $\frac{\Theta^2-\Theta-2}{(\Theta-1)^2} > 0$ for $\Theta > 2$) and $\varepsilon' = \frac{\Theta-1}{4\Theta+4} \varepsilon$. Consider the set of requests $\sigma_{\Theta>2} = \{\sigma_1, \sigma_2^{(1)}, \sigma_2^{(2)}, \sigma_3\}$ with

$$\begin{aligned}
 \sigma_1 &:= (1, 1; 0), \\
 \sigma_2^{(1)} &:= \left(\frac{\Theta - 2}{2\Theta - 2} + \varepsilon', 1; \frac{1}{\Theta - 1} + \varepsilon' \right), \\
 \sigma_2^{(2)} &:= \left(-\frac{1}{\Theta - 1} + \varepsilon', -\frac{1}{\Theta - 1} + \varepsilon'; \frac{1}{\Theta - 1} + \varepsilon' \right), \\
 \sigma_3 &:= \left(1, 1; \frac{\Theta + 1}{(\Theta - 1)^2} + \varepsilon' \right).
 \end{aligned}$$

We compute SMARTERSTART 's completion time for the set of requests $\sigma_{\Theta>2}$ in the case $2 < \Theta \leq 1 + \sqrt{2}$ and in the case $\Theta > 1 + \sqrt{2}$.

Lemma 11 *Let the capacity $c \in \mathbb{N} \cup \{\infty\}$ of the server be arbitrary but fixed and let $2 < \Theta \leq 1 + \sqrt{2}$. We have*

$$\frac{\text{SMARTERSTART}(\sigma_{\Theta>2})}{\text{OPT}(\sigma_{\Theta>2})} \geq \frac{3\Theta^2 - 2\Theta + 1}{\Theta^2 - 1} - \varepsilon.$$

In particular, we have

$$\frac{\text{SMARTERSTART}(\sigma_{\Theta>2})}{\text{OPT}(\sigma_{\Theta>2})} > \rho^* \approx 2.6662.$$

for $\Theta \in (2, 1 + \sqrt{2}] \approx (2, 2.4142]$ and sufficiently small ε .

Proof For all $t \geq 0$, we have $L(t, 0, \{\sigma_1\}) = 1$. Thus, SMARTERSTART starts its first schedule S_1 at time $t_1 = \frac{\Theta}{\Theta-1}$ and reaches position $p_2 = 1$ at time $\frac{\Theta}{\Theta-1}$. We have $\varepsilon < \frac{4\Theta+4}{\Theta-1} \cdot \frac{\Theta}{2\Theta-2}$, i.e., $\varepsilon' < \frac{\Theta}{2\Theta-2}$, which implies

$$0 < \frac{\Theta - 2}{2\Theta - 2} + \varepsilon' \stackrel{\varepsilon' < \frac{\Theta}{2\Theta-2}}{<} 1$$

for $\Theta > 2$, i.e. the starting position of $\sigma_2^{(1)}$ is between 0 and 1. For $t \geq \frac{\Theta}{\Theta-1}$ we have

$$\begin{aligned} L(t, 0, \{\sigma_1, \sigma_2^{(1)}, \sigma_2^{(2)}\}) &= \left| 0 - \left(-\frac{1}{\Theta - 1} + \varepsilon' \right) \right| + 2\varepsilon' + \left| \left(-\frac{1}{\Theta - 1} + \varepsilon' \right) - 1 \right| \\ &= \frac{2}{\Theta - 1} + 1. \end{aligned}$$

Thus, the second schedule S_2 is not started before time

$$\frac{L\left(\frac{\Theta}{\Theta-1}, 0, \{\sigma_1, \sigma_2^{(1)}, \sigma_2^{(2)}\}\right)}{\Theta - 1} = \frac{2}{(\Theta - 1)^2} + \frac{1}{\Theta - 1} = \frac{\Theta + 1}{(\Theta - 1)^2}.$$

By assumption of the lemma, we have $\Theta \leq 1 + \sqrt{2}$, which implies that for the time $\frac{\Theta}{\Theta-1}$, when SMARTERSTART reaches position $p_2 = 1$, the inequality

$$\frac{L\left(\frac{\Theta}{\Theta-1}, 0, \{\sigma_1, \sigma_2^{(1)}, \sigma_2^{(2)}\}\right)}{\Theta - 1} = \frac{\Theta + 1}{(\Theta - 1)^2} \geq \frac{\Theta}{\Theta - 1}$$

holds. Thus, SMARTERSTART has a waiting period and starts the schedule S_2 at time

$$t_2 = \frac{L\left(\frac{\Theta}{\Theta-1}, 0, \{\sigma_1, \sigma_2^{(1)}, \sigma_2^{(2)}\}\right)}{\Theta - 1} = \frac{\Theta + 1}{(\Theta - 1)^2}.$$

If SMARTERSTART serves $\sigma_2^{(2)}$ before serving $\sigma_2^{(1)}$, the time it needs is at least

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

The best schedule that serves $\sigma_2^{(2)}$ after serving $\sigma_2^{(1)}$ needs time

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

Thus, SMARTERSTART serves $\sigma_2^{(2)}$ after serving $\sigma_2^{(1)}$ and finishes S_2 at position $p_3 = -\frac{1}{\Theta-1} + \varepsilon'$ at time

$$t_2 + L(t_2, p_2, \{\sigma_2^{(1)}, \sigma_2^{(2)}\}) = \frac{\Theta + 1}{(\Theta - 1)^2} + \frac{2\Theta}{\Theta - 1} - 3\varepsilon' = \frac{2\Theta^2 - \Theta + 1}{(\Theta - 1)^2} - 3\varepsilon'.$$

For all $t \geq \frac{2\Theta^2 - \Theta + 1}{(\Theta - 1)^2} - 3\varepsilon'$, we have the equation

$$\begin{aligned} L(t, 0, \{\sigma_1, \sigma_2^{(1)}, \sigma_2^{(2)}, \sigma_3\}) &= \left| 0 - \left(-\frac{1}{\Theta - 1}\right) \right| + \left| \left(-\frac{1}{\Theta - 1}\right) - 1 \right| \\ &= \frac{2}{\Theta - 1} + 1. \end{aligned}$$

Therefore the final schedule is not started before time

$$\frac{L(t, 0, \{\sigma_1, \sigma_2^{(1)}, \sigma_2^{(2)}, \sigma_3\})}{\Theta - 1} = \frac{2}{(\Theta - 1)^2} + \frac{1}{\Theta - 1} = \frac{\Theta + 1}{(\Theta - 1)^2},$$

which is equal to t_2 and thus smaller than $t_2 + L(t_2, p_2, \{\sigma_2^{(1)}, \sigma_2^{(2)}\})$. Therefore, the starting time of the schedule S_3 is the ending time of the schedule S_2 and we have

$$t_3 = \frac{2\Theta^2 - \Theta + 1}{(\Theta - 1)^2} - 3\varepsilon'.$$

The schedule S_3 needs time

$$L(t_3, p_3, \{\sigma_3\}) = \left| \left(-\frac{1}{\Theta - 1} + \varepsilon\right) - 1 \right| = \frac{1}{\Theta - 1} + 1 - \varepsilon' = \frac{\Theta}{\Theta - 1} - \varepsilon'$$

To sum it up, we have

$$\begin{aligned} \text{SMARTERSTART}(\sigma) &= t_3 + L(t_3, p_3, \{\sigma_3\}) \\ &= \frac{2\Theta^2 - \Theta + 1}{(\Theta - 1)^2} - 3\varepsilon' + \frac{\Theta}{\Theta - 1} - \varepsilon' \\ &= \frac{3\Theta^2 - 2\Theta + 1}{(\Theta - 1)^2} - 4\varepsilon'. \end{aligned}$$

On the other hand, OPT goes from the origin straight to position $-\frac{1}{\Theta-1} + \varepsilon'$ serving request $\sigma_2^{(2)}$ at time $\frac{1}{\Theta-1} + \varepsilon'$ (i.e., it has to wait for $2\varepsilon'$ units of time after it reaches position $-\frac{1}{\Theta-1}$). Then OPT walks straight to position $\frac{\Theta-2}{2\Theta-2} + \varepsilon'$ collecting the request $\sigma_2^{(1)}$. Note that the release time of $\sigma_2^{(2)}$ is the same as of $\sigma_2^{(1)}$ and thus OPT has no waiting time at position $\frac{\Theta-2}{2\Theta-2} + \varepsilon'$. OPT reaches position 1 and delivers $\sigma_2^{(1)}$ at time

$\frac{2}{\Theta-1} + 1 = \frac{\Theta+1}{\Theta-1}$. By assumption we have $\Theta > 2$ and $\varepsilon < \frac{4\Theta+4}{\Theta-1} \cdot \frac{\Theta^2-\Theta-2}{(\Theta-1)^2}$, i.e., $\varepsilon' < \frac{\Theta^2-\Theta-2}{(\Theta-1)^2}$, which implies

$$\frac{\Theta + 1}{\Theta - 1} = \frac{\Theta + 1}{(\Theta - 1)^2} + \frac{\Theta^2 - \Theta - 2}{(\Theta - 1)^2} > \frac{\Theta + 1}{(\Theta - 1)^2} + \varepsilon'.$$

Thus, OPT has no waiting time at position 1 and can serve the requests σ_1 and σ_3 at arrival. To sum it up, we have

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

Note that OPT can do this even if $c = 1$ since $\sigma_2^{(1)}$ is the only transportation request and no other request lies between its starting position and destination. Since we have $\varepsilon' = \frac{\Theta-1}{4\Theta+4}\varepsilon$, we finally obtain

$$\begin{aligned} \frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} &= \frac{\frac{3\Theta^2-2\Theta+1}{(\Theta-1)^2} - 4\varepsilon'}{\frac{\Theta+1}{\Theta-1}} \\ &= \frac{3\Theta^2 - 2\Theta + 1}{\Theta^2 - 1} - \varepsilon =: g_1(\Theta) - \varepsilon. \end{aligned}$$

The function g_1 is monotonically decreasing on $(2, 1 + \sqrt{2}]$. Therefore, we have

$$\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} + \varepsilon > g_1(1 + \sqrt{2}) = 2\sqrt{2} > 2.82 > \rho^*$$

for all $\Theta \in (2, 1 + \sqrt{2}]$, and thus $\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} > \rho^*$ for sufficiently small ε . \square

Lemma 12 *Let the capacity $c \in \mathbb{N} \cup \{\infty\}$ of the server be arbitrary but fixed and let $\Theta > 1 + \sqrt{2}$. We have*

$$\frac{\text{SMARTERSTART}(\sigma_{\Theta>2})}{\text{OPT}(\sigma_{\Theta>2})} \geq \frac{4\Theta}{\Theta + 1} - \varepsilon.$$

In particular, we have

$$\frac{\text{SMARTERSTART}(\sigma_{\Theta>2})}{\text{OPT}(\sigma_{\Theta>2})} > \rho^* \approx 2.6662.$$

for $\Theta \in (1 + \sqrt{2}, \infty) \approx (2.4142, \infty)$ and sufficiently small ε .

Proof For all $t \geq 0$, we have $L(t, 0, \{\sigma_1\}) = 1$. Thus, SMARTERSTART starts its first schedule S_1 at time $t_1 = \frac{1}{\Theta-1}$ and reaches position $p_2 = 1$ at time $\frac{\Theta}{\Theta-1}$. We have $\varepsilon < \frac{4\Theta+4}{\Theta-1} \cdot \frac{\Theta}{2\Theta-2}$, i.e., $\varepsilon' < \frac{\Theta}{2\Theta-2}$, which implies

$$0 < \frac{\Theta - 2}{2\Theta - 2} + \varepsilon' \stackrel{\varepsilon' < \frac{\Theta}{2\Theta-2}}{<} 1$$

for $\Theta > 2$, i.e. the starting position of $\sigma_2^{(1)}$ is between 0 and 1. For $t \geq \frac{\Theta}{\Theta-1}$ we have

$$\begin{aligned} L(t, 0, \{\sigma_1, \sigma_2^{(1)}, \sigma_2^{(2)}\}) &= \left| 0 - \left(-\frac{1}{\Theta - 1} + \varepsilon' \right) \right| + 2\varepsilon' + \left| \left(-\frac{1}{\Theta - 1} + \varepsilon' \right) - 1 \right| \\ &= \frac{2}{\Theta - 1} + 1. \end{aligned}$$

Thus, the second schedule S_2 is not started before time

$$\frac{L\left(\frac{\Theta}{\Theta-1}, 0, \{\sigma_1, \sigma_2^{(1)}, \sigma_2^{(2)}\}\right)}{\Theta - 1} = \frac{2}{(\Theta - 1)^2} + \frac{1}{\Theta - 1} = \frac{\Theta + 1}{(\Theta - 1)^2}.$$

By assumption of the lemma, we have $\Theta > 1 + \sqrt{2}$, which implies that for the time $\frac{\Theta}{\Theta-1}$, when SMARTERSTART reaches position $p_2 = 1$, the inequality

$$\frac{L\left(\frac{\Theta}{\Theta-1}, 0, \{\sigma_1, \sigma_2^{(1)}, \sigma_2^{(2)}\}\right)}{\Theta - 1} = \frac{\Theta + 1}{(\Theta - 1)^2} < \frac{\Theta}{\Theta - 1}$$

holds. Thus, SMARTERSTART has no waiting period and the starting time of the schedule S_2 is the ending time of the schedule S_1 . We have

$$t_2 \frac{\Theta}{\Theta - 1}.$$

If SMARTERSTART serves $\sigma_2^{(2)}$ before serving $\sigma_2^{(1)}$ the time it needs is at least

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

The best schedule that serves $\sigma_2^{(2)}$ after serving $\sigma_2^{(1)}$ needs time

$$\begin{aligned} &\left| 1 - \left(\frac{\Theta - 2}{2\Theta - 2} + \varepsilon' \right) \right| + \left| \left(\frac{\Theta - 2}{2\Theta - 2} + \varepsilon' \right) - 1 \right| + \left| 1 - \left(-\frac{1}{\Theta - 1} + \varepsilon' \right) \right| \\ &= \frac{\Theta}{2\Theta - 2} - \varepsilon' + \frac{\Theta}{2\Theta - 2} - \varepsilon' + \frac{\Theta}{\Theta - 1} - \varepsilon' \end{aligned}$$

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

Thus, SMARTERSTART serves $\sigma_2^{(2)}$ after serving $\sigma_2^{(1)}$ and finishes S_2 at position $p_3 = -\frac{1}{\Theta-1} + \varepsilon'$ at time

$$t_2 + L(t_2, p_2, \{\sigma_2^{(1)}, \sigma_2^{(2)}\}) = \frac{\Theta}{\Theta - 1} + \frac{2\Theta}{\Theta - 1} - 3\varepsilon' = \frac{3\Theta}{\Theta - 1} - 3\varepsilon'.$$

For all $t \geq \frac{3\Theta}{\Theta-1} - 3\varepsilon'$, we have the equation

$$\begin{aligned} L(t, 0, \{\sigma_1, \sigma_2^{(1)}, \sigma_2^{(2)}, \sigma_3\}) &= \left| 0 - \left(-\frac{1}{\Theta - 1}\right) \right| + \left| \left(-\frac{1}{\Theta - 1}\right) - 1 \right| \\ &= \frac{2}{\Theta - 1} + 1. \end{aligned}$$

Therefore the final schedule is not started before time

$$\frac{L(t, 0, \{\sigma_1, \sigma_2^{(1)}, \sigma_2^{(2)}, \sigma_3\})}{\Theta - 1} = \frac{2}{(\Theta - 1)^2} + \frac{1}{\Theta - 1} = \frac{\Theta + 1}{(\Theta - 1)^2},$$

which is, as before, smaller than t_2 , and thus also smaller than time $t_2 + L(t_2, p_2, \{\sigma_2^{(1)}, \sigma_2^{(2)}\})$. Therefore, the starting time of the schedule S_3 is the ending time of the schedule S_2 and we have

$$t_3 = \frac{3\Theta}{\Theta - 1} - 3\varepsilon'.$$

The schedule S_3 needs time

$$L(t_3, p_3, \{\sigma_3\}) = \left| \left(-\frac{1}{\Theta - 1} + \varepsilon\right) - 1 \right| = \frac{1}{\Theta - 1} + 1 - \varepsilon' = \frac{\Theta}{\Theta - 1} - \varepsilon'.$$

To sum it up, we have

$$\begin{aligned} \text{SMARTERSTART}(\sigma) &= t_3 + L(t_3, p_3, \{\sigma_3\}) \\ &= \frac{3\Theta}{\Theta - 1} - 3\varepsilon' + \frac{\Theta}{\Theta - 1} - \varepsilon' \\ &= \frac{4\Theta}{\Theta - 1} - 4\varepsilon'. \end{aligned}$$

On the other hand, OPT goes from the origin straight to position $-\frac{1}{\Theta-1} + \varepsilon'$ serving request $\sigma_2^{(2)}$ at time $\frac{1}{\Theta-1} + \varepsilon'$ (i.e., it has to wait for $2\varepsilon'$ units of time after it reaches position $-\frac{1}{\Theta-1}$). Then OPT walks straight to position $\frac{\Theta-2}{2\Theta-2} + \varepsilon'$ collecting the request $\sigma_2^{(1)}$. Note that the release time of $\sigma_2^{(2)}$ is the same as of $\sigma_2^{(1)}$ and thus OPT has no

waiting time at position $\frac{\Theta-2}{2\Theta-2} + \varepsilon'$. OPT reaches position 1 and delivers $\sigma_2^{(1)}$ at time $\frac{2}{\Theta-1} + 1 = \frac{\Theta+1}{\Theta-1}$. By assumption of the lemma, we have $\Theta > 2$ and $\varepsilon < \frac{4\Theta+4}{\Theta-1} \cdot \frac{1}{\Theta-1}$, i.e., $\varepsilon' < \frac{1}{\Theta-1}$, which implies

$$\frac{\Theta + 1}{\Theta - 1} = \frac{\Theta}{\Theta - 1} + \frac{1}{\Theta - 1} > \frac{\Theta}{\Theta - 1} + \varepsilon'.$$

Thus, OPT has no waiting time at position 1 and can serve the requests σ_1 and σ_3 at arrival. To sum it up, we have

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

Note that OPT can do this even if $c = 1$ since $\sigma_2^{(1)}$ is the only transportation request and no other request lies between its starting position and destination. Since we have $\varepsilon' = \frac{\Theta-1}{4\Theta+4}\varepsilon$, we finally obtain

$$\begin{aligned} \frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} &= \frac{\frac{4\Theta}{\Theta-1} - 4\varepsilon'}{\frac{\Theta+1}{\Theta-1}} \\ &= \frac{4\Theta}{\Theta + 1} - \varepsilon =: g_2(\Theta) - \varepsilon. \end{aligned}$$

The function g_1 is monotonically increasing on $(1 + \sqrt{2}, \infty)$. Therefore, we have

$$\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} + \varepsilon > g_2(1 + \sqrt{2}) = 2\sqrt{2} > 2.82 > \rho^*$$

for all $\Theta \in (1 + \sqrt{2}, \infty)$, and thus $\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} > \rho^*$ for sufficiently small ε . \square

Lemma 13 *Let $\Theta > 2$. There is a set of requests $\sigma_{\Theta>2}$ such that*

$$\frac{\text{SMARTERSTART}(\sigma_{\Theta>2})}{\text{OPT}(\sigma_{\Theta>2})} > \rho^* \approx 2.6662.$$

Proof This is an immediate consequence of Lemma 11 and Lemma 12. \square

Figure 5 shows the upper and lower bounds that we have established. Theorem 2 now follows from Theorem 3 combined with Propositions 3 and 4, as well as Lemma 13.

Proof of Theorem 2 We have shown in Proposition 3 that the upper bound

$$\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} \leq f_1(\Theta) = \frac{2\Theta^2 - \Theta + 1}{\Theta^2 - 1}$$

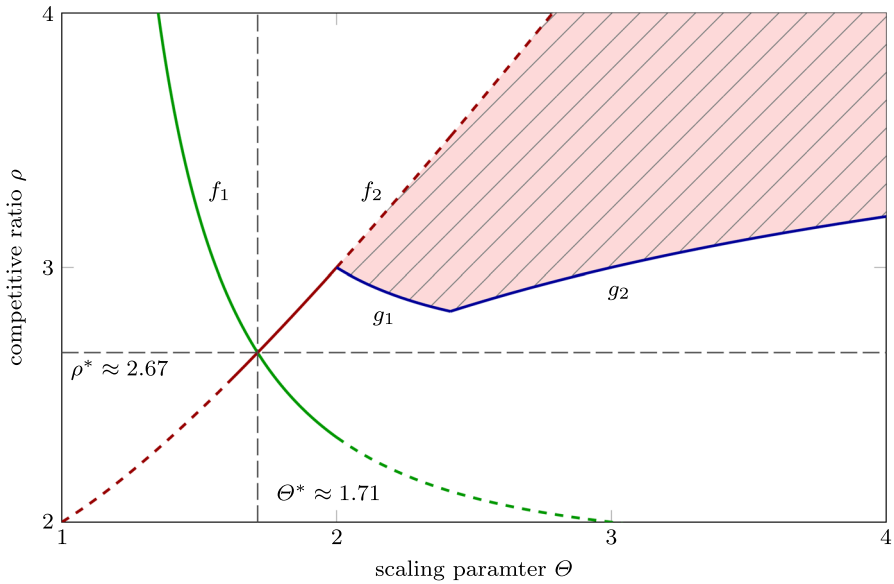


Fig. 5 Overview of our bounds for SMARTERSTART. The functions f_1 (green) / f_2 (red) are upper bounds for the cases where SMARTERSTART waits / does not wait before starting the final schedule, respectively. The upper bounds are drawn solid in the domains where they are tight for their corresponding case. The functions g_1 and g_2 (blue) are general lower bounds (Color figure online)

established in Proposition 1 for the case, where SMARTERSTART waits before starting the final schedule, is tight for all $\theta \in (1, 2)$. Furthermore, we have shown in Proposition 4 that the upper bound

$$\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} \leq f_2(\theta) = \frac{3\theta^2 + 3}{2\theta + 1}$$

established in Proposition 2 for the case, where SMARTERSTART does not wait before starting the final schedule, is tight for all $\theta \in (\frac{1}{2}(1 + \sqrt{5}), 2]$. Since $\theta^* \approx 1.71249$ lies in those ranges, the competitive ratio of $\text{SMARTERSTART}_{\theta^*}$ 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 13, the competitive ratio of SMARTERSTART with parameter $\theta \in (2, \infty)$ is larger than ρ^* . By monotonicity of f_1 , every function value in $(1, \theta^*)$ is larger than $f_1(\theta^*) = \rho^*$. Thus, the competitive ratio of SMARTERSTART with parameter $\theta \in (1, \theta^*)$ is larger than ρ^* , since f_1 exactly captures the worst-case behavior of SMARTERSTART for $\theta \in (1, \theta^*)$ by Proposition 3. Similarly, by monotonicity of f_2 , every function value in $(\theta^*, 2]$ is larger than $f_2(\theta^*) = \rho^*$. Thus, the competitive ratio of SMARTERSTART with parameter $\theta \in (\theta^*, 2]$ is larger than ρ^* , since f_2 exactly captures the worst-case behavior of SMARTERSTART for $\theta \in (\theta^*, 2]$ by Proposition 4. □

References

1. Ascheuer, N., Krumke, S.O., Rambau, J.: Online dial-a-ride problems: minimizing the completion time. In: Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science (STACS), pp. 639–650 (2000)
2. Atallah, M.J., Kosaraju, S.R.: Efficient solutions to some transportation problems with applications to minimizing robot arm travel. *SIAM J. Comput.* **17**(5) (1988)
3. Ausiello, G., Feuerstein, E., Leonardi, S., Stougie, L., Talamo, M.: Algorithms for the on-line travelling salesman. *Algorithmica* **29**(4), 560–581 (2001)
4. Bienkowski, M., Kraska, A., Liu, H.: Traveling repairperson, unrelated machines, and other stories about average completion times. In: Bansal, N., Merelli, E., Worrell, J. (eds.), Proceedings of the 48th International Colloquium on Automata, Languages, and Programming (ICALP), pp. 28:1–28:20 (2021)
5. Bienkowski, M., Liu, H.: An improved online algorithm for the traveling repairperson problem on a line. In: Proceedings of the 44th International Symposium on Mathematical Foundations of Computer Science (MFCS), pp. 6:1–6:12 (2019)
6. Birx, A.: Competitive analysis of the online Dial-a-Ride problem. Ph.D. thesis, TU Darmstadt (2020)
7. Birx, A., Disser, Y.: Tight analysis of the smartstart algorithm for online Dial-a-Ride on the line. *SIAM J. Discrete Math.* **34**(2), 1409–1443 (2020)
8. Birx, A., Disser, Y., Schewior, K.: Improved bounds for open online Dial-a-Ride on the line. In: Proceedings of the 22nd International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX), volume 145, page 21(22) (2019)
9. Bjelde, A., Disser, Y., Hackfeld, J., Hansknecht, C., Lipmann, M., Meißner, J., Schewior, K., Schlöter, M., Stougie, L.: Tight bounds for online tsp on the line. *ACM Trans. Algorithms* **17**(1) (2020)
10. Blom, M., Krumke, S.O., de Paepe, W.E., Stougie, L.: The online TSP against fair adversaries. *INFORMS J. Comput.* **13**(2), 138–148 (2001)
11. Charikar, M., Raghavachari, B.: The finite capacity dial-a-ride problem. In: Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS), pp. 458–467 (1998)
12. de Paepe, W.E., Lenstra, J.K., Sgall, J., Sitters, R.A., Stougie, L.: Computer-aided complexity classification of Dial-a-Ride problems. *INFORMS J. Comput.* **16**(2), 120–132 (2004)
13. Feuerstein, E., Stougie, L.: On-line single-server Dial-a-Ride problems. *Theoret. Comput. Sci.* **268**(1), 91–105 (2001)
14. Gilmore, P.C., Gomory, R.E.: Sequencing a one state-variable machine: a solvable case of the traveling salesman problem. *Oper. Res.* **12**(5), 655–679 (1964)
15. Guan, D.J.: Routing a vehicle of capacity greater than one. *Discrete Appl. Math.* **81**(1–3), 41–57 (1998)
16. Hauptmeier, D., Krumke, S.O., Rambau, J.: The online Dial-a-Ride problem under reasonable load. In: Proceedings of the 4th Italian Conference on Algorithms and Complexity (CIAC), pp. 125–136 (2000)
17. Jaillet, P., Wagner, M.R.: Generalized online routing: new competitive ratios, resource augmentation, and asymptotic analyses. *Oper. Res.* **56**(3), 745–757 (2008)
18. Krumke, S.O.: Online optimization competitive analysis and beyond. Habilitation thesis (2001)
19. Krumke, S.O., de Paepe, W.E., Poensgen, D., Lipmann, M., Marchetti-Spaccamela, A., Stougie, L.: 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), pp. 258–269 (2006)
20. Krumke, S.O., de Paepe, W.E., Poensgen, D., Stougie, L.: News from the online traveling repairman. *Theoret. Comput. Sci.* **295**(1–3), 279–294 (2003)
21. Krumke, S.O., Laura, L., Lipmann, M., Marchetti-Spaccamela, A., de Paepe, W., Poensgen, D., Stougie, L.: 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), pp. 200–214 (2002)
22. Lipmann, M., Lu, X., de Paepe, W.E., Sitters, R.A., Stougie, L.: On-line Dial-a-Ride problems under a restricted information model. *Algorithmica* **40**(4), 319–329 (2004)
23. Yi, F., Tian, L.: On the online dial-a-ride problem with time-windows. In: Proceedings of the 1st International Conference on Algorithmic Applications in Management (AAIM), pp. 85–94 (2005)
24. Yi, F., Xu, Y., Xin, C.: 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), pp. 22–31 (2006)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.