Tight Bounds for Online TSP on the Line

ANTJE BJELDE and JAN HACKFELD, Humboldt-Universität Berlin, Germany
YANN DISSER, Technische Universität Darmstadt, Germany
CHRISTOPH HANSKNECHT, Technische Universität Braunschweig, Germany
MAARTEN LIPMANN, Amsterdam, The Netherlands
JULIE MEIßNER, Technische Universität Berlin, Germany
MIRIAM SCHLÖTER, Eidgenössische Technische Hochschule Zürich, Switzerland
KEVIN SCHEWIOR, Universität zu Köln, Germany
LEEN STOUGIE, Centrum Wiskunde & Informatica and Vrije Universiteit and INRIA-Erable

We consider the online traveling salesperson problem (TSP), where requests appear online over time on the real line and need to be visited by a server initially located at the origin. We distinguish between closed and open online TSP, depending on whether the server eventually needs to return to the origin or not. While online TSP on the line is a very natural online problem that was introduced more than two decades ago, no tight competitive analysis was known to date. We settle this problem by providing tight bounds on the competitive ratios for both the closed and the open variant of the problem. In particular, for closed online TSP, we provide a 1.64-competitive algorithm, thus matching a known lower bound. For open online TSP, we give a new upper bound as well as a matching lower bound that establish the remarkable competitive ratio of 2.04.

Additionally, we consider the online Dial-A-Ride problem on the line, where each request needs to be transported to a specified destination. We provide an improved non-preemptive lower bound of 1.75 for this setting, as well as an improved preemptive algorithm with competitive ratio 2.41.

Finally, we generalize known and give new complexity results for the underlying offline problems. In particular, we give an algorithm with running time $O(n^2)$ for closed offline TSP on the line with release dates and show that both variants of offline Dial-A-Ride on the line are NP-hard for any capacity $c \geq 2$ of the server.

CCS Concepts: • **Theory of computation** → Computational complexity and cryptography; *Approximation algorithms analysis*; **Online algorithms**;

Additional Key Words and Phrases: TSP, online algorithms, competitive analysis, approximation algorithms, computational complexity

A preliminary version of this article appeared in Reference [8].
Leen Stougie was supported by the NWO in the Gravitation Programme Networks, Grant No. 024.002.003.
Authors' addresses: A. Bjelde and J. Hackfeld, Humboldt-Universität Berlin, Berlin, Germany; email: {antje.bjelde, jan.hackfeld}@hu-berlin.de; Y. Disser, Department of Mathematics, Technische Universität Darmstadt, Dolivostraße 15, 64293 Darmstadt, Germany; email: disser@mathematik. tu-darmstadt.de; C. Hansknecht, Technische Universität Braunschweig, Braunschweig, Germany; email: c.hansknecht@tu-braunschweig.de; M. Lipmann, Amsterdam, Amsterdam, The Netherlands; email: Maarten.lipmann@gmail.com; J. Meißner, Technische Universität Berlin, Berlin, Germany; email: jmeiss@math.tu-berlin.de; M. Schlöter, Eidgenössische Technische Hochschule Zürich, Zurich, Switzerland; email: miriam.schloeter@ifor.math.ethz.ch; K. Schewior, Universität zu Köln, Cologne, Germany; email: kschewior@gmail.com; L. Stougie, Centrum Wiskunde & Informatica, Amsterdam, The Netherlands; Vrije Universiteit; and INRIA-Erable; email: Leen.Stougie@cwi.nl.

1549-6325/2020/12-ART3 $15.00
https://doi.org/10.1145/3422362

ACM Reference format:
Antje Bjelde, Jan Hackfeld, Yann Disser, Christoph Hansknecht, Maarten Lipmann, Julie Meißner, Miriam Schlöter, Kevin Schewior, and Leen Stougie. 2020. Tight Bounds for Online TSP on the Line. *ACM Trans. Algorithms* 17, 1, Article 3 (December 2020), 58 pages.
https://doi.org/10.1145/3422362

1 INTRODUCTION

In the online Traveling Salesperson Problem (TSP) on the line, we consider a server initially located at the origin of the real line that has to serve requests that appear over time. The server has unit speed and serves requests (in any order) by moving to the position of the corresponding request at some time after its release. The objective in online TSP on the line is to minimize the makespan, i.e., the time until all requests have been served. In the *closed* variant of the problem, the server needs to return to the origin after serving all requests, while the *open* variant has no such requirement.

Online TSP is a natural online problem similar to the classical k-server problem [25]. In the latter, the order in which requests need to be served is prescribed, and the problem thus becomes trivial on the line for $k = 1$ server. In contrast, online TSP on the line is a non-trivial problem that arises in 1-dimensional collection/delivery problems. Examples include robotic welding/screwing/depositing material, horizontal/vertical item delivery systems, and the collection of objects from mass storage shelves or along shorelines [2, 26]. The online Dial-A-Ride problem additionally allows transportation requests that specify a source and destination that need to be visited by the server in this order. If the capacity of the server is finite, then it limits the number of requests that can be transported simultaneously. In that case, we distinguish the *preemptive* variant of the problem where requests can be unloaded at points other than their destination and be picked up later to be transported further to their destination, and the *non-preemptive* variant where this is not permitted. The online Dial-A-Ride problem on the line arises, e.g., when controlling robot arms and industrial or personal elevators [4, 11].

While both online TSP and online Dial-A-Ride on the line are among the most natural online problems and have been studied extensively over the past two decades [3, 5–7, 9, 12, 18, 20–22], no satisfactory (tight) analysis was known for either problem in terms of competitive ratios. We address this shortcoming for TSP on the line by providing a tight upper bound for the closed variant, as well as tight bounds for the open variant. We emphasize that our results for the open and closed variant of the problem are independent and require substantially different approaches. Aside from our results for online TSP, we narrow the gaps for online Dial-A-Ride on the line by giving improved bounds. In addition to online results, we study the computational complexity of the underlying offline problems.

1.1 Our Results

We have the following results[1] (cf. Tables 1 and 2):

Tight bounds for online TSP on the line. Our main results are best-possible online algorithms for both the open and closed variant of online TSP on the line, as well as a new (tight) lower bound for the open variant. Our algorithm for the closed variant has a competitive ratio of $(9 + \sqrt{17})/8 \approx 1.64$, matching a lower bound of Ausiello et al. [7] and improving on their 1.75-competitive algorithm. For open TSP on the line, we provide a 2.04-competitive online algorithm, improving on the 2.33-competitive algorithm by Ausiello et al. [7]. We show that this algorithm

[1]Parts of our results were already claimed in Reference [24], but mostly with weaker bounds and without a conclusive proof. Nevertheless, some of our ideas are inspired by the approaches described in Reference [24].

Table 1. Overview of Our Results for Online TSP on the Line and Online Dial-A-Ride on the Line

Online	Closed		Open	
	Lower bound	Upper bound	Lower bound	Upper bound
Online TSP on the line				
new		**1.64** (Th. 3.5)	**2.04** (Th. 4.1)	**2.04** (Th. 5.2)
old	1.64 [6, 7]	1.75 [6, 7]	2 [5, 7]	2.33 [5, 7]
Dial-A-Ride on the line				
preemptive	1.64 [6, 7]	2 [3]	**2.04** (Th. 4.1)	**2.41** (Th. 6.2)
non-preemptive	**1.75** (Th. 6.4)			3.41 [20]

Table 2. Overview of Our Results for Offline TSP and Dial-A-Ride on the Line with Release Times

Offline	Closed	Open
TSP on the line	$O(n^2)$ (Th. 7.4)	$O(n^2)$ [26]
Dial-A-Ride on the line		
non-preemptive	NP-hard (Th. 7.8)	NP-hard (Th. 7.8)

is best-possible by giving a matching lower bound. Our results settle online TSP on the line from the perspective of competitive analysis.

Improved bounds for online Dial-A-Ride on the line. Our lower bounds for online TSP on the line immediately apply to preemptive and non-preemptive online Dial-A-Ride on the line. In particular, our lower bound of 2.04 is the first bound greater than 2 for the open variant of the problem. Additionally, we provide a simple preemptive 2.41-competitive algorithm, which improves a (non-preemptive) 3.41-competitive algorithm by Krumke [20]. In the uncapacitated case, this algorithm can be generalized to work in the Euclidean space for both the open and closed Dial-A-Ride variants. For the closed variant on the line, the lower bound of 1.64 by Ausiello et al. [7] was improved for one server with unit capacity without preemption to 1.71 by Ascheuer et al. [3]. We improve this bound further to 1.75 for any finite capacity $c \geq 1$. The best known algorithm for closed Dial-A-Ride on the line for finite capacity $c \geq 1$ is 2-competitive and was given by Ascheuer et al. [3].

New offline complexity results. Regarding offline TSP on the line with release times, Psaraftis et al. [26] showed a dynamic program that solves the open variant in quadratic time. We refute their claim that all optimal closed tours have a very simple structure with a counterexample, and we adapt their algorithm to find an optimal closed tour in quadratic time. For the non-preemptive offline Dial-A-Ride problem on the line, results have previously been obtained for the closed variant without release times. For capacity $c = 1$ Gilmore and Gomory [15] and Atallah and Kosaraju [4] gave polynomial time algorithms, and Guan [16] proved hardness for the case $c = 2$. We show that both the open and closed variant of the problem are NP-hard for *any* capacity $c \geq 2$. Additionally, we show that the case with release times and any $c \geq 1$ is NP-hard. The complexity of offline Dial-A-Ride on the line with unbounded capacity remains open.

1.2 Further Related Work

For the online TSP problem in general metric spaces, Ausiello et al. [7] show a lower bound of 2 on the competitive ratio for the open version and a 1.64 lower bound for the closed version, both bounds being achieved on the real line. For the open online TSP, they present a 2.5-competitive algorithm, and for the closed version they give a 2-competitive algorithm. Jaillet and Wagner [18] give 2-competitive algorithms for the closed version that can additionally deal with precedence constraints or multiple servers. Blom et al. [9] consider the closed online TSP problem on the non-negative part of the real line and present a best possible algorithm with competitive ratio 1.5. They also study a "fair" setting where the optimum does not travel outside the convex hull of the known requests, and they derive an algorithm for the real half-line with a better competitive ratio of 1.28 for this setting. Krumke et al. [22] show that there cannot be a competitive algorithm for open online TSP with the objective of minimizing the maximum flow time instead of minimizing the makespan. For the real line they define a fair setting and give a competitive algorithm for it.

The *online repairperson problem* is the open online TSP problem with the objective of minimizing the weighted sum of completion times. Feuerstein and Stougie [12] show a lower bound of 2.41 on the best-possible competitive ratio for this problem and provide a 9-competitive algorithm for the real line. Krumke et al. [21] give an online algorithm with competitive ratio 5.83 for general metric spaces, and Hwang and Jaillet [17] improve this competitive ratio to 5.14.

For the closed online Dial-A-Ride problem without preemption, Feuerstein and Stougie [12] show a lower bound of 2 for the competitive ratio in general, and present an algorithm with a best-possible competitive ratio of 2 for the case that the server has infinite capacity. Ascheuer et al. [3] analyze different algorithms for the same setting and present a 2-competitive algorithm for any finite capacity $c \geq 1$. For minimizing the sum of completion times instead of the makespan, Feuerstein and Stougie [12] further show a lower bound of 3 for a server with unit capacity and a lower bound of 2.41 independent of the capacity. Moreover, they provide a 15-competitive algorithm for the real line and unlimited capacity. For the same objective function, Krumke et al. [21] present an algorithm with a competitive ratio of 5.83 for a server with unit capacity in an arbitrary metric space.

The offline version of the TSP problem is a well-studied NP-hard problem (e.g., see Reference [23]). Afrati et al. [1] show that the offline traveling repairperson problem on the line can be solved in polynomial time, but becomes hard if requests have deadlines. Sitters [27] shows that the problem without deadlines is hard on weighted trees. There are many offline variants of the Dial-A-Ride problem, differing in capacities, the underlying metric space, release times and deadlines, open versus closed tours, and in whether preemption is allowed (e.g., see Reference [11]). The special case without release times and unit capacity is known as the *stacker crane problem*. Attalah and Kosaraju [4] present a polynomial time algorithm for the closed, non-preemptive stacker crane problem on the real line. Frederickson and Guan [13] show that this problem is NP-complete on trees. Guan [16] shows that the Dial-A-Ride problem remains easy on the line with capacities larger than one if preemption is allowed, and that it remains hard on trees. Finally, Charikar and Raghavachari [10] give a $O(\sqrt{c}\log n \log\log n)$-approximation for the closed Dial-A-Ride problem in metric spaces with n points and without preemption. In the same paper, they claim a 2-approximation for the problem on the line, however this result seems to be incorrect (personal communication).

2 PROBLEM DEFINITION AND NOTATION

We consider a server that moves along the real line with (at most) unit speed. We let $\text{pos}(t)$ denote the position of the server at time $t \geq 0$ and assume (without loss of generality) that $\text{pos}(0) = 0$. With this notation, the speed limitation of the server can equivalently be expressed via $|\text{pos}(t) - \text{pos}(t')| \leq |t - t'|$ for all $t, t' \geq 0$. A series of requests $\sigma_1, \dots, \sigma_n$ arrives over time with $\sigma_i = (a_i, b_i; r_i)$, where $r_i \geq 0$ denotes the release time of the request and $a_i, b_i \in \mathbb{R}$ denote its source and target position, respectively.[2] For TSP, we have $a_i = b_i$ and write

[2]For readability, we use a semi-colon to separate source and destination *positions* of a request from its release *time*.

$\sigma_i = (a_i; r_i)$. If not stated otherwise, then we assume $r_1 \leq r_2 \leq \cdots \leq r_n$. We use the notation $p^{\mathrm{R}} := \max_{i=1,\ldots,n} \max\{a_i, b_i, 0\}$ to denote the rightmost point that needs to be visited by the server, and similarly $p^{\mathrm{L}} := \min_{i=1,\ldots,n} \min\{a_i, b_i, 0\}$. Here and throughout, we refer to the negative direction of the real line as *left* and the positive direction as *right*.

We observe that we may assume without loss of generality that $r_i \geq |a_i|$ holds: On the one hand, the server cannot reach σ_i before time $|a_i|$ and it only helps the algorithm to know a request earlier; on the other hand, any algorithm can simply ignore σ_i until time $|a_i|$. We make this assumption from now on.

Observation 1. *We may assume that $r_i \geq |a_i|$ holds, for all requests $\sigma_i = (a_i, b_i; r_i)$.*

In both TSP and Dial-A-Ride on the line, all requests need to be *served*. For TSP, we consider a request served if $\mathrm{pos}(t) = a_i$ for some time $t \geq r_i$. For Dial-A-Ride, the server may collect request σ_i at time $t \geq r_i$ if $\mathrm{pos}(t) = a_i$. In the preemptive Dial-A-Ride problem, the server can drop off any request it is carrying at its current location at any time. If request σ_i is dropped off at point $\mathrm{pos}(t)$ at time t, then we consider it to be modified to the new request $(\mathrm{pos}(t), b_i; t)$. In the non-preemptive Dial-A-Ride problem, the server may only drop off a request at its target location. We consider a request served if it is ever dropped off at its target location.

In TSP on the line, the behavior of the server in our algorithms at time t will mostly depend on so-called *extreme requests*. For $t \geq 0$, we denote by $\sigma^{\mathrm{R}}(t) = (\mathrm{R}(t); r^{\mathrm{R}}(t))$ the unserved request that is rightmost of the position of the server $\mathrm{pos}(t)$, provided such a request exists, i.e., the unserved request $\sigma = (a; t')$ with $t' \leq t$, $a > \mathrm{pos}(t)$, and maximizing a. Analogously, $\sigma^{\mathrm{L}}(t) = (\mathrm{L}(t); r^{\mathrm{L}}(t))$ denotes the unserved request that is leftmost of the position of the server $\mathrm{pos}(t)$. If there is more than one right-most (left-most) request, then we choose the one with the largest release time.

In the Dial-A-Ride, if the server has finite capacity $c \geq 1$, it can carry at most c requests at any time. We assume that no time is needed for picking up and dropping off requests, so that the server can pick up and drop off any number of requests at the same time, as long as its capacity is not exceeded.

We refer to a valid trajectory of the server together with the description of when it picks up and drops requests as a tour T. If the tour ends at 0, then we call it *closed*; otherwise, it is *open*. We denote the makespan of the tour T by $|T|$. The objective in the open (closed) version of both TSP and Dial-A-Ride is to find an open (closed) tour T that serves all requests and minimizes $|T|$.

In the *offline* setting, we assume all requests to be known from the start. We let T^{opt} denote an optimal offline tour. In the *online* setting, we assume that request σ_i is revealed at its release time r_i, at which point the tour of the server until time r_i must already have been fixed irrevocably. Note that until time r_i even the existence of the request σ_i is unknown. We measure the quality of an online algorithm via its competitive ratio, i.e., the supremum over all sequences of requests of the ratio between the makespan of the tour it produces and $|T^{\mathrm{opt}}|$.

To describe the trajectory of the server, we use the notation "move(a)" for the tour that moves the server from its current position with unit speed to the point $a \in \mathbb{R}$ and the notation "waituntil(s)" for the tour that keeps the server stationary until time s. We use the operator $\oplus$ to concatenate tours. For example, if T_0 is a tour of the server that ends at time t_0 at position $\mathrm{pos}(t_0)$, then $T_0 \oplus \mathrm{move}(a)$ describes the tour that ends at time $t_0 + |a - \mathrm{pos}(t_0)|$, is identical to the tour T_0 until time t_0 and satisfies $\mathrm{pos}(t) = \mathrm{pos}(t_0) + \mathrm{sign}(a - \mathrm{pos}(t_0))(t - t_0)$ for $t_0 \leq t \leq t_0 + |a - \mathrm{pos}(t_0)|$. Similarly, $T_0 \oplus \mathrm{waituntil}(s)$ is the tour that ends at time $\max\{t_0, s\}$, which is identical to the tour T_0 until time t_0 and that satisfies $\mathrm{pos}(t_0) = \mathrm{pos}(t)$ for all $t \in [t_0, \max\{t_0, s\}]$. Note that if $\max\{t_0, s\} = t_0$, we have $T_0 \oplus \mathrm{waituntil}(s) = T_0$. For TSP on the line, we do not explicitly specify when a request is served, but we assume that the server serves a request whenever possible, i.e., whenever the server passes the location of a request that is already released and not yet served.

3 ALGORITHM FOR CLOSED ONLINE TSP

In this section, we consider the closed online TSP problem and describe a best-possible algorithm with competitive ratio $\rho = (9 + \sqrt{17})/8 \approx 1.64$, where ρ is the nonnegative root of the polynomial $4x^2 - 9x + 4$.

We start by developing some intuition for our algorithm. In the following T^{alg} is the tour planned by an algorithm Alg. Observe that the decision of how to move the server at time t only depends on its position $\text{pos}(t)$ and the location of the left- and rightmost extreme requests $\sigma^{\text{L}}(t) = (\text{L}(t); r^{\text{L}}(t))$ and $\sigma^{\text{R}}(t) = (\text{R}(t); r^{\text{R}}(t))$: all other requests can be served during any tour serving $\sigma^{\text{L}}(t)$ and $\sigma^{\text{R}}(t)$. Remember that $\text{L}(t)$ and $\text{R}(t)$ are the locations of the leftmost and rightmost extreme at time t, respectively, and $r^{\text{L}}(t)$ and $r^{\text{R}}(t)$ are their respective release dates. The following lemma shows that in this setting, we can assume that all requests are extreme requests at the time they are released and $\text{L}(t) < 0$ and $\text{R}(t) > 0$, provided these extremes exist.

Lemma 3.1. *We can assume that the released requests fulfill the following properties without loss of generality:*

(1) All requests are extreme requests with a location outside the interval $[0, \text{pos}(t)]$ *at the time they are released.*
(2) We can assume $\text{L}(t) < 0$ *if the leftmost extreme exists and* $\text{R}(t) > 0$ *if the rightmost extreme exists.*

Proof. We can assume the first property without loss of generality as all requests with a location in the interval $(\text{L}(t), \text{R}(t))$ will be served while the algorithm serves the two extreme requests $\sigma^{\text{L}}(t)$ and $\sigma^{\text{R}}(t)$, and can therefore be ignored. Analogously, all requests in $[0, \text{pos}(t)]$ will be served at the latest when the server returns to the origin.

Suppose that $\text{L}(t) \geq 0$. Then $\sigma^{\text{L}}(t)$ will be served while the server returns to the origin after having served $\sigma^{\text{R}}(t)$. Hence, we can simply delete $\sigma^{\text{L}}(t)$ from the input in this case. The case $\text{R}(t) \leq 0$ is symmetrical. □

An algorithm Alg has, on a high level, three possible courses of action at a time t at which $|\text{L}(t)|, |\text{R}(t)| > 0$. Either Alg immediately decides to serve $\sigma^{\text{L}}(t)$ and $\sigma^{\text{R}}(t)$ in one of the two possible orders, or it waits for some time for additional information to make a more informed decision. Intuitively, the critical case for the competitiveness of Alg is the case where it decides to serve $\sigma^{\text{L}}(t)$ and $\sigma^{\text{R}}(t)$ in a different order than T^{opt}. If both $\sigma^{\text{L}}(t)$ and $\sigma^{\text{R}}(t)$ do exist, then let $T^{\text{RL}}(t)$ and $T^{\text{LR}}(t)$ be the tours that start at the origin at time 0 and then move as follows:

$$T^{\text{RL}}(t) := \text{waituntil}(r^{\text{R}}(t) - |\text{R}(t)|) \oplus \text{move}(\text{R}(t)) \oplus \text{move}(\text{L}(t)) \oplus \text{move}(0),$$
$$T^{\text{LR}}(t) := \text{waituntil}(r^{\text{L}}(t) - |\text{L}(t)|) \oplus \text{move}(\text{L}(t)) \oplus \text{move}(\text{R}(t)) \oplus \text{move}(0).$$

If $\sigma^{\text{L}}(t)$ does not exist, then $T^{\text{RL}}(t) := \text{waituntil}(r^{\text{R}}(t) - |\text{R}(t)|) \oplus \text{move}(\text{R}(t)) \oplus \text{move}(0)$ and $T^{\text{LR}}(t)$ is undefined. Similarly, if $\sigma^{\text{R}}(t)$ does not exist, then $T^{\text{LR}}(t) := \text{waituntil}(r^{\text{L}}(t) - |\text{L}(t)|) \oplus \text{move}(\text{L}(t)) \oplus \text{move}(0)$ and $T^{\text{RL}}(t)$ is undefined. Note that $|T^{\text{RL}}(t)|$ (resp. $|T^{\text{LR}}(t)|$) is a lower bound for the makespan of the shortest tour serving $\sigma^{\text{R}}(t)$ before $\sigma^{\text{L}}(t)$ (resp. $\sigma^{\text{L}}(t)$ before $\sigma^{\text{R}}(t)$). We now describe a worst case situation against which a ρ-competitive algorithm needs to be able to defend itself. Say that at time t, we have $|\text{L}(t)|, |\text{R}(t)| > 0$ and Alg greedily decides to immediately start serving the extremes in the same order as $T^{\text{LR}}(t)$. To see how this can fail, assume that T^{opt} initially follows the tour $T^{\text{RL}}(t)$, but continues to move to the left after serving $\sigma^{\text{L}}(t)$. The time when T^{opt} reaches $\text{L}(t)$ is $t' = r^{\text{R}}(t) + |\text{R}(t)| + |\text{L}(t)|$, since $r^{\text{R}}(t) \geq |\text{R}(t)|$ by assumption. Let t_0 be the time when Alg reaches the origin 0 after serving $\sigma^{\text{L}}(t)$, and assume that $t' \leq t_0$. Now a new request $\sigma' = (p'; t_0)$ may arrive at time t_0 and position $p' = -|\text{L}(t)| - (t_0 - t') = -t_0 + r^{\text{R}}(t) + |\text{R}(t)|$, that the optimum can serve immediately at time t_0. We then have

$$\begin{aligned}|T^{\mathrm{opt}}| &= t_0 + |p'| = t_0 + |\mathrm{L}(t)| + (t_0 - t')\\ &= 2t_0 - r^{\mathrm{R}}(t) - |\mathrm{R}(t)|.\end{aligned}$$

At time t_0 our algorithm still needs to serve σ' and σ^{R}, and hence

$$|T^{\mathrm{alg}}| = t_0 + 2|p'| + 2|\mathrm{R}(t)| = 3t_0 - 2r^{\mathrm{R}}(t).$$

The algorithm Alg is ρ-competitive if

$$\frac{|T^{\mathrm{opt}}|}{|T^{\mathrm{alg}}|} \leq \rho \Leftrightarrow t_0 \geq \frac{\rho|\mathrm{R}(t)| - (2-\rho)r^{\mathrm{R}}(t)}{2\rho - 3}.$$

We denote by SafeReturn($\sigma^{\mathrm{R}}(t), t$) the earliest time a ρ-competitive algorithm may return to the origin after having served $\sigma^{\mathrm{L}}(t)$ while $\sigma^{\mathrm{R}}(t)$ still needs to be served afterwards. The setting described above motivates the definition

$$\textsc{SafeReturn}(\sigma^{\mathrm{R}}(t), t) := \frac{\rho|\mathrm{R}(t)| - (2-\rho)r^{\mathrm{R}}(t)}{2\rho - 3}. \tag{1}$$

The time SafeReturn($\sigma^{\mathrm{L}}(t), t$) is defined symmetrically. Thus, waiting is sometimes necessary for an algorithm to be competitive. However, we can obviously not afford to wait too long. To quantify this, we introduce a lower bound on the length of T^{opt}.

Definition 3.2. If $\sigma^{\mathrm{L}}(t)$ and $\sigma^{\mathrm{R}}(t)$ both do exist, then we define the *greedy tour* $T^{\mathrm{greedy}}(t)$ at time t as

$$T^{\mathrm{greedy}}(t) := \begin{cases} T^{\mathrm{LR}}(t) & \text{if } |T^{\mathrm{LR}}(t)| \leq |T^{\mathrm{RL}}(t)|, \text{ and}\\ T^{\mathrm{RL}}(t) & \text{otherwise.}\end{cases}$$

If $\sigma^{\mathrm{R}}(t)$ does not exist, then we set $T^{\mathrm{greedy}}(t) := T^{\mathrm{LR}}(t)$, and $T^{\mathrm{greedy}}(t) := T^{\mathrm{RL}}(t)$ if $\sigma^{\mathrm{L}}(t)$ does not exist.

Observation 2. *If* $|\mathrm{L}(t)|, |\mathrm{R}(t)| > 0$, *then we have*

$$\begin{aligned}|T^{\mathrm{RL}}(t)| &= r^{\mathrm{R}}(t) + |\mathrm{R}(t)| + 2|\mathrm{L}(t)|,\\ |T^{\mathrm{LR}}(t)| &= r^{\mathrm{L}}(t) + |\mathrm{L}(t)| + 2|\mathrm{R}(t)|,\end{aligned}$$

and $|T^{\mathrm{greedy}}(t)| = \min\{|T^{\mathrm{LR}}(t)|, |T^{\mathrm{RL}}(t)|\} \leq |T^{\mathrm{opt}}|$.

Assume Alg is still waiting at the origin at time t, i.e. $\mathrm{pos}(t) = 0$. From Observation 2, we conclude that if $t \leq \rho|T^{\mathrm{greedy}}(t)| - 2|\mathrm{R}(t)| - 2|\mathrm{L}(t)|$, we can wait until time $\rho|T^{\mathrm{greedy}}(t)| - 2|\mathrm{R}(t)| - 2|\mathrm{L}(t)|$, and then still serve $\sigma^{\mathrm{R}}(t)$ and $\sigma^{\mathrm{L}}(t)$ and return to the origin 0 until time $\rho|T^{\mathrm{greedy}}(t)| \leq \rho|T^{\mathrm{opt}}|$, i.e., we can stay ρ-competitive if no further requests arrive. Formally, we make the following definition.

Definition 3.3. We define $\sigma^{\mathrm{near}}(t) = (\mathrm{near}(t), r^{\mathrm{near}}(t))$ and $\sigma^{\mathrm{far}}(t) = (\mathrm{far}(t), r^{\mathrm{far}}(t))$ to denote the extreme request that is closer to the origin and the extreme request that is further away from the origin at time t, respectively, provided that both extreme requests exist at time t. If only one extreme request does exist at time t, then we define $\sigma^{\mathrm{near}}(t) = (0, t)$, while $\sigma^{\mathrm{far}}(t)$ is defined as before. If both extreme requests have the same distance to the origin, then we set $\sigma^{\mathrm{near}}(t)$ to be the leftmost extreme and $\sigma^{\mathrm{far}}(t)$ to be the rightmost extreme.

Let the *safe tour* $T^{\mathrm{safe}}(t)$ at time t be defined as

$$T^{\mathrm{safe}}(t) := T^{\mathrm{wait}} \oplus \mathrm{waituntil}(r^{\mathrm{far}}(t) - |\mathrm{far}(t)|) \oplus \mathrm{move}(\mathrm{far}(t)) \oplus \mathrm{move}(\mathrm{near}(t)) \oplus \mathrm{move}((0)),$$

with

$$T^{\text{wait}} := \text{waituntil}\Big(\rho|T^{\text{greedy}}(t)| - 2|\text{R}(t)| - 2|\text{L}(t)|\Big).$$

It turns out that $T^{\text{safe}}(t)$ does not return to the origin too early after having served $\sigma^{\text{far}}(t)$, which ensures that the worst case situation described above cannot occur (see Lemma 3.6 below). Thus, it always makes sense for an algorithm to follow the safe tour whenever possible.

Intuitively, we think of a safe release time t as a point in time where a new request is released and the server is not too far away from $T^{\text{safe}}(t)$ and is able to follow it and return to the origin by time $\rho|T^{\text{greedy}}(t)|$. We formalize this in the following observation.

Definition 3.4. Let t be a time at which a new extreme request is released. We say that t is a *safe release time* if

$$t + |\text{pos}(t) - \text{far}(t)| + |\text{far}(t)| + 2|\text{near}(t)| \leq \rho|T^{\text{greedy}}(t)|.$$

Otherwise, we say that t is an *unsafe release time*. Furthermore, we say an interval $[t_1, t_2]$ is *safe* if every release time $t \in [t_1, t_2]$ is safe and *unsafe* if *every* release time in $[t_1, t_2]$ is unsafe.

Note that if t is a safe release time, then, by the definition of the safe tour (Definition 3.3), we have $|T^{\text{alg}}(t)| = \rho|T^{\text{greedy}}(t)| \leq \rho|T^{\text{opt}}|$ for the newly computed tour $T^{\text{alg}}(t)$. Also note that an *interval* may be neither safe nor unsafe.

Before we formally describe our algorithm (cf. Algorithm 1), we need to introduce additional notation for $i \in \{1, 2\}$.

- We define $\sigma^{\textsc{Safe}_i}(t) = (\textsc{Safe}_i(t), r^{\textsc{Safe}_i}(t))$ and $\sigma^{\textsc{Greedy}_i}(t) = (\textsc{Greedy}_i(t), r^{\textsc{Greedy}_i}(t))$ to be the two extreme requests on the tour $T^{\text{safe}}(t)$ and $T^{\text{greedy}}(t)$, respectively, in the order in which they are served.
- We let $\{\sigma_1 = (a_1, t_1), \sigma_2 = (a_2, t_2)\} := \{\sigma^{\text{L}}(t), \sigma^{\text{R}}(t)\}$.
- We define $\textsc{Return}(\sigma_1, \sigma_2, t)$ to be the point in time at which the tour that starts at time t at position $\text{pos}(t)$ and executes $\text{move}(a_1) \oplus \text{move}(a_2) \oplus \text{move}(0)$ returns to the origin. Hence, we have

$$\textsc{Return}(\sigma_1, \sigma_2, t) = t + |a_1(t) - \text{pos}(t)| + |a_1(t)| + 2|a_2(t)|.$$

- Similarly, we define $\textsc{Return}(\sigma_1, t) := \textsc{Return}(\sigma_1, (0, t), t)$, which yields

$$\textsc{Return}(\sigma_1, t) = t + |a_1(t) - \text{pos}(t)| + |a_1(t)|.$$

- We define $p^{\text{L}}(t) := \min_{t' \leq t}\{\text{L}(t')\}$ and $p^{\text{R}}(t) := \max_{t' \leq t}\{\text{R}(t')\}$.

We are now ready to describe our algorithm (cf. Algorithm 1). By Lemma 3.1, it suffices to define the algorithm so that it takes the current time, the current position of the server, and a pair of extreme requests as input and then recomputes the tour of the algorithm.

We argued that it is a safe option to follow $T^{\text{safe}}(t)$ to stay ρ-competitive, provided no further requests appear. It will turn out that it is indeed always good enough to follow $T^{\text{safe}}(t)$, if possible. However, at time t, the server may be too far away from $\text{far}(t)$ to catch up with the safe tour in time, in which case the algorithm has to resort to secondary strategies. If time t is an unsafe release time, then it instead bases its behavior on the greedy tour as an estimate for T^{opt}. Surprisingly, this estimate turns out to be sufficient to obtain a best possible online algorithm.

There are three situations that can occur at time t if t is an unsafe release time. If the online server is on the same side of the origin as $\textsc{Greedy}_1(t)$, i.e., $\text{sign}(\text{pos}(t)) = \text{sign}(\textsc{Greedy}_1(t))$, then our algorithm decides to follow the greedy tour. If $\text{sign}(\text{pos}(t)) \neq \text{sign}(\textsc{Greedy}_1(t))$, then we have to ensure that the algorithm does not return to the origin too early when serving $\textsc{Greedy}_2(t)$ first. If $\textsc{Return}(\sigma^{\textsc{Greedy}_2}(t), t) \geq \textsc{SafeReturn}(\sigma^{\textsc{Greedy}_1}(t), t)$, then the algorithm serves $\textsc{Greedy}_2(t)$ first,

ALGORITHM 1: $\text{Update}(t, \text{pos}(t), \sigma^{\text{L}}(t), \sigma^{\text{R}}(t))$ for the closed online TSP Problem

```
        ▷ This function is called upon release of a new extreme request.
        Input: Current time $t$
                 Current position $\text{pos}(t)$
                 Unserved extreme requests $\sigma^{\text{L}}(t)$ and $\sigma^{\text{R}}(t)$
        Output: A closed TSP tour serving all remaining requests
        $t^{\text{wait}} \leftarrow \rho|T^{\text{greedy}}(t)| - (|\text{pos}(t) - \text{far}(t)| + |\text{far}(t)| + 2|\text{near}(t)|)$
        if $t^{\text{wait}} \geq t$ then                                                    ▷ $t$ is safe
(A)         $T^{\text{alg}} \leftarrow \text{waituntil}(t^{\text{wait}}) \oplus \text{move}(\text{far}(t)) \oplus \text{move}(\text{near}(t)) \oplus \text{move}(0)$
        else if
        $\text{sign}(\text{pos}(t)) = \text{sign}(\text{Greedy}_1(t))$ or $\text{Return}(\text{Greedy}_2(t), t) < \text{SafeReturn}(\text{Greedy}_1(t), t)$
        then
(B1) or (B2)  $T^{\text{alg}} \leftarrow \text{move}(\text{Greedy}_1(t)) \oplus \text{move}(\text{Greedy}_2(t)) \oplus \text{move}(0)$
        else
(C)         $T^{\text{alg}} \leftarrow \text{move}(\text{Greedy}_2(t)) \oplus \text{move}(\text{Greedy}_1(t)) \oplus \text{move}(0)$
        return $T^{\text{alg}}$
```

i.e., it serves the extremes in a different order than $T^{\text{greedy}}(t)$. Otherwise, as we will see, we can deduce from $\text{Return}(\sigma^{\text{Greedy}_2}(t), t) < \text{SafeReturn}(\sigma^{\text{Greedy}_1}(t), t)$ that we can afford to serve the opposite extreme first, i.e., to follow $T^{\text{greedy}}(t)$.

In the following sections, we analyze each of the above cases to obtain the main result of this section:

Theorem 3.5. *Algorithm 1 is a $(9 + \sqrt{17})/8 \approx 1.64$-competitive algorithm for closed online TSP on the line.*

Analysis of Algorithm 1

We will show that Algorithm 1 is ρ-competitive where ρ is the nonnegative root of the polynomial $4x^2 - 9x + 4$.

The analysis of Algorithm 1 relies on multiple lemmas describing the behavior of the server in certain situation. For all of the lemmas stated in the following also the symmetric versions (obtained by swapping left and right) hold. Throughout this section, we refer to the symmetric version of a lemma by $\text{Lemma}_{\text{sym}}$.

The following lemma motivates our choice of ρ.

Lemma 3.6. *The safe tour $T^{\text{safe}}(t)$ arrives at the origin at time t_0 after having served $\sigma^{\text{far}}(t)$ with $t_0 \geq \text{SafeReturn}(\sigma^{\text{near}}(t), t)$.*

Proof. The safe tour $T^{\text{safe}}(t)$ returns to the origin after having served $\sigma^{\text{far}}(t)$ at time

$$\begin{aligned} t_0 = \rho|T^{\text{greedy}}(t)| - 2|\text{near}(t)| &\geq \rho(2|\text{near}(t)| + 2|\text{far}(t)|) - 2|\text{near}(t)| \\ &\geq (4\rho - 2)|\text{near}(t)|. \end{aligned}$$

To deduce that $(4\rho - 2)|\text{near}(t)| \geq \text{SafeReturn}(\sigma^{\text{near}}(t), r)$, note that

$$\frac{(2\rho - 2)|\text{near}(t)|}{2\rho - 3} \geq \frac{\rho|\text{near}(t)| - (2 - \rho)r^{\text{near}}(t)}{2\rho - 3},$$

because $\rho \leq 2$ and $r^{\text{near}}(t) \geq |\text{near}(t)|$. Finally,

$$(4\rho - 2)|\text{near}(t)| \geq \frac{(2\rho - 2)|\text{near}(t)|}{2\rho - 3} \Leftrightarrow (4\rho^2 - 9\rho + 4) \geq 0, \tag{2}$$

which proves the lemma for our choice of ρ. □

Recall that $T^{\text{greedy}}(t)$ denotes the fastest (offline) tour serving the unserved requests at time t. Also recall $|T^{\text{greedy}}(t)| = \min\{|T^{\text{LR}}(t)|, |T^{\text{RL}}(t)|\}$ (Observation 2).

Definition 3.7. We call an interval $[t_1, t_2]$ *stable* if Alg does not serve requests at any time in (t_1, t_2).

To analyze Algorithm 1, we assume that the algorithm planned some tour for an initial sequence of requests. At time t a new extreme request is released and thus an updated tour $T^{\text{alg}}(t)$ is computed. We show that this tour is ρ-competitive. We assume without loss of generality that the extreme request released at time t is a *rightmost extreme* $\sigma^{\text{R}}(t)$. Thus, $r^{\text{R}}(t) = t$.

Recall that if t is a safe release time, then we have $|T^{\text{alg}}(t)| = \rho|T^{\text{greedy}}(t)| \leq \rho|T^{\text{opt}}|$ (Definitions 3.3 and 3.4). Thus, we will in the following assume that t is unsafe.

The next two lemmas restrict the situation that we need to consider.

Lemma 3.8. *If $\sigma^{\text{L}}(t)$ does not exist, then $T^{\text{alg}}(t)$ is ρ-competitive.*

Proof. Lower bounds for the offline optimum are $|T^{\text{opt}}| \geq t + |\text{R}(t)|$ and $|T^{\text{opt}}| \geq 2|p^{\text{R}}(t)| + 2|p^{\text{L}}(t)|$. Because $|\text{L}(t)| = 0$ and $|\text{pos}(t) - \text{R}(t)| \leq |p^{\text{L}}(t)| + |p^{\text{R}}(t)| \leq T^{\text{opt}}/2$, we have

$$|T^{\text{alg}}| = t + |\text{pos}(t) - |\text{R}(t)|| + |\text{R}(t)| \leq 3/2|T^{\text{opt}}| < \rho|T^{\text{opt}}|,$$

and thus the tour is ρ-competitive. □

Lemma 3.9. *Consider an interval $[r^{\text{L}}(t), t]$ with $r^{\text{R}}(t) = t$ and $\text{Greedy}_1(t) = \text{R}(t)$. Then t is a safe release time.*

Proof. First suppose $\text{far}(t) = \text{R}(t)$, i.e., $\text{R}(t) \geq |\text{L}(t)|$. In this case, we need to verify $\text{Return}(\sigma^{\text{R}}(t), t) + 2|\text{L}(t)| \leq \rho|T^{\text{greedy}}(t)|$.

By Lemma 3.1, we have $\text{pos}(t) \in [\text{L}(t), \text{R}(t)]$, hence $|\text{pos}(t) - \text{R}(t)| \leq |\text{L}(t)| + |\text{R}(t)|$ and $t = r^{\text{R}}(t) \geq |\text{R}(t)|$ by assumption. Together with $\rho \geq 1.5$ and $\text{Greedy}_1(t) = \text{R}(t)$, we thus have

$$\begin{aligned}
\text{Return}(\sigma^{\text{R}}(t), t) &= t + |\text{pos}(t) - \text{R}(t)| + |\text{R}(t)| \\
&\leq t + 2|\text{R}(t)| + |\text{L}(t)| \\
&= \rho\Big(t + |\text{R}(t)| + 2|\text{L}(t)|\Big) + |\text{R}(t)| \\
&\quad + (1-\rho)(t + |\text{R}(t)|) + (1-2\rho)|\text{L}(t)| \\
&\overset{\rho \geq 1.5}{\leq} \rho(t + |\text{R}(t)| + 2|\text{L}(t)|) + |\text{R}(t)| - |\text{R}(t)| - 2|\text{L}(t)| \\
&= \rho(t + |\text{R}(t)| + 2|\text{L}(t)|) - 2|\text{L}(t)| \\
&\overset{\text{Obs. 2}}{=} \rho|T^{\text{RL}}(t)| - 2|\text{L}(t)| \\
&= \rho|T^{\text{greedy}}(t)| - 2|\text{L}(t)|.
\end{aligned}$$

Thus, time t is a safe release time.

Next suppose $\text{far}(t) = \text{L}(t)$, i.e., $|\text{L}(t)| \geq \text{R}(t)$. Using Lemma 3.1, we can again deduce that $|\text{pos}(t) - \text{L}(t)| \leq |\text{R}(t)| + |\text{L}(t)| \leq 2|\text{L}(t)|$. Again, we use $\rho \geq 1.5$, $\text{Greedy}_1(T) = \text{R}(t)$ and $t \geq |\text{R}(t)|$

to obtain

$$\begin{aligned}
\textsc{Return}(\sigma^{\mathrm{L}}(t), t) &= t + |\mathrm{pos}(t) - \mathrm{L}(t)| + |\mathrm{L}(t)| \\
&\leq t + 3|\mathrm{L}(t)| \\
&= \rho(t + |\mathrm{R}(t)| + 2|\mathrm{L}(t)|) + (3 - 2\rho)|\mathrm{L}(t)| + (1 - \rho)t - \rho|\mathrm{R}(t)| \\
&\overset{\rho \geq 1.5}{\leq} \rho(t + |\mathrm{R}(t)| + 2|\mathrm{L}(t)|) - 2|\mathrm{R}(t)| \\
&\overset{\text{Obs. 2}}{=} \rho|T^{\mathrm{RL}}(t)| - 2|\mathrm{R}(t)| \\
&= \rho|T^{\mathrm{greedy}}(t)| - 2|\mathrm{R}(t)|.
\end{aligned}$$

Thus, t is again a safe release time. □

The remaining analysis of Algorithm 1 deals with the following case:

$$\text{At time } t \text{ a rightmost extreme is released, } t \text{ is unsafe and } \textsc{Greedy}_1(t) = \mathrm{L}(t). \tag{3}$$

Note that $\textsc{Greedy}_1(t) = \mathrm{L}(t)$ in particular implies that $\sigma^{\mathrm{L}}(t)$ does exist. To further analyze Algorithm 1, we have to take the behavior of the server at time $r^{\mathrm{L}} := r^{\mathrm{L}}(t)$ into account. Recall that $r^{\mathrm{L}}(t)$ is the release time of the leftmost extreme $\sigma^{\mathrm{L}}(t) = \sigma^{\mathrm{L}}(r^{\mathrm{L}})$ at time t. Since the request $\sigma^{\mathrm{L}}(r^{\mathrm{L}}(t))$ is still unserved at time t, no new leftmost extreme is released during $(r^{\mathrm{L}}, t]$ and also no leftmost extreme is served during this interval. We therefore denote $\mathrm{L}(t)$ simply by L from now on.

The analysis of Algorithm 1 is subdivided into several cases.

Case 1: *The interval* $[r^{\mathrm{L}}, t]$ *is stable.*

Case 1.1: *There is a release time* $t^{\mathrm{safe}} \in [r^{\mathrm{L}}, t)$ *that is safe.*
This case is treated in Lemmas 3.13 through 3.15.

Case 1.2: *The interval* $[r^{\mathrm{L}}, t]$ *is unsafe.*
This case is treated in Lemma 3.16.

Case 2: *$\textsc{Alg}$ serves a rightmost extreme during* $[r^{\mathrm{L}}, t)$.
Denote by t^{serve} the latest time at which a rightmost extreme is served during $[r^{\mathrm{L}}(t), t]$. Let $r \in [r^{\mathrm{L}}, t^{\mathrm{serve}})$ be the release time of the last extreme that is released before time t^{serve}. The request $\sigma^{\mathrm{L}}(t)$ is still unserved at time t. Thus, at time r the server starts a tour in the direction of $\sigma^{\mathrm{R}}(r)$ and does not turn around before it is served. Denote by t_0 the point in time at which the tour planned at time r returns to the origin for the first time after having served $\sigma^{\mathrm{R}}(r)$. Lemma 3.17 provides us with a lower bound for t_0, which holds in all but one special case. This Lemma is an integral part to show ρ-competitiveness in Case 2.1. In the following, we will distinguish between the cases in which the lower bound on t_0 holds in general and the case where it does not.

Case 2.1: *We have* $\textsc{Greedy}_1(r) = \mathrm{L}(r)$ *or* $\textsc{Greedy}_1(r) = \mathrm{R}(r)$ *while* $\mathrm{far}(r) = \mathrm{R}(r)$.
This case is treated in Lemmas 3.17 through 3.18.

Case 2.2: *We have* $\textsc{Greedy}_2(r) = \mathrm{R}(r)$ *and* $\mathrm{far}(r) = \mathrm{L}(t')$.
This case is treated in Lemmas 3.20 through 3.24.

Before we start analyzing each of the cases above in detail in the following, we first state a few general lemmas regarding the behavior of $\textsc{Alg}$.

General Lemmas Regarding the Behavior of $\textsc{Alg}$

The general properties of Algorithm 1 implied by the lemmas stated in this section (and also Lemma 3.9) are used repeatedly in the proofs of the lemmas deriving more fine grained properties of the behavior of the algorithm in Cases 1 and 2.

Lemma 3.10. *Let $t = r^{\text{near}}(t)$ be unsafe and $|\text{far}(t)|, |\text{near}(t)| > 0$. Then,*

$$t + |\text{pos}(t)| > (2\rho - 2)(|\text{near}(t)| + |\text{far}(t)|)$$

and

$$2|\text{near}(t)| + |\text{far}(t)| - |\text{pos}(t)| < (\rho - 1)(t + |\text{far}(t)| + 2|\text{near}(t)|).$$

Proof. By assumption t is an unsafe release time. Using $t + |\text{far}(t) - \text{pos}(t)| \le t + |\text{far}(t)| + |\text{pos}(t)|$, this implies

$$\begin{aligned} t + |\text{pos}(t)| &\overset{\text{Def. 3.4}}{>} \rho|T^{\text{greedy}}(t)| - 2|\text{far}(t)| - 2|\text{near}(t)| \\ &\ge \rho(2|\text{far}(t)| + 2|\text{near}(t)|) - 2|\text{far}(t)| - 2|\text{near}(t)| \\ &= (2\rho - 2)(|\text{far}(t)| + |\text{near}(t)|). \end{aligned} \tag{4}$$

Using this together with $4\rho^2 - 5\rho - 2 = 0$ by our choice of ρ and $|\text{far}(t)| \ge |\text{near}(t)|$, we get that

$$\begin{aligned} &2|\text{near}(t)| + |\text{far}(t)| - |\text{pos}(t)| \\ &\quad \le 2|\text{near}(t)| + |\text{far}(t)| - |\text{pos}(t)| + (2-\rho)|\text{pos}(t)| + (4\rho^2 - 5\rho - 2)|\text{near}(t)| \\ &\quad = |\text{far}(t)| + (1-\rho)|\text{pos}(t)| + (2\rho^2 - 3\rho)|\text{near}(t)| + (2\rho^2 - 2\rho)|\text{near}(t)| \\ &\quad \le (2\rho^2 - 3\rho + 1)|\text{far}(t)| + (2\rho^2 - 2\rho)|\text{near}(t)| + (1-\rho)|\text{pos}(t)| \\ &\quad = (\rho - 1)((2\rho - 2)\big(|\text{far}(t)| + |\text{near}(t)|\big) - |\text{pos}(t)| + |\text{far}(t)| + 2|\text{near}(t)|) \\ &\quad \overset{(4)}{<} (\rho - 1)(t + |\text{far}(t)| + 2|\text{near}(t)|). \end{aligned}$$

□

Lemma 3.11. *Let t be a release date of a request such that $\text{Greedy}_1(t) = \text{near}(t)$ and $\text{sign}(\text{pos}(t)) = \text{sign}(\text{far}(t))$. Then, $\text{Alg}_1(t) = \text{far}(t)$ and $\text{Return}(\sigma^{\text{far}}(t)(t), t) \ge \text{SafeReturn}(\sigma^{\text{near}}(t), t)$.*

Proof. If t is a safe release time, then $\text{Alg}_1(t) = \text{far}(t)$ and $\text{Return}(\sigma^{\text{far}}(t)(t), t) = \text{SafeReturn}(\sigma^{\text{near}}(t), t)$ by the definition of the algorithm.

If t is an unsafe release time, then $t > t^{\text{wait}}$ in the notation of the algorithm. That is, if the server serves $\sigma^{\text{far}}(t)$ first, it finishes its tour later than the safe tour. Since $\text{Greedy}_2(t) = \text{far}(t)$ by assumption, Lemma 3.6 implies that Case (B2) does not hold, i.e., $\text{Return}(\sigma^{\text{far}}(t)(t), t) = \text{SafeReturn}(\sigma^{\text{near}}(t), t)$. We also have that the algorithm is not in Case (B1) by assumption. Hence, we indeed have $\text{Alg}_1(t) = \text{far}(t)$. □

Lemma 3.12. *Let $[r^{\text{L}}(t), t]$ be unsafe such that $t = r^{\text{R}}(t)$ and $\text{Greedy}_1(t) = \text{L}$. If there is a time $r \in [r^{\text{L}}, t)$ such that, at time r, the algorithm starts or continues moving to the left, then the server changes its direction at most once during $(r, t]$ before serving any extreme request.*

Proof. Assume that at some time $r' \in (r, t]$ a new rightmost request $\sigma^{\text{R}}(r')$ is released such that $\text{Alg}_1(r') = \text{R}(r')$, i.e., Alg turns around before having served $\sigma^{\text{L}}(t)$. By Lemma$_{\text{sym}}$ 3.9, and since r' is unsafe, $\text{Greedy}_1(r') = \text{L}$. As r' is unsafe and $\text{Alg}_1(r') \ne \text{Greedy}_1(r')$, we have $\text{sign}(\text{pos}(r')) \ne \text{sign}(\text{L})$, so $\text{pos}(r') \ge 0$.

Aiming for a contradiction assume that at time $r'' \in (r', t]$, before Alg can serve $\sigma^{\text{R}}(r')$, a new rightmost request $\sigma^{\text{R}}(t'')$ is released such that $\text{Alg}_1(r'') = \text{L}$. We have $|\text{R}(r'')| > |\text{R}(r')|$, because $\sigma^{\text{R}}(r')$ is still unserved when $\sigma^{\text{R}}(r'')$ is released. Again, we can deduce using Lemma 3.9 that $\text{Greedy}_1(r'') = \text{L}$. Because of $\text{pos}(r') \ge 0$ and the fact that between r' and r'' the server only moves to the right, we obtain $\text{pos}(r'') \ge 0$. It follows with Lemma 3.11 that $\text{far}(r'') = \text{L}$. So $\text{far}(r') = \text{L}$ as well, and we have that $\text{sign}(\text{pos}(r')) \ne \text{sign}(\text{Greedy}_1(r'')) = \text{sign}(\text{L})$, i.e., the algorithm is not in Case (B1) at time r'' (it is also not in Case (A) by assumption). Thus, since we have $\textsc{Alg}_1(r'') = \text{L}$ by assumption, the algorithm is in Case (B2), and thus we have $\textsc{Return}(\sigma^{\text{R}}(r''), r'') < \textsc{SafeReturn}(\sigma^{\text{L}}(t), r'')$.

At time r', we have two possible scenarios, $\text{far}(r') = \text{L}$ and $\text{far}(r') = \text{R}(r')$. We show that both of them lead to a contradiction, which concludes the proof of this lemma. First suppose that $\text{far}(r') = \text{R}(r')$. This immediately implies $|\text{L}| \leq |\text{R}(r')| < |\text{R}(r'')|$, a contradiction, because we assumed that $|\text{L}| \geq |\text{R}(r'')|$. Hence, the algorithm does not turn around again after the release of $\sigma^{\text{R}}(r')$.

More interestingly, suppose that $\text{far}(r') = \text{L}$, i.e., $|\text{L}| \geq |\text{R}(r')|$. By assumption the time r' is unsafe. Also recall that $\text{pos}(r') \geq 0$. Using these facts together with Equation (2) yields

$$\begin{aligned} \textsc{Return}(\sigma^{\text{R}}(r'), r') &= r' + 2|\text{R}(r')| - |\text{pos}(r')| \\ &\overset{\text{Def. 3.4}}{>} \rho|T^{\text{greedy}}(t)| - 2|\text{L}| \\ &\geq (4\rho - 2)|\text{L}| \\ &\overset{(2)}{\geq} \textsc{SafeReturn}(\sigma^{\text{L}}(t), r''). \end{aligned} \tag{5}$$

At time r'' the server directly starts to move to left. Hence, in this case, we have $|r'' - r'| = |\text{pos}(r') - \text{pos}(r'')|$. Because of $\text{pos}(r'') \geq \text{pos}(r') \geq 0$, it also holds that $|\text{pos}(r') - \text{pos}(r'')| = |\text{pos}(r'')| - |\text{pos}(r')|$. This implies $r'' + 2|\text{R}(r')| - |\text{pos}(r'')| = r' + 2|\text{R}(r')| - |\text{pos}(r')|$. Together with $|\text{R}(r'')| > |\text{R}(r')|$, we have

$$r'' + 2|\text{R}(r'')| - |\text{pos}(r'')| > r' + 2|\text{R}(r')| - |\text{pos}(r')| \overset{(5)}{\geq} \textsc{SafeReturn}(\sigma^{\text{L}}(t), r''),$$

contradicting $\textsc{Return}(\sigma^{\text{R}}(r''), r'') < \textsc{SafeReturn}(\sigma^{\text{L}}(t), r'')$. □

Lemmas Regarding Case 1

For the remainder of the section, we will refer to *Setting 1* as the case when all the assumptions in Equation (3) hold at time t and the server is in Case 1.

Setting 1. *Let $[r^{\text{L}}, t]$ be stable such that $t = r^{\text{R}}(t)$, t is unsafe, and $\textsc{Greedy}_1(t) = \text{L}$.*

Case 1.1. In the following two lemmas, we derive certain behavioral properties of Algorithm 1 that we then use to prove that the tour planned in Case 1.1 is ρ-competitive.

Lemma 3.13. *In Setting 1, if a new rightmost request is released at some time $t_1 \in (r^{\text{L}}, t]$ and $\textsc{Greedy}_1(t_1) = \text{L}$, then*

$$\rho|T^{\text{greedy}}(t_1)| = \rho|T^{\text{LR}}(t_1)| > \rho|T^{\text{greedy}}(t_2)| + 2|\text{R}(t_1) - \text{R}(t_2)|$$

for all $t_2 \in [r^{\text{L}}, t_1)$.

Proof. We have

$$|T^{\text{LR}}(t_1)| = |T^{\text{LR}}(t_2)| + 2|\text{R}(t_1) - \text{R}(t_2)|.$$

This implies $\rho|T^{\text{LR}}(t_1)| = \rho|T^{\text{LR}}(t_2)| + 2\rho|\text{R}(t_1) - \text{R}(t_2)|$ and

$$\begin{aligned} \rho|T^{\text{greedy}}(t_1)| &= \rho|T^{\text{LR}}(t_1)| \\ &= \rho|T^{\text{LR}}(t_2)| + 2\rho|\text{R}(t_1) - \text{R}(t_2)| \\ &\geq \rho|T^{\text{greedy}}(t_2)| + 2\rho|\text{R}(t_1) - \text{R}(t_2)| \\ &> \rho|T^{\text{greedy}}(t_2)| + 2|\text{R}(t_1) - \text{R}(t_2)|. \end{aligned}$$ □

Lemma 3.14. *Assume the algorithm is in Setting 1 and that there is at least one safe release time during $[r^{\text{L}}, t)$ (i.e., the algorithm is in Case 1.1). Let $t^{\text{safe}} \in [r^{\text{L}}, t)$ be the* last *safe release time in this interval. Then $\text{pos}(t) \leq 0$, $\text{far}(t^{\text{safe}}) = \text{L}$ and the server does not turn around after time t^{safe} before $\sigma^{\text{L}}(t)$ is served.*

Proof. We will show that all cases except for the one in the lemma lead to a contradiction. Since $[r^{\mathrm{L}}, t]$ is stable, the server does not serve any extremes during $[r^{\mathrm{L}}, t)$. Also, $\sigma^{\mathrm{L}}(t)$ is not served before time t. Thus, a new rightmost request $\sigma^{\mathrm{R}}(t')$ with $|\mathrm{R}(t')| \geq |\mathrm{R}(t^{\mathrm{safe}})|$ is released at some time $t' \in (t^{\mathrm{safe}}, t]$ before Alg serves any extreme. By definition of t^{safe} the release time t' is unsafe. By Lemma 3.9, $\textsc{Greedy}_1(t') = \mathrm{L}$ for all times $r \in (t^{\mathrm{safe}}, t]$ at which a new request is released. Thus, by Lemma 3.13, we have

$$\rho|T^{\mathrm{greedy}}(t')| = \rho|T^{\mathrm{LR}}(t')| > \rho|T^{\mathrm{greedy}}(t^{\mathrm{safe}})| + 2|\mathrm{R}(t') - \mathrm{R}(t^{\mathrm{safe}})|. \tag{6}$$

Aiming for a contradiction assume $\mathrm{far}(t^{\mathrm{safe}}) = \mathrm{R}(t^{\mathrm{safe}})$, i.e., $|\mathrm{R}(t')| \geq |\mathrm{R}(t^{\mathrm{safe}})| \geq |\mathrm{L}|$. The server can be in three situations at time t'. Either Alg is already moving along the safe tour toward $\sigma^{\mathrm{R}}(t^{\mathrm{safe}})$, it is still waiting, or it is moving toward the safe tour to catch up with it. As $(r^{\mathrm{L}}, t]$ is stable, $\sigma^{\mathrm{R}}(t^{\mathrm{safe}})$ is still unserved at time t'. In each of the three situations, Definition 3.4 implies

$$\begin{aligned}\textsc{Return}(\sigma^{\mathrm{R}}(t^{\mathrm{safe}}), \sigma^{\mathrm{L}}(t'), t') &= t' + |\mathrm{pos}(t') - \mathrm{R}(t^{\mathrm{safe}})| + |\mathrm{R}(t^{\mathrm{safe}})| + 2|\mathrm{L}| \\ &\leq \rho|T^{\mathrm{greedy}}(t^{\mathrm{safe}})|.\end{aligned} \tag{7}$$

By using Equations (6), (7), and $|\mathrm{R}(t')| \geq |\mathrm{R}(t^{\mathrm{safe}})|$, we get that

$$\begin{aligned}&\textsc{Return}(\sigma^{\mathrm{R}}(t'), \sigma^{\mathrm{L}}(t), t') \\ &\quad= t' + |\mathrm{pos}(t') - \mathrm{R}(t')| + |\mathrm{R}(t')| + 2|\mathrm{L}| \\ &\quad= t' + |\mathrm{pos}(t') - \mathrm{R}(t^{\mathrm{safe}})| + 2|\mathrm{R}(t') - \mathrm{R}(t^{\mathrm{safe}})| + |\mathrm{R}(t^{\mathrm{safe}})| + 2|\mathrm{L}| \\ &\quad\overset{(7)}{\leq} \rho|T^{\mathrm{greedy}}(t^{\mathrm{safe}})| + 2|\mathrm{R}(t') - |\mathrm{R}(t^{\mathrm{safe}})| \\ &\quad\overset{(6)}{<} \rho|T^{\mathrm{greedy}}(t')|.\end{aligned}$$

Thus, by Definition 3.4 the release time $t' > t^{\mathrm{safe}}$ is safe, a contradiction. Thus, we have $\mathrm{far}(t^{\mathrm{safe}}) = \mathrm{L}$, i.e., $|\mathrm{L}| \geq |\mathrm{R}(t^{\mathrm{safe}})|$.

Next, we show that $|\mathrm{L}| < |\mathrm{R}(t')|$. Again, we aim for a contradiction and assume $|\mathrm{L}| \geq |\mathrm{R}(t')|$. In this case, we also have $\mathrm{far}(t') = \mathrm{L}$. By the same argumentation as above using Equation (6), we get that

$$\begin{aligned}\textsc{Return}(\sigma^{\mathrm{L}}(t), \sigma^{\mathrm{R}}(t'), t') &= t' + |\mathrm{pos}(t') - \mathrm{L}| + |\mathrm{L}| + 2|\mathrm{R}(t')| \\ &= t' + |\mathrm{pos}(t') - \mathrm{L}| + |\mathrm{L}| \\ &\quad + 2|\mathrm{R}(t^{\mathrm{safe}})| + 2|\mathrm{R}(t') - \mathrm{R}(t^{\mathrm{safe}})| \\ &\overset{(6)}{\leq} \rho|T^{\mathrm{greedy}}(t^{\mathrm{safe}})| + 2|\mathrm{R}(t') - \mathrm{R}(t^{\mathrm{safe}})| \\ &< \rho|T^{\mathrm{greedy}}(t')|.\end{aligned}$$

Hence, t' is a safe release time, a contradiction. Thus, we have $|\mathrm{L}| < |\mathrm{R}(t')|$.

First suppose $\mathrm{pos}(t') \geq 0$. Then, since at time t^{safe} the safe tour serves $\sigma^{\mathrm{L}}(t)$ first and $\textsc{Greedy}_1(t^{\mathrm{safe}}) = \mathrm{L}$, we get again as above using Equation (6)

$$\begin{aligned}\textsc{Return}(\sigma^{\mathrm{L}}(t), \sigma^{\mathrm{R}}(t'), t') &= t' + |\mathrm{pos}(t')| + 2|\mathrm{L}| + 2|\mathrm{R}(t')| \\ &= t' + |\mathrm{pos}(t')| + 2|\mathrm{L}| + 2|\mathrm{R}(t^{\mathrm{safe}})| + 2|\mathrm{R}(t') - \mathrm{R}(t^{\mathrm{safe}})| \\ &\overset{(7)}{\leq} \rho|T^{\mathrm{greedy}}(t^{\mathrm{safe}})| + 2|\mathrm{R}(t') - \mathrm{R}(t^{\mathrm{safe}})| \\ &\overset{(6)}{<} \rho|T^{\mathrm{greedy}}(t')|.\end{aligned}$$

Thus, again t' is safe, a contradiction.

It follows that $\mathrm{pos}(t') < 0$, i.e., $\mathrm{sign}(\textsc{Greedy}_1(t')) = \mathrm{sign}(\mathrm{pos}(t'))$. Thus, the algorithm is in Case (B1) at time t', i.e., the server immediately starts a tour first serving $\sigma^{\mathrm{L}}(t)$ at time t'. Whenever at some time $t'' \in (t', t]$ a new request is released, Case (B1) continues to hold and $\textsc{Greedy}_1(t'') = L$. This proves the lemma. □

We are now ready to prove that the tour planned in Case 1.1 is ρ-competitive.

Lemma 3.15. *Let $t = r^{\mathrm{R}}(t)$ be unsafe, $\textsc{Greedy}_1(t) = \mathrm{L}$, assume that there is at least one safe release time during $[r^{\mathrm{L}}, t)$ (Case 1.1) and denote the* last *safe release time in the interval $[r^{\mathrm{L}}, t)$ by t^{safe}. If the interval $[t^{\mathrm{safe}}, t]$ is stable, then the tour planned at time t is ρ-competitive.*

Proof. Lemma 3.14 implies that in this setting, we have $\mathrm{pos}(t) \leq 0$. Using this and the fact that at time t the request $\sigma^{\mathrm{L}}(t)$ is still unserved, we get

$$|T^{\mathrm{alg}}| \leq t + 2|\mathrm{L}| + 2|\mathrm{R}(t)| - |\mathrm{pos}(t)|. \tag{8}$$

Since the server moves to the left without stopping during $[t^{\mathrm{safe}}, t]$, we get $t = t^{\mathrm{safe}} + |\mathrm{pos}(t^{\mathrm{safe}}) - \mathrm{pos}(t)|$. Together with the fact that t^{safe} is a safe release time and $\mathrm{far}(t^{\mathrm{safe}}) = \mathrm{L}$ this yields with Definition 3.4

$$t \leq \rho|T^{\mathrm{greedy}}(t^{\mathrm{safe}})| - 2|\mathrm{L}| - 2|\mathrm{R}(t^{\mathrm{safe}})| + |\mathrm{pos}(t)|. \tag{9}$$

Also note that with $t_1 = t$ and $t_2 = t^{\mathrm{safe}}$ the requirements for Lemma 3.13 are fulfilled, and thus we get

$$\rho|T^{\mathrm{greedy}}(t)| = \rho|T^{\mathrm{LR}}(t)| > \rho|T^{\mathrm{greedy}}(t^{\mathrm{safe}})| + 2|\mathrm{R}(t) - \mathrm{R}(t^{\mathrm{safe}})|. \tag{10}$$

Overall this yields

$$\begin{aligned}
|T^{\mathrm{alg}}| &\overset{(8)}{\leq} t + 2|\mathrm{L}| + 2|\mathrm{R}(t)| - |\mathrm{pos}(t)| \\
&\overset{(9)}{\leq} \rho|T^{\mathrm{greedy}}(t^{\mathrm{safe}})| - 2|\mathrm{L}| - 2|\mathrm{R}(t^{\mathrm{safe}})| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| \\
&= \rho|T^{\mathrm{greedy}}(t^{\mathrm{safe}})| + 2|\mathrm{R}(t) - \mathrm{R}(t^{\mathrm{safe}})| \\
&\overset{(10)}{<} \rho|T^{\mathrm{greedy}}(t)| \leq \rho|T^{\mathrm{opt}}|.
\end{aligned}$$
□

Case 1.2. Using Lemmas 3.9 through 3.12, we can show that the tour planned in Case 1.2 is also ρ-competitive, which finishes our analysis of Case 1.

Lemma 3.16. *Suppose the algorithm is in Setting 1. If the stable interval $[r^{\mathrm{L}}, t]$ is unsafe (Case 1.2), then the tour planned at time t is ρ-competitive.*

Proof. To analyze this case, we have to take the behavior of Alg at time r^{L} into account.

At first note that if at time $t'' \in (r^{\mathrm{L}}, t]$ a request is released, the released request has to be a rightmost extreme, because $\sigma^{\mathrm{L}}(r^{\mathrm{L}})$ is still unserved at time t. By Lemma 3.9, $\textsc{Greedy}_1(t'') = \mathrm{L}$.

Case A: *At time r^{L} the algorithm starts a (unsafe) tour in the direction of $\sigma^{\mathrm{L}}(t)$.*

Case A(i): *The server does* ***not*** *turn around before serving $\sigma^{\mathrm{L}}(t)$.*

Observation 2 implies $|T^{\mathrm{opt}}| \geq r^{\mathrm{L}} + |\mathrm{L}| + 2|\mathrm{R}(t)|$. Together with $|T^{\mathrm{opt}}|/2 \geq |p^{\mathrm{L}}(t)| + |p^{\mathrm{R}}(t)| \geq |\mathrm{pos}(r^{\mathrm{L}}) - \mathrm{L}|$, we get

$$|T^{\mathrm{alg}}| = r^{\mathrm{L}} + |\mathrm{pos}(r^{\mathrm{L}}) - \mathrm{L}| + |\mathrm{L}| + 2|\mathrm{R}(t)| \leq 3/2|T^{\mathrm{opt}}|.$$

Hence, Alg is ρ-competitive.

Case A(ii): *The server does turn around before having served $\sigma^{\mathrm{L}}(t)$.*
Suppose that at some time $t' \in (r^{\mathrm{L}}, t]$ a new rightmost request $\sigma^{\mathrm{R}}(t')$ is released such that Alg turns around before having served $\sigma^{\mathrm{L}}(t)$. By our arguing above, we have $\text{Greedy}_1(t') = \mathrm{L}$. As t' is unsafe by assumption, the algorithm will only start a tour in the direction of $\sigma^{\mathrm{R}}(t')$ at time t' if $\operatorname{sign}(\operatorname{pos}(t')) \neq \operatorname{sign}(\text{Greedy}_1(t'))$, i.e., $\operatorname{pos}(t') \geq 0$ (Case (C)). Lemma 3.12 implies that the server does not turn around again during $[r^{\mathrm{L}}, t]$ before having served an extreme. Hence, using $\operatorname{pos}(r^{\mathrm{L}}) > \operatorname{pos}(t') \geq 0$ and the fact that $\sigma^{\mathrm{R}}(t)$ must be released before $\sigma^{\mathrm{R}}(t')$ is served by assumption, we get that

$$|T^{\mathrm{alg}}| \leq r^{\mathrm{L}} + |\operatorname{pos}(r^{\mathrm{L}})| + 2|\mathrm{R}(t)| + 2|\mathrm{L}|.$$

Again, using $r^{\mathrm{L}} + |\mathrm{L}| + 2|\mathrm{R}(t)| \leq |T^{\mathrm{opt}}|$ and $|\operatorname{pos}(r^{\mathrm{L}})| + |\mathrm{L}| \leq |p^{\mathrm{R}}(t)| + |p^{\mathrm{L}}(t)| \leq |T^{\mathrm{opt}}|/2$, we get

$$|T^{\mathrm{alg}}| \leq 3|T^{\mathrm{opt}}|/2,$$

hence Alg is ρ-competitive.

Case B: *At time r^{L} the server starts a (non-safe) tour in the direction of $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$.*
This can only happen if $|\mathrm{R}(r^{\mathrm{L}})| > 0$. By Lemma 3.9, we have $T^{\mathrm{greedy}}(r^{\mathrm{L}}) = T^{\mathrm{RL}}(r^{\mathrm{L}})$.

Case B(i): *The server does **not** turn around before serving $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$.*
By assumption $\sigma^{\mathrm{R}}(t)$ has to be released before $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$ is served. Thus, using $|T^{\mathrm{opt}}| \geq r^{\mathrm{L}} + |\mathrm{L}| + 2|\mathrm{R}(t)|$, we have

$$\begin{aligned} |T^{\mathrm{alg}}| &= \text{Return}(\sigma^{\mathrm{R}}(t), \sigma^{\mathrm{L}}(r^{\mathrm{L}}), r^{\mathrm{L}}) \\ &\leq r^{\mathrm{L}} + 2|\mathrm{R}(t)| + |\operatorname{pos}(r^{\mathrm{L}})| + 2|\mathrm{L}| \\ &\leq |T^{\mathrm{opt}}| + |\mathrm{L}| + |\operatorname{pos}(r^{\mathrm{L}})|. \end{aligned}$$

Note that if $|\operatorname{pos}(r^{\mathrm{L}})| \leq |\mathrm{R}(t)|$ or $\operatorname{pos}(r^{\mathrm{L}}) \geq 0$, we are done, because then $|\mathrm{L}| + |\operatorname{pos}(r^{\mathrm{L}})| \leq |p^{\mathrm{L}}(t)| + |p^{\mathrm{R}}(t)| \leq |T^{\mathrm{opt}}|/2$ by using $\operatorname{pos}(t) \in [\mathrm{L}, \mathrm{R}(t)]$ (Lemma 3.1).

Next suppose $\operatorname{pos}(r^{\mathrm{L}}) < 0$. Then, $\operatorname{sign}(\operatorname{pos}(r^{\mathrm{L}})) \neq \operatorname{sign}(\text{Greedy}_1(r^{\mathrm{L}}))$. Together with the fact that r^{L} is unsafe this implies that Alg is in Case (C). Now Lemma 3.11 implies that $T^{\mathrm{safe}}(r^{\mathrm{L}})$ serves $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$ first, i.e., $|\mathrm{R}(r^{\mathrm{L}})| > |\mathrm{L}| \geq |\operatorname{pos}(r^{\mathrm{L}})|$. As the server does not serve a rightmost extreme during $[r^{\mathrm{L}}, t)$, we get $|\mathrm{R}(t)| \geq |\mathrm{R}(r^{\mathrm{L}})| \geq |\operatorname{pos}(r^{\mathrm{L}})|$. Thus, we can apply the same reasoning as before to deduce that $|\mathrm{L}| + |\operatorname{pos}(r^{\mathrm{L}})| \leq |T^{\mathrm{opt}}|/2$.

Case B(ii): *The server does turn around before having served $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$.*
We will in the following show that this case cannot occur. Assume that the server turns around before having served $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$. This implies that at some point in time $t' \in (r^{\mathrm{L}}, t]$ before Alg can serve $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$ a new rightmost request with $|\mathrm{R}(t')| > |\mathrm{R}(r^{\mathrm{L}})|$ is released. Recall that as a consequence of Lemma 3.9, we have $\text{Greedy}_1(t') = \mathrm{L}$.

First suppose that $\operatorname{far}(t') = \mathrm{R}(t')$, i.e., $|\mathrm{R}(t')| \geq |\mathrm{L}|$. We then have

$$\begin{aligned} \rho|T^{\mathrm{LR}}(t')| &= \rho(r^{\mathrm{L}} + |\mathrm{L}| + 2|\mathrm{R}(t')|) \\ &\geq r^{\mathrm{L}} + (\rho - 1)|\mathrm{L}| + \rho|\mathrm{L}| + 2\rho|\mathrm{R}(t')| \\ &= r^{\mathrm{L}} + 2\rho(|\mathrm{L}| + |\mathrm{R}(t')|) - |\mathrm{L}|. \end{aligned}$$

In the first inequality, we used that $r^{\mathrm{L}} \geq |\mathrm{L}|$. Using $\rho > 1.5$, $|\mathrm{R}(t')| \geq |\mathrm{L}|$ and $\mathrm{pos}(r^{\mathrm{L}}) \in [\mathrm{L}, \mathrm{R}(r^{\mathrm{L}})]$, this implies that

$$\begin{aligned}
\rho|T^{\text{greedy}}(t')| - 2|\mathrm{L}| - |\mathrm{R}(t')| & \\
&= \rho|T^{\mathrm{LR}}(t')| - 2|\mathrm{L}| - |\mathrm{R}(t')| \\
&\geq r^{\mathrm{L}} + (2\rho - 3)\Big(|\mathrm{L}| + |\mathrm{R}(t')|\Big) + 2|\mathrm{R}(t')| \\
&\geq r^{\mathrm{L}} + (2\rho - 3)\Big(|\mathrm{L}| + |\mathrm{R}(t')|\Big) + |\mathrm{L}| + |\mathrm{R}(t')| \\
&> r^{\mathrm{L}} + |\mathrm{L}| + |\mathrm{R}(t')| \\
&\geq r^{\mathrm{L}} + |\mathrm{pos}(r^{\mathrm{L}}) - \mathrm{R}(t')| \\
&= t' + |\mathrm{pos}(t') - \mathrm{R}(t')|.
\end{aligned}$$

Hence, the release time t' is safe, contradicting our assumption that $[r^{\mathrm{L}}, t]$ is unsafe.

Next suppose that $\mathrm{far}(t') = \mathrm{L}$. Then $|\mathrm{R}(r^{\mathrm{L}})| < |\mathrm{R}(t')| < |\mathrm{L}|$. If $\mathrm{sign}(\mathrm{pos}(r^{\mathrm{L}})) = \mathrm{sign}(\mathrm{far}(r^{\mathrm{L}}))$, then Lemma 3.11 implies $\textsc{Alg}_1(r^{\mathrm{L}}) = \mathrm{far}(r^{\mathrm{L}}) = \mathrm{L}$. Thus, we have $\mathrm{pos}(r^{\mathrm{L}}) \geq 0$.

By assumption r^{L} is unsafe, and we have $|\mathrm{L}| > |\mathrm{R}(r^{\mathrm{L}})|$. Thus, at time r^{L}, all conditions for Lemma 3.10 are fulfilled, and we get that

$$(\rho - 1)(r^{\mathrm{L}} + |\mathrm{L}| + 2|\mathrm{R}(r^{\mathrm{L}})|) > 2|\mathrm{R}(r^{\mathrm{L}})| + |\mathrm{L}| - |\mathrm{pos}(r^{\mathrm{L}})|. \tag{11}$$

At time t' the server turns around and can reach the origin before time $r^{\mathrm{R}} + 2|\mathrm{R}(r^{\mathrm{L}})| - |\mathrm{pos}(r^{\mathrm{L}})|$, because at time t' the request $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$ is still unserved. Thus, $r^{\mathrm{L}} + |\mathrm{pos}(r^{\mathrm{L}})| \leq r^{\mathrm{R}} + 2|\mathrm{R}(r^{\mathrm{L}})| - |\mathrm{pos}(r^{\mathrm{L}})|$. Using this, we get that

$$\begin{aligned}
&\rho|T^{\text{greedy}}(t')| - 2|\mathrm{L}| - 2|\mathrm{R}(t')| \\
&= \rho|T^{\mathrm{LR}}(t')| - 2|\mathrm{L}| - 2|\mathrm{R}(t')| \\
&= \rho(r^{\mathrm{L}} + |\mathrm{L}| + 2|\mathrm{R}(t')|) - 2|\mathrm{L}| - 2|\mathrm{R}(t')| \\
&\overset{(11)}{>} (2|\mathrm{R}(r^{\mathrm{L}})| + |\mathrm{L}| - |\mathrm{pos}(r^{\mathrm{L}})|) + r^{\mathrm{L}} + |\mathrm{L}| + 2|\mathrm{R}(r^{\mathrm{L}})| \\
&\quad - 2|\mathrm{L}| - 2|\mathrm{R}(r^{\mathrm{L}})| \\
&= r^{\mathrm{L}} + 2|\mathrm{R}(r^{\mathrm{L}})| - |\mathrm{pos}(r^{\mathrm{L}})| \\
&\geq t' + |\mathrm{pos}(t')|.
\end{aligned}$$

Thus, the release time t' is safe, contradicting our assumption. □

Lemmas Regarding Case 2

For the remainder of the section, we will refer to *Setting 2* as the case when all the assumptions in Equation (3) hold at time t and the server is in Case 2.

SETTING 2. *Let $[r^{\mathrm{L}}, t]$ be unstable such that $t = r^{\mathrm{R}}(t)$ is unsafe, $|\mathrm{L}| > 0$, and $\textsc{Greedy}_1(t) = \mathrm{L}$. Let t^{serve} be the last time during $[r^{\mathrm{L}}(t), t]$ at which a rightmost extreme is served. Let r be the last release time with $r < t^{\text{serve}}$.*

Case 2.1. The following lemma is central for proving that the tour planned by the algorithm in Case 2.1 is ρ-competitive.

Lemma 3.17. *Assume the algorithm is in Setting 2. If* $\textsc{Greedy}_1(r) = \mathrm{L}$*, or* $\textsc{Greedy}_1(r) = \mathrm{R}(r) = \mathrm{far}(r)$*, then*

$$t^{\mathrm{serve}} + \mathrm{pos}(t^{\mathrm{serve}}) \geq \textit{SafeReturn}(\sigma^{\mathrm{L}}(t), t).$$

Proof. Note that at time $t^{\mathrm{serve}} + \mathrm{pos}(t^{\mathrm{serve}})$ the server returns to the origin for the first time after having served the rightmost extreme released at time r.

First suppose $\textsc{Greedy}_1(r) = \mathrm{L}$. In this case the server starts a tour in the direction of the rightmost extreme if $\mathrm{far}(r) = \mathrm{R}(r)$ and r is safe (Case (A)), or the algorithm is in Case (C). In both cases, we have

$$t^{\mathrm{serve}} + \mathrm{pos}(t^{\mathrm{serve}}) \geq \textsc{SafeReturn}(\sigma^{\mathrm{L}}(t), t). \tag{12}$$

Next suppose $\textsc{Greedy}_1(r) = \mathrm{R}(r) = \mathrm{far}(r)$, i.e., $|\mathrm{R}(r)| \geq |\mathrm{L}|$. Since $T^{\mathrm{alg}}(r)$ also serves $\sigma^{\mathrm{R}}(r)$ first, the server is either on the safe tour (Case (A)) or it follows the safe tour without being able to catch up with it (Case (B1) and Case (B2)). We are done by Lemma 3.6. □

With the next lemma the analysis of Case 2.1 is complete.

Lemma 3.18. *Assume the algorithm is in Setting 2. If at time r, we have* $\textsc{Greedy}_1(r) = \mathrm{L}$*, or* $\textsc{Greedy}_1(r) = \mathrm{R}(r) = \mathrm{far}(r)$*, then the tour planned at time t is ρ-competitive.*

Proof. By Lemma 3.17, we know that

$$t^{\mathrm{serve}} + \mathrm{pos}(t^{\mathrm{serve}}) \geq \textsc{SafeReturn}(\sigma^{\mathrm{L}}(t), t). \tag{13}$$

Note that we also have $t^{\mathrm{serve}} + \mathrm{pos}(t^{\mathrm{serve}}) \leq t + |\mathrm{pos}(t)|$. By Lemma 3.9, $\textsc{Greedy}_1(r) = \mathrm{L}$. We next prove that the following inequality holds if after time t no new request is released:

$$|T^{\mathrm{alg}}| \leq t + 2|\mathrm{L}| + 2|\mathrm{R}(t)| - |\mathrm{pos}(t)|. \tag{14}$$

To show this, we make a case distinction regarding $\mathrm{pos}(t)$.

At first suppose $\mathrm{pos}(t) \leq 0$. In this case the algorithm is in Case (B1), because $\mathrm{sign}(\mathrm{pos}(t)) = \mathrm{sign}(\textsc{Greedy}_1(t)) = \mathrm{sign}(\mathrm{L})$, which means that at time t the server immediately starts a tour in the direction or $\sigma^{\mathrm{L}}(t)$. Since $\mathrm{pos}(t) \leq 0$ by assumption, this implies that Equation (14) holds.

Next suppose $\mathrm{pos}(t) > 0$. Then the algorithm is not in Case (B1) and $\mathrm{sign}(\mathrm{pos}(t)) \neq \mathrm{sign}(\textsc{Greedy}_1(t))$. If $\mathrm{near}(t) = \textsc{Greedy}_1(t)$, then $\mathrm{sign}(\mathrm{pos}(t)) = \mathrm{sign}(\mathrm{far}(t))$ and $\textsc{Alg}_1(t) = \textsc{Greedy}_2(t) = \mathrm{R}(t)$ by Lemma 3.11. If $\mathrm{far}(t) = \textsc{Greedy}_1(t)$, then we obtain

$$t + 2|\mathrm{R}(t)| - |\mathrm{pos}(t)| \geq t + |\mathrm{pos}(t)| \geq t^{\mathrm{serve}} + \mathrm{pos}(t^{\mathrm{serve}}) \geq \textsc{SafeReturn}(\sigma^{\mathrm{L}}(t), t),$$

using that $\mathrm{pos}(t) \in [\mathrm{L}, \mathrm{R}(t)]$ by Lemma 3.1. Thus, the algorithm is in Case (C) at time t. Hence, Equation (14) also holds in this case.

With $t + |\mathrm{R}(t)| \leq |T^{\mathrm{opt}}|$, it thus suffices to prove that

$$2|\mathrm{L}| + |\mathrm{R}(t)| - |\mathrm{pos}(t)| \leq (\rho - 1)|T^{\mathrm{opt}}| \tag{15}$$

to show ρ-competitiveness in this case, but this follows with Lemma 3.10. □

Case 2.2. The next three lemmas are needed to prove that the tour planned in Case 2.2 is ρ-competitive.

Lemma 3.19. *Assume the algorithm is in Setting 2. If at time r, we have* $\textsc{Greedy}_1(r) = \mathrm{L}$*, or* $\textsc{Greedy}_1(r) = \mathrm{R}(r) = \mathrm{far}(r)$*, then*

$$2|\mathrm{L}| + |\mathrm{R}(t)| - |\mathrm{pos}(t)| \leq (\rho - 1)|T^{\mathrm{opt}}|.$$

Proof. Lemma 3.17 implies

$$t^{\text{serve}} + \text{pos}(t^{\text{serve}}) \geq \frac{\rho|\text{L}| - (2-\rho)r^{\text{L}}}{2\rho - 3}. \tag{16}$$

We now make a case distinction on the value $|\text{R}(t)|$.

First suppose

$$|\text{R}(t)| \leq \frac{3-\rho}{2\rho-3}|\text{L}| + \frac{1-\rho}{2\rho-3}r^{\text{L}} + |\text{pos}(t)|. \tag{17}$$

L Using $1.640 \leq \rho \leq 2$ and Inequality Equation (16), we get

$$\begin{aligned} 2|\text{L}| + |\text{R}(t)| - |\text{pos}(t)| &\overset{(17)}{\leq} (2-\rho)\left(\frac{3-\rho}{2\rho-3}|\text{L}| + \frac{1-\rho}{2\rho-3}r^{\text{L}} + |\text{pos}(t)| - |\text{R}(t)|\right) \\ &\quad + 2|\text{L}| + |\text{R}(t)| - |\text{pos}(t)| \\ &= (\rho-1)\left(\frac{\rho|\text{L}| - (2-\rho)r^{\text{L}}}{2\rho-3} - |\text{pos}(t)| + |\text{R}(t)|\right) \\ &\overset{(16)}{\leq} (\rho-1)(t_0 - |\text{pos}(t)| + |\text{R}(t)|) \\ &\leq (\rho-1)(t + |\text{R}(t)|) \\ &\leq (\rho-1)|T^{\text{opt}}|. \end{aligned}$$

Next, suppose

$$|\text{R}(t)| > \frac{3-\rho}{2\rho-3}|\text{L}| + \frac{1-\rho}{2\rho-3}r^{\text{L}} + |\text{pos}(t)|. \tag{18}$$

We obtain

$$\begin{aligned} 2|\text{L}| + |\text{R}(t)| - |\text{pos}(t)| =& ((3-\rho)|\text{L}| + (1-\rho)r^{\text{L}}) - (2\rho-3)|\text{R}(t)| \\ & - |\text{pos}(t)| + (\rho-1)(r^{\text{L}} + |\text{L}| + 2|\text{R}(t)|) \\ \overset{(18)}{<}& -(2\rho-3)|\text{pos}(t)| - |\text{pos}(t)| + (\rho-1)(r^{\text{L}} + |\text{L}| + 2|\text{R}(t)|) \\ \leq& (\rho-1)(r^{\text{L}} + |\text{L}| + 2|\text{R}(t)|) \\ \leq& (\rho-1)|T^{\text{opt}}|. \end{aligned}$$

□

Lemma 3.20. *Assume the algorithm is in Setting 2 and that* Greedy$_1(r) = \text{R}(r)$ *and* $\text{far}(r) = \text{L}$. *Then,* $r = r^{\text{L}}$, $\text{pos}(r^{\text{L}}) \geq 0$ *and* r^{L} *is unsafe. Moreover, if there is a safe release time in* $(r^{\text{L}}, t]$, *then the tour planned at time* t *is* ρ*-competitive and if* $(r^{\text{L}}, t]$ *is unsafe, then* Greedy$_1(t') = \text{L}$ *for every release time in* $(r^{\text{L}}, t]$.

Proof. By the definition of r, we have Alg$_1(r) = \text{R}(r) \neq \text{far}(r)$. Thus, r is unsafe. Therefore, and because Greedy$_1(r) = \text{R}(r)$, we have $r = r^{\text{L}}$ by Lemma 3.9 (since if $r > r^{\text{L}}$, a rightmost extreme would be released at time r). Also, by Lemma 3.11, we have $\text{pos}(r) \geq 0$, since Alg$_1(r) = \text{R}(r)$. Thus, $\text{pos}(r) \geq 0$.

To show $r = r^{\text{L}}$, we again assume to the contrary that $r > r^{\text{L}}$. Then a new rightmost extreme is released at time r. Now Lemma 3.9 implies that r is safe, a contradiction. Hence, $r = r^{\text{L}}$.

If $(r^{\text{L}}, t]$ is unsafe, then Greedy$_1(t') = L$ by Lemma 3.9. If there is a safe release time in $(r^{\text{L}}, t]$, then denote by $t^{\text{safe}} \in (r^{\text{L}}, t)$ the latest safe release time in this interval.

First suppose that during $[t^{\text{safe}}, t]$ no rightmost extreme is served. Note that then the algorithm is in Setting 1 except for the assumption that during $[r^{\text{L}}, t]$ no rightmost extreme is served. With Lemma 3.15 this implies that we can apply Lemma 3.15, which yields that the tour planned at time t is ρ-competitive.

Next assume that after time $t^{\text{safe}} > r^{\text{L}}$ a rightmost extreme is served. The served rightmost extreme is released at some time $\tilde{t} \in [t^{\text{safe}}, r^{\text{L}}]$, in particular $\tilde{t} > r^{\text{L}}$. This means that at time $\tilde{t}$, we have $\textsc{Greedy}_1(\tilde{t}) = \text{L}(\tilde{t})$, or $\textsc{Greedy}_1(\tilde{t}) = \text{R}(\tilde{t})$ and the safe tour at time $\tilde{t}$ serves $\sigma^{\text{L}}(\tilde{t})$ first, because the other remaining case can only occur at time $\tilde{t} = r^{\text{L}}$ by our argument above. Lemma 3.18 now implies that the tour planned at time t is ρ-competitive. □

Lemma 3.21. *Let $[r^{\text{R}}(t), t]$ with $t = r^{\text{L}}(t)$ be unsafe and set $r^{\text{R}} = r^{\text{R}}(t)$ and $r^{\text{L}} = r^{\text{L}}(t)$. Assume that the following assumptions hold:*

- *The interval $[r^{\text{R}}, r^{\text{L}}]$ is stable and safe.*
- *We have $\text{pos}(r^{\text{L}}) \geq 0$.*
- *The server moves to the left during $[r^{\text{R}}, r^{\text{L}}]$.*

Then there exists a time $r \in [r^{\text{R}}, r^{\text{L}}]$ at which the server turns to the right after moving to the left immediately before. Assume that r is the last time with this property in $[r^{\text{R}}, r^{\text{L}}]$.

Then the server only moves to the left during $[r^{\text{R}}, r]$ and to the right during $[r, r^{\text{L}}]$, and we have that $\text{sign}(\text{pos}(r^{\text{R}})) = \text{sign}(\text{pos}(r))$.

Proof. Since $[r^{\text{R}}, r^{\text{L}}]$ is unsafe, Lemma 3.9 implies that

$$\textsc{Greedy}_1(r) = \text{R}(r) \text{ for all } t' \in (r^{\text{R}}, r^{\text{L}}] \text{ at which a leftmost extreme is released.} \tag{19}$$

Note that at time r^{L} the server starts a tour in the direction of $\sigma^{\text{R}}(r^{\text{L}})$ because of $\text{sign}(\textsc{Greedy}_1(r^{\text{L}})) = \text{sign}(\text{pos}(r^{\text{L}}))$. Together with the assumption that the server moves to the left at some point during $[r^{\text{R}}, r^{\text{L}}]$ this implies that there exits a time $r \in [r^{\text{R}}, r^{\text{L}}]$ at which the server turns to the right after moving to the left immediately before.

To show the remaining statements of the lemma, we have to take the tour of the server immediately before time r into account. By our assumptions the interval $[r^{\text{R}}, r^{\text{L}}]$ is unsafe. Thus, immediately before time r the server was on a non-safe tour first serving the leftmost extreme, because $\sigma^{\text{R}}(r^{\text{L}})$ is not served during $[r^{\text{R}}, r^{\text{L}})$ and the server moves to the left immediately before time r. Assume r' is the release time of the last request released strictly before time r. By our arguing above, at time r' the server starts a tour first moving to the left. This in particular means $|\text{L}(r')| > 0$. We make a case distinction regarding the position of the server at time r.

First suppose $\text{pos}(r) \geq 0$. This implies $\text{pos}(r') \geq 0$ as the server by definition only moves to the left during $[r', r]$. The fact that $\textsc{Alg}_1(r') = \text{L}(r')$ implies that $\textsc{Greedy}_1(r') = \text{L}(r')$, because otherwise the algorithm would start a tour first serving $\sigma^{\text{R}}(r')$ at time r', which yields a contradiction. By Equation (19), we know that $\textsc{Greedy}_1(r') = \text{R}(r')$ if $r' > r^{\text{R}}$. Hence, we have $r' = r^{\text{R}}$, which implies that during $[r^{\text{R}}, r]$ the server only moves to the left, and it also holds $\text{sign}(\text{pos}(r)) = \text{sign}(\text{pos}(r^{\text{R}}))$, which means that the assertions of the lemma are fulfilled.

Next suppose $\text{pos}(r) < 0$. This in particular means that $r \neq r^{\text{L}}$, because $\text{pos}(r^{\text{L}}) \geq 0$ by the assumptions of the lemma. We have $r \in (r^{\text{R}}, r^{\text{L}}]$. Hence, we know by Equation (19) that $\textsc{Greedy}_1(r) = \text{R}(r)$. Thus, since the algorithms starts a tour in the direction of $\text{R}(r^{\text{L}})$ at time r, the algorithm is in Case (B2) at time r. Thus,

$$\textsc{Return}(\sigma^{\text{L}}(r), r) = r + 2|\text{L}(r)| - |\text{pos}(r)| < \textsc{SafeReturn}(\sigma^{\text{R}}(r), r). \tag{20}$$

The fact that the server starts a tour first serving $\text{R}(r)$ at time r also implies $\text{far}(r) = \text{R}(r)$ by Lemma 3.11. Using this and the fact that at time r' the algorithm is going in the direction of a leftmost extreme, there are only two cases that can occur at time r'.

First suppose $\textsc{Greedy}_1(r') = \text{L}(r')$. This in particular implies that $r' = r^{\text{R}}$ by Equation (19). At time r, a new leftmost extreme is released and, as no leftmost extremes are served during $[r^{\text{R}}, r^{\text{L}}]$, we have $|\text{L}(r)| > |\text{L}(r^{\text{R}})|$. As the safe tour at time r serves $\sigma^{\text{R}}(r)$, we also get $|\text{R}(r^{\text{L}})| > |\text{L}(r^{\text{R}})|$, i.e., $\mathrm{far}(r^{\mathrm{R}}) = \mathrm{R}(r^{\mathrm{R}})$. This implies $\mathrm{pos}(r^{\mathrm{R}}) < 0$ by Lemma 3.11. Thus, we have that $\mathrm{sign}(\mathrm{pos}(r)) = \mathrm{sign}(\mathrm{pos}(r^{\mathrm{R}}))$, and by definition of $r' = r^{\mathrm{R}}$ the server only moves to the left during $[r^{\mathrm{R}}, r]$. Hence, again the assertions of the lemma are fulfilled.

Next suppose $\textsc{Greedy}_1(r') = \mathrm{R}(r')$. As $|\mathrm{L}(r')| > 0$ by assumption, we can use Lemma 3.9 to deduce that $r' > r^{\mathrm{R}}$, as otherwise the server could reach the safe tour at time r^{R}. At time r' the algorithm starts a tour toward the leftmost extreme. As $\textsc{Greedy}_1(r'') = \mathrm{R}(r')$, the algorithm is in Case (C) at time r'. Thus, we have $\mathrm{sign}(\mathrm{pos}(r')) \neq \mathrm{sign}(\textsc{Greedy}_1(r')) = \mathrm{sign}(\mathrm{R}(r'))$, i.e., $\mathrm{pos}(r') < 0$ and

$$\textsc{Return}(\sigma^{\mathrm{L}}(r'), r') = r' + 2|\mathrm{L}(r')| - |\mathrm{pos}(r')| \geq \textsc{SafeReturn}(\sigma^{\mathrm{R}}(r), r). \tag{21}$$

We deduced above that $r' > r^{\mathrm{R}}$. Hence, at time r' a leftmost extreme with $|\mathrm{L}(r')| < |\mathrm{L}(r)|$ is released, because no leftmost extreme is served during $[r^{\mathrm{L}}, r^{\mathrm{R}}]$. Between r' and r, the server constantly moves to the left. Thus, we have $(r - r') = |\mathrm{pos}(r') - \mathrm{pos}(r)|$. Since, $\mathrm{pos}(r) \leq \mathrm{pos}(r') < 0$, we have that $|\mathrm{pos}(r') - \mathrm{pos}(r)| = |\mathrm{pos}(r)| - |\mathrm{pos}(r')|$. Thus, we have $r + 2|\mathrm{L}(r')| - |\mathrm{pos}(r)| = r' + 2|\mathrm{L}(r')| - |\mathrm{pos}(r')|$. Using this and $|\mathrm{L}(r)| > |\mathrm{L}(r')|$, we get

$$r + 2|\mathrm{L}(r)| - |\mathrm{pos}(r)| \quad > \quad r' + 2|\mathrm{L}(r')| - |\mathrm{pos}(r')| \overset{(21)}{\geq} \textsc{SafeReturn}(\sigma^{\mathrm{R}}(r), r),$$

contradicting Inequality Equation (20). □

The next two lemmas are important building blocks for proving that the tour planned in Case 2.2 is ρ-competitive.

Lemma 3.22. *Assume the algorithm is in Setting 2 and that* $\textsc{Greedy}_1(r) = \mathrm{R}(r)$ *and* $\mathrm{far}(r) = \mathrm{L}$. *Then,*

$$|T^{\mathrm{alg}}| \leq r^{\mathrm{L}} + 2|\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| - |\mathrm{pos}(r^{\mathrm{L}})|. \tag{22}$$

Proof. Lemma 3.20 yields $r = r^{\mathrm{L}}$ and $\mathrm{pos}(r^{\mathrm{L}}) \geq 0$. If $[r^{\mathrm{L}}, t)$ contains a safe release time, then Lemma 3.20 implies that the tour computed at time t is ρ-competitive. Hence, we only need to consider the case where $[r^{\mathrm{L}}, t)$ is unsafe. In this case Lemma 3.20 implies

$$\textsc{Greedy}_1(\hat{t}) = \mathrm{L}, \text{ for all } \hat{t} \in (r^{\mathrm{L}}, t] \text{ at which a rightmost extreme appears.}$$

Recall that at time r^{L} the server starts a tour in the direction of $\mathrm{R}(r^{\mathrm{L}})$ and that during $(r^{\mathrm{L}}, t]$ only rightmost extremes are released. Our claim for $|T^{\mathrm{alg}}|$ follows immediately in two situations: The first situation occurs if between r^{L} and t there are no new requests released; the other situation occurs if all rightmost requests that are released after $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$ was served force the server onto a non-safe tour in the direction of $\sigma^{\mathrm{L}}(t)$. We will in the following analyze the behavior of the algorithm if at a time $r' \leq t$ after $\textsc{Alg}$ has served $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$ a new rightmost request is released such that the tour planned at time r' is non-safe with $\textsc{Alg}_1(r') = \sigma^{\mathrm{R}}(r')$. Also assume that r' is the first time at which the server moves toward a rightmost extreme after having served $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$.

Note, that the server does not serve $\sigma^{\mathrm{R}}(r')$ during $[r', t]$ as r^{L} is the release time of the last rightmost extreme that is served during $[r^{\mathrm{L}}, t]$. Furthermore, Lemma 3.12 implies that the server does turn around again before a request is served. We also know that at time r' the greedy tour serves $\sigma^{\mathrm{L}}(r')$ first. Thus, if $\mathrm{pos}(r') < 0$, then we have $\mathrm{sign}(\mathrm{pos}(r')) = \mathrm{sign}(\textsc{Greedy}_1(r'))$ and the server always starts a tour in the direction of $\sigma^{\mathrm{L}}(t)$ at time r' (because r' is unsafe) (Case (B1)). This contradicts the definition of r', which implies that $\mathrm{pos}(r') \geq 0$. Also note that between times

$\sigma^R(r^L)$ and r' the server does not leave $[\mathrm{pos}(r'), \mathrm{R}(r^L)]$. Overall, this yields

$$\begin{aligned}|T^{\mathrm{alg}}| \leq & r^L + |\mathrm{R}(r^L)| - |\mathrm{pos}(r^L)| + |\mathrm{R}(r^L) - \mathrm{pos}(r')| \\ & - |\mathrm{pos}(r')| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| \\ \leq & r^L + |\mathrm{R}(r^L)| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| - |\mathrm{pos}(r^L)|,\end{aligned}$$

as claimed. □

Lemma 3.23. *Assume the algorithm is in Setting 2 and that with* $\textsc{Greedy}_1(r) = \mathrm{R}(r)$ *and* $\mathrm{far}(r) = \mathrm{L}$*. Further assume that in the optimal offline solution* $\sigma^R(r)$ *is served before* $\sigma^L(t)$ *while* $\sigma^R(t)$ *is served after* $\sigma^L(t)$ *and that* $|\mathrm{R}(r)| > |\mathrm{R}(t)|$*. Then the tour planned at time* t *is* ρ*-competitive.*

Proof. Lemma 3.20 yields $r = r^L$ and $\mathrm{pos}(r^L) \geq 0$. If $[r^L, t)$ contains a safe release time, then Lemma 3.20 implies that the tour computed at time t is ρ-competitive. Hence, we only need to consider the case where $[r^L, t)$ is unsafe. In this case, Lemma 3.20 implies

$$\textsc{Greedy}_1(\hat{t}) = \mathrm{L}, \text{ for all } \hat{t} \in (r^L, t] \text{ at which a rightmost extreme appears.}$$

Define $r^R := r^R(r^L)$ to be the release time of $\sigma^R(r^L)$. Our assumption on the offline optimum solution implies

$$|T^{\mathrm{opt}}| \geq r^R + |\mathrm{R}(r^L)| + 2|\mathrm{L}| + 2|\mathrm{R}(t)|. \tag{23}$$

To analyze this case, we have to regard the behavior of Alg before $\sigma^L(t)$ is released, i.e., we are interested in the time interval between r^R and r^L. By definition, no new rightmost request appears during $(r^R, r^L]$. We again distinguish several cases:

Case A: *During* $[r^R, r^L]$ *no leftmost extreme is served.*

Case A(i): *During* $[r^R, r^L)$ *there is at least one safe release time.*

Recall that by assumption r^L is unsafe. Let $t^{\mathrm{safe}} \in [r^R, r^L)$ be the last safe release time in this interval. Recall that at the unsafe release time r^L, we have $|\mathrm{R}(r^L)| > 0$ and $\textsc{Greedy}_1(r^L) = \mathrm{R}(r^L)$ by assumption. Also during $[r^R, r^L]$ is stable. Thus, by Lemma$_{\mathrm{sym}}$ 3.14, we have that $\mathrm{pos}(r^L) \geq 0$, $\mathrm{far}(t^{\mathrm{safe}}) = \mathrm{R}(r^L)$ and the server does not turn around after t^{safe} before $\sigma^R(r^L)$ is served. Using the fact t^{safe} is safe and that the server only moves to the right during $[t^{\mathrm{safe}}, r^L]$, we get with Definition 3.4,

$$r^L \leq \rho|T^{\mathrm{greedy}}(t^{\mathrm{safe}})| - 2|\mathrm{L}(t^{\mathrm{safe}})| - 2|\mathrm{R}(t^{\mathrm{safe}})| + |\mathrm{pos}(r^L)|. \tag{24}$$

Also note that the requirements for Lemma$_{\mathrm{sym}}$ 3.13 are fulfilled, and we have that

$$\rho|T^{\mathrm{greedy}}(r^L)| = \rho|T^{\mathrm{RL}}(r^L)| > \rho|T^{\mathrm{greedy}}(t^{\mathrm{safe}})| + 2|\mathrm{L} - \mathrm{L}(t^{\mathrm{safe}})|. \tag{25}$$

Recall that at time r^L the greedy tour serves $\sigma^R(r^L)$ first, which implies

$$|T^{\mathrm{opt}}| \overset{(23)}{\geq} r^R + |\mathrm{R}(r^L)| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| = T^{\mathrm{greedy}}(r^L) + 2|\mathrm{R}(t)|. \tag{26}$$

Thus, all in all, we get (note that $\mathrm{R}(t^{\mathrm{safe}}) = \mathrm{R}(r^L)$)

$$\begin{aligned}|T^{\mathrm{alg}}| &\overset{\text{Lem. 3.22}}{\leq} r^L + 2|\mathrm{R}(r^L)| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| - |\mathrm{pos}(r^L)| \\ &\overset{(24)}{\leq} \rho|T^{\mathrm{greedy}}(t^{\mathrm{safe}})| - 2|\mathrm{L}(t^{\mathrm{safe}})| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| \\ &= \rho|T^{\mathrm{greedy}}(t^{\mathrm{safe}})| + 2|\mathrm{L} - \mathrm{L}(t^{\mathrm{safe}})| + 2|\mathrm{R}(t)| \\ &\overset{(25)}{<} \rho|T^{\mathrm{greedy}}(r^L)| + 2|\mathrm{R}(t)| \overset{(26)}{<} \rho|T^{\mathrm{opt}}|.\end{aligned}$$

Case A(ii): *The interval* $[r^{\mathrm{R}}, r^{\mathrm{L}})$ *is unsafe.*
In this case, we have $\textsc{Greedy}_1(t') = \mathrm{R}(t')$ whenever at time $t \in (r^{\mathrm{R}}, r^{\mathrm{L}}]$ a leftmost extreme is released, by Lemma$_{\mathrm{sym}}$ 3.9.

In the following, we will regard two cases. Either the server does not move to the left at all during $[r^{\mathrm{R}}, r^{\mathrm{L}}]$ or it moves to the left but is forced to turn around before reaching a leftmost request. Recall that at time r^{L} the server starts a tour in the direction of $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$. Hence, if the server moves to the left during $[r^{\mathrm{R}}, r^{\mathrm{L}}]$, then it is forced to turn around at the latest at time r^{L}.

First suppose that Alg does not move to the left at all between r^{R} and r^{L}. Then, since $[r^{\mathrm{R}}, r^{\mathrm{L}}]$ is unsafe, the server directly starts to move to the right without interruption at time r^{R} (Case (B1), Case (B2) or Case (C)). Thus, we have $r^{\mathrm{L}} = r^{\mathrm{R}} + |\mathrm{pos}(r^{\mathrm{R}}) - \mathrm{pos}(r^{\mathrm{L}})|$, because $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$ is still unserved at time r^{L}. Using Lemma 3.22, Equation (23), and $|\mathrm{pos}(t) - \mathrm{R}(r^{\mathrm{L}})| \le |T^{\mathrm{opt}}|/2$ by Lemma 3.1, we get

$$\begin{aligned}
|T^{\mathrm{alg}}| &\overset{\text{Lem. 3.22}}{\le} r^{\mathrm{L}} + 2|\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| - |\mathrm{pos}(r^{\mathrm{L}})| \\
&= r^{\mathrm{R}} + |\mathrm{pos}(r^{\mathrm{R}}) - \mathrm{pos}(r^{\mathrm{L}})| + |\mathrm{R}(r^{\mathrm{L}})| - |\mathrm{pos}(r^{\mathrm{L}})| \\
&\quad + |\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| \\
&\le r^{\mathrm{R}} + |\mathrm{pos}(r^{\mathrm{R}}) - \mathrm{R}(r^{\mathrm{L}})| + |\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| \\
&\overset{(23)}{\le} 3|T^{\mathrm{opt}}|/2 < \rho|T^{\mathrm{opt}}|.
\end{aligned}$$

Next, suppose that the server does move to the left during $[r^{\mathrm{R}}, r^{\mathrm{L}}]$. Denote by $r' \in (r^{\mathrm{R}}, r^{\mathrm{L}}]$ the last point in time at which the server turns around to move to the right after moving to the left before. As by assumption, no leftmost request is served during this interval, this in particular means that at time r' a new leftmost request is released. The release of $\sigma^{\mathrm{L}}(r')$ thus has the effect that the server starts to move to the right without having served the previous leftmost request. Note, that such a r' always exists in the interval $(r^{\mathrm{R}}, r^{\mathrm{L}}]$ (if we assume that Alg is moving to the left at some point in this time frame), because at time r^{L} the server starts a tour in the direction of $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$. Now the requirements of Lemma 3.21 are fulfilled, which implies that during $[r^{\mathrm{R}}, r']$ the server only moves to the left, during $[r', r^{\mathrm{L}}]$ the server only moves to the right, and $\mathrm{sign}(\mathrm{pos}(r^{\mathrm{R}})) = \mathrm{sign}(\mathrm{pos}(r'))$.

In particular, we have $|\mathrm{L}(r^{\mathrm{R}})| > 0$ as the server starts a tour in the direction of a leftmost extreme at time r^{R}.
We now make a case distinction regarding the position of the server at time r'.
First suppose $\mathrm{pos}(r') \ge 0$. Since during $[r^{\mathrm{R}}, r']$ the server only moves to the left by assumption this implies we have $\mathrm{pos}(r^{\mathrm{R}}) \ge \mathrm{pos}(r') \ge 0$ and $r' = r^{\mathrm{R}} + |\mathrm{pos}(r') - \mathrm{pos}(r^{\mathrm{R}})| = r^{\mathrm{R}} + |\mathrm{pos}(r^{\mathrm{R}})| - |\mathrm{pos}(r')|$. During $[r', r^{\mathrm{L}}]$ the server only moves to the right, thus $r^{\mathrm{L}} = r' + |\mathrm{pos}(r^{\mathrm{L}}) - \mathrm{pos}(r')| = r' + |\mathrm{pos}(r^{\mathrm{L}})| - |\mathrm{pos}(r')|$, because $\mathrm{pos}(r^{\mathrm{L}}) \ge \mathrm{pos}(r') \ge 0$ (we have $\mathrm{pos}(r^{\mathrm{L}}) \ge 0$ by our assumption from the beginning of the proof). Note that we have $|\mathrm{pos}(r^{\mathrm{R}})| < |\mathrm{R}(r^{\mathrm{L}})| \le |\mathrm{L}|$, because $\mathrm{far}(r^{\mathrm{L}}) = \mathrm{L}$ by assumption. Thus, we have $|\mathrm{pos}(r^{\mathrm{R}})| + |\mathrm{R}(r^{\mathrm{L}})| \le |\mathrm{R}(r^{\mathrm{L}})| + |\mathrm{L}| \le |T^{\mathrm{opt}}|/2$. This implies together with Lemma 3.22 that

$$\begin{aligned}
|T^{\mathrm{alg}}| &\overset{\text{Lem. 3.22}}{\le} r^{\mathrm{L}} + 2|\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| - |\mathrm{pos}(r^{\mathrm{L}})| \\
&= r' + 2|\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| + |\mathrm{pos}(r')| \\
&= r^{\mathrm{R}} + |\mathrm{pos}(r^{\mathrm{R}})| + 2|\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| \\
&\overset{(23)}{\le} 3|T^{\mathrm{opt}}|/2.
\end{aligned}$$

Next suppose $\mathrm{pos}(r') < 0$, i.e., also $\mathrm{pos}(r^{\mathrm{R}}) < 0$. Between r^{R} and r' the server moves to the left without interruption and between r' and r^{L} it moves to the right without interruption. With $\mathrm{pos}(r^{\mathrm{L}}) \geq 0$ by the assumptions from the beginning of the proof and $\mathrm{pos}(r^{\mathrm{R}}) < 0$ this implies $r^{\mathrm{L}} \leq r^{\mathrm{R}} + |\mathrm{L}(r^{\mathrm{R}}) - \mathrm{pos}(r^{\mathrm{R}})| + |\mathrm{L}(r^{\mathrm{R}}) - \mathrm{pos}(r^{\mathrm{L}})| = r^{\mathrm{R}} + |\mathrm{L}(r^{\mathrm{R}})| - |\mathrm{pos}(r^{\mathrm{R}})| + |\mathrm{L}(r^{\mathrm{R}})| + |\mathrm{pos}(r^{\mathrm{L}})|$. Using Lemma 3.22 this yields

$$\begin{aligned}|T^{\mathrm{alg}}| &\overset{\text{Lem. 3.22}}{\leq} r^{\mathrm{L}} + 2|\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| - |\mathrm{pos}(r^{\mathrm{L}})| \\ &\leq r^{\mathrm{R}} + 2|\mathrm{L}(r^{\mathrm{R}})| + 2|\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| - |\mathrm{pos}(r^{\mathrm{R}})|.\end{aligned}$$

As

$$|T^{\mathrm{opt}}| \overset{(23)}{\geq} r^{\mathrm{R}} + |\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}| + 2|\mathrm{R}(t)|,$$

we are left to prove that

$$2|\mathrm{L}(r^{\mathrm{R}})| + |\mathrm{R}(r^{\mathrm{L}})| - |\mathrm{pos}(r^{\mathrm{R}})| \leq (\rho - 1)|T^{\mathrm{opt}}|.$$

At time r^{R}, we have $|\mathrm{L}(r^{\mathrm{R}})| > 0$, $|\mathrm{R}(r^{\mathrm{R}})| = |\mathrm{R}(r^{\mathrm{L}})| \geq |\mathrm{L}(r^{\mathrm{R}})|$ by assumption (see Equation (28)). Also r^{R} is unsafe. Hence, at time r^{R} all conditions for Lemma 3.10 are fulfilled, which implies

$$\begin{aligned}2|\mathrm{L}(r^{\mathrm{R}})| + |\mathrm{R}(r^{\mathrm{L}})| - |\mathrm{pos}(r^{\mathrm{L}})| &< (\rho - 1)(r^{\mathrm{R}} + |\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}|) \\ &\overset{(23)}{\leq} (\rho - 1)|T^{\mathrm{opt}}|.\end{aligned}$$

Case B: *During* $[r^{\mathrm{R}}, r^{\mathrm{L}}]$ *a leftmost extreme is served.*
Let $r' \in [r^{\mathrm{R}}, r^{\mathrm{L}}]$ be the last release date of a request before ALG serves a leftmost extreme for the last time in $[r^{\mathrm{R}}, r^{\mathrm{L}}]$. In particular this means $|\mathrm{L}(r')| > 0$. At time r' the server starts a tour in the direction of the current leftmost extreme $\sigma^{\mathrm{L}}(r')$ and does not turn around before serving it. Recall that at time r^{L} a new leftmost extreme is released and that r^{L} is unsafe.

Suppose that at time r', we have $\textsc{Greedy}_1(r') = \mathrm{R}(r')$, or $\textsc{Greedy}_1(r') = \mathrm{L}(r')$ and $\mathrm{far}(r') = \mathrm{L}(r')$. Then the requirements for Lemma$_{\mathrm{sym}}$ 3.19 are fulfilled, and we have that

$$2|\mathrm{R}(r^{\mathrm{L}})| + |\mathrm{L}| - |\mathrm{pos}(r^{\mathrm{L}})| \leq (\rho - 1)|T^{\mathrm{opt}}|,$$

using Equation (23). Next suppose that $\textsc{Greedy}_1(r') = \mathrm{L}(r')$ and $\mathrm{far}(r') = \mathrm{R}(r')$, i.e., $\mathrm{R}(r') \geq |\mathrm{L}(r')|$.

By Lemma 3.11, we have $\mathrm{pos}(r') < 0$. As the safe tour at time r' serves $\sigma^{\mathrm{R}}(r')$ first while the server starts a tour in the direction of $\sigma^{\mathrm{L}}(r')$, we know that time r' is unsafe. Assume $r' > r^{\mathrm{R}}$. Then by definition of r', a new leftmost extreme is released at time r'. Lemma 3.9 implies that r' is safe, a contradiction. Hence, we have $r' = r^{\mathrm{R}}$. Now Lemma 3.10 implies using Equation (23) that $2|\mathrm{R}(r^{\mathrm{L}})| + |\mathrm{L}(r^{\mathrm{L}})| - |\mathrm{pos}(r^{\mathrm{L}})| \leq (\rho - 1)|T^{\mathrm{opt}}|$.
Hence, in general, we have that

$$2|\mathrm{R}(r^{\mathrm{L}})| + |\mathrm{L}| - |\mathrm{pos}(r^{\mathrm{L}})| \leq (\rho - 1)|T^{\mathrm{opt}}|.$$

Thus, we can deduce

$$\begin{aligned}|T^{\mathrm{alg}}| &\leq r^{\mathrm{L}} + 2|\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| - |\mathrm{pos}(r^{\mathrm{L}})| \\ &\leq r^{\mathrm{L}} + |\mathrm{L}| + 2|\mathrm{R}(t)| + (\rho - 1)|T^{\mathrm{opt}}| \\ &\leq \rho|T^{\mathrm{opt}}|.\end{aligned}$$

□

With the next lemma the analysis of Case 2.2 is complete.

Lemma 3.24. *Assume the algorithm is in Setting 2 and that* $\text{Greedy}_1(r) = \mathrm{R}(r)$ *and* $\mathrm{far}(r) = \mathrm{L}$. *Then the tour planned at time t is ρ-competitive.*

Proof. Lemma 3.20 yields $r = r^{\mathrm{L}}$ and $\mathrm{pos}(r^{\mathrm{L}}) \geq 0$. If $[r^{\mathrm{L}}, t)$ contains a safe release time, then Lemma 3.20 implies that the tour computed at time t is ρ-competitive. Hence, we only need to consider the case where $[r^{\mathrm{L}}, t)$ is unsafe. In this case Lemma 3.20 implies

$$\text{Greedy}_1(\hat{t}) = \mathrm{L}, \text{ for all } \hat{t} \in (r^{\mathrm{L}}, t] \text{ at which a rightmost extreme appears.}$$

At time r^{L}, we have $|\mathrm{L}| \geq |\mathrm{R}(r^{\mathrm{L}})|$, because $\mathrm{far}(r^{\mathrm{L}}) = \mathrm{L}$ by assumption. Also, recall that $|\mathrm{R}(r^{\mathrm{L}})| > 0$ and that at time r^{L} a leftmost request is released. Thus, at time r^{L} all conditions for Lemma 3.10 are fulfilled, which implies

$$2|\mathrm{R}(r^{\mathrm{L}})| + |\mathrm{L}| - |\mathrm{pos}(r^{\mathrm{L}})| < (\rho - 1)(r^{\mathrm{L}} + |\mathrm{L}| + 2|\mathrm{R}(r^{\mathrm{L}})|). \tag{27}$$

We have to distinguish several cases. First suppose that $|\mathrm{R}(r^{\mathrm{L}})| \leq |\mathrm{R}(t)|$. Lemma 3.22 and Equation (27) and $r^{\mathrm{L}} + |\mathrm{L}| + 2|\mathrm{R}(r^{\mathrm{L}})| \leq r^{\mathrm{L}} + |\mathrm{L}| + 2|\mathrm{R}(t)| \leq |T^{\mathrm{opt}}|$ yield

$$\begin{aligned}
|T^{\mathrm{alg}}| &\overset{\text{Lem. 3.22}}{\leq} r^{\mathrm{L}} + 2|\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| - |\mathrm{pos}(r^{\mathrm{L}})| \\
&\overset{(27)}{\leq} (\rho - 1)(r^{\mathrm{L}} + |\mathrm{L}| + 2|\mathrm{R}(r^{\mathrm{L}})|) + r^{\mathrm{L}} + |\mathrm{L}| + 2|\mathrm{R}(t)| \\
&\leq \rho(r^{\mathrm{L}} + |\mathrm{L}| + 2|\mathrm{R}(t)|) \\
&\leq \rho|T^{\mathrm{opt}}|.
\end{aligned}$$

Next suppose that

$$|\mathrm{R}(r^{\mathrm{L}})| > |\mathrm{R}(t)|. \tag{28}$$

To analyze this case, we have to take the behavior of the offline optimum into account.

First, suppose that in the offline optimal solution $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$ is served after $\sigma^{\mathrm{L}}(t)$, i.e.,

$$|T^{\mathrm{opt}}| \geq r^{\mathrm{L}} + |\mathrm{L}| + 2|\mathrm{R}(r^{\mathrm{L}})|. \tag{29}$$

Inequality Equation (27) and $|\mathrm{R}(r^{\mathrm{L}})| > |\mathrm{R}(t)|$ imply $2|\mathrm{R}(t)| + |\mathrm{L}| - |\mathrm{pos}(r^{\mathrm{L}})| < (\rho - 1)(r^{\mathrm{L}} + |\mathrm{L}| + 2|\mathrm{R}(r^{\mathrm{L}})|)$. Thus, the following holds:

$$\begin{aligned}
|T^{\mathrm{alg}}| &\overset{\text{Lem. 3.22}}{\leq} r^{\mathrm{L}} + 2|\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| - |\mathrm{pos}(r^{\mathrm{L}})| \\
&\leq (\rho - 1)\left(r^{\mathrm{L}} + |\mathrm{L}| + 2|\mathrm{R}(r^{\mathrm{L}})|\right) + r^{\mathrm{L}} + |\mathrm{L}| + 2|\mathrm{R}(r^{\mathrm{L}})| \\
&= \rho\left(r^{\mathrm{L}} + |\mathrm{L}| + 2|\mathrm{R}(t)|\right) \\
&\overset{(29)}{\leq} \rho|T^{\mathrm{opt}}|.
\end{aligned}$$

Second, suppose that in the offline optimal solution the requests $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$ and $\sigma^{\mathrm{R}}(t)$ are served before $\sigma^{\mathrm{L}}(t)$. Then,

$$|T^{\mathrm{opt}}| \geq t + |\mathrm{R}(t)| + 2|\mathrm{L}|. \tag{30}$$

By assumption the request $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$ is served during $[r^{\mathrm{L}}, t]$. Thus, the $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$ is served before the release of $\sigma^{\mathrm{R}}(t)$ at time t, which implies

$$t \geq r^{\mathrm{L}} + |\mathrm{R}(r^{\mathrm{L}})| - |\mathrm{pos}(r^{\mathrm{L}})|. \tag{31}$$

Hence, again using Lemma 3.22, $|\mathrm{R}(r^{\mathrm{L}})| \leq |\mathrm{L}|$, $\mathrm{pos}(t) \in [\mathrm{L}, \mathrm{R}(t)]$ by Lemma 3.1, and our assumption regarding the offline optimum, we get that

$$\begin{aligned} |T^{\mathrm{alg}}| &\overset{\text{Lem. 3.22}}{\leq} r^{\mathrm{L}} + 2|\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| - |\mathrm{pos}(r^{\mathrm{L}})| \\ &\overset{(31)}{\leq} t + |\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}| + 2|\mathrm{R}(t)| \\ &\overset{(30)}{\leq} |T^{\mathrm{opt}}| + |\mathrm{R}(r^{\mathrm{L}})| + |\mathrm{R}(t)| \\ &\leq |T^{\mathrm{opt}}| + |\mathrm{L}| + |\mathrm{R}(t)| \\ &\overset{\text{Lem. 3.1}}{\leq} 3|T^{\mathrm{opt}}|/2. \end{aligned}$$

The last case when in the optimal offline solution $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$ is served before $\sigma^{\mathrm{L}}(t)$ while $\sigma^{\mathrm{R}}(t)$ is served after $\sigma^{\mathrm{L}}(t)$ is handled by Lemma 3.23. ◻

Proof of Theorem 3.5

We are now finally able to to give the proof of Theorem 3.5.

Proof of Theorem 3.5. Recall that in our analysis of Algorithm 1, we assumed without loss of generality that at time t a new rightmost extreme is released. If t is safe, then $T^{\mathrm{alg}}(t)$ is ρ-competitive by the definition of the safe tour. Also, we have shown in Lemmas 3.8 and 3.9 that $T^{\mathrm{alg}}(t)$ is ρ-competitive if $|\mathrm{L}| = 0$ or $\textsc{Greedy}_1(t) = \mathrm{R}(t)$.

It remains to analyze the algorithm when the assumptions in Equation (3) are fulfilled. This was done by analyzing Cases 1 and 2. We showed in Lemmas 3.15, 3.16, 3.18, and 3.24 that the tours planned in the considered subcases are ρ-competitive. ◻

4 LOWER BOUND FOR OPEN ONLINE TSP

In this section, we consider open online TSP on the line and give a tight lower bound on the best-possible competitive ratio. Note that a lower bound of 2 is obvious: At time 1, we present a request either at -1 or 1, whichever is further away from the online server. The online tour has length at least 2 while the optimum tour has length 1. Remarkably, we are able to show a slightly larger bound that turns out to be tight.

Theorem 4.1. *Let $\rho \approx 2.04$ be the second-largest root (out of the four real roots) of $9\rho^4 - 18\rho^3 - 78\rho^2 + 210\rho - 107$. There is no $(\rho - \varepsilon)$-competitive algorithm for open TSP on the line for any $\varepsilon > 0$.*

In the following, we fix any online algorithm $\textsc{Alg}$ and $\rho' \in (2, \rho)$ and describe an adversarial strategy that forces $|T^{\mathrm{alg}}|$ to be larger than $|T^{\mathrm{opt}}|$ by a factor of at least ρ'. After the first request σ_0^{R} is released at $r_0 = 1$ to the right[3] of the origin, we alternatingly present leftmost and rightmost extreme requests, in the i-th iteration called σ_i^{L} and σ_i^{R}, respectively, depending on $\textsc{Alg}$'s behavior. Roughly, a new leftmost request σ_i^{L} appears whenever the last rightmost request $\sigma_{i-1}^{\mathrm{R}}$ is served, and a new rightmost request σ_i^{R} appears when $\textsc{Alg}$ has moved close enough to σ_i^{L}. Importantly, we will show that, eventually, some pair $(\sigma_i^{\mathrm{L}}, \sigma_i^{\mathrm{R}})$ is *critical*, in the following sense.

Definition 4.2. We call the last two requests $\sigma_0^* = (a_0^*; |a_0^*|)$ and $\sigma_1^* = (a_1^*; |a_1^*|)$ of a request sequence with $\mathrm{sign}(a_0^*) \neq \mathrm{sign}(a_1^*)$ and $0 < |a_0^*| \leq |a_1^*|$ *critical* for $\textsc{Alg}$ if the following conditions hold:

(i) Both tours $\mathrm{move}(a_0^*) \oplus \mathrm{move}(a_1^*)$ and $\mathrm{move}(a_1^*) \oplus \mathrm{move}(a_0^*)$ serve all the requests presented until time $|a_1^*|$.

[3] We assume $\mathrm{pos}(1) \leq 0$; the other case is symmetrical.

(ii) Alg serves both σ_0^* and σ_1^* after time $|a_1^*|$, and $\mathrm{pos}(|a_1^*|)$ lies between a_0^* and a_1^*.
(iii) Let $k \in \{0,1\}$ be such that Alg serves σ_k^* before σ_{1-k}^*. Then Alg serves σ_k^* no earlier than $t^* := (2\rho' - 2) \cdot |a_{1-k}^*| + (\rho' - 2) \cdot |a_k^*|$.
(iv) We have $|a_{1-k}^*|/|a_k^*| \leq 2$.

Indeed, we will show the following lemma.

Lemma 4.3. *If there is a request sequence with two critical requests for Alg, then we can release additional requests such that Alg is not $(\rho' - \varepsilon)$-competitive on the resulting instance for any $\varepsilon > 0$.*

In the proof, we use the notation from Definition 4.2. We assume that $\mathrm{sign}(a_k^*) \geq 0$; the other case is symmetric. For the sake of readability, we define $\sigma^{\mathrm{L}} = (\mathrm{L}; -\mathrm{L}) := \sigma_{1-k}^*$ and $\sigma^{\mathrm{R}} = (\mathrm{R}; \mathrm{R}) := \sigma_k^*$.

Conceptually, we want to present additional requests after $|a_1^*|$ so that Alg serves σ^{R} before σ^{L}. However, it will turn out that serving σ^{R} first is a mistake for Alg compared to using the tour $T^{\mathrm{LR}} := \mathrm{move}(\sigma^{\mathrm{L}}) \oplus \mathrm{move}(\sigma^{\mathrm{R}})$. Roughly, we make Opt follow the tour T^{LR} and then let it continue moving to the right until all requests are served by Alg. Accordingly, we will ensure that all additional requests we introduce coincide with Opt's position at their release time.

Assume that we could force Alg to serve σ^{L} immediately after σ^{R}, before serving any additional requests. In this case, we could simply introduce another request at position R at time $|T^{\mathrm{LR}}|$, and, by Definition 4.2 (iii), we would have

$$\begin{aligned} |T^{\mathrm{alg}}| &\geq t^* + 2(|\mathrm{R}| + |\mathrm{L}|) \\ &= (2\rho' - 2) \cdot |\mathrm{L}| + (\rho' - 2) \cdot |\mathrm{R}| + 2(|\mathrm{R}| + |\mathrm{L}|) \\ &= \rho'(2|\mathrm{L}| + |\mathrm{R}|) = \rho' |T^{\mathrm{opt}}|, \end{aligned} \tag{32}$$

as claimed.

In general, however, Alg may not serve σ^{L} immediately after σ^{R}, for example, by waiting for a while at R—which forces us to postpone the release of additional requests. Of course, Alg needs to start moving toward σ^{L} at some point to stay competitve if no new requests appear. Our goal is to balance these two effects by introducing one or two new requests.

The additional requests we use depend on the tour that Alg takes after time $|a_1^*|$. Toward this, let t^{**} be the earliest possible time that a server starting in R at time t^* could serve σ^{L}, that is,

$$\begin{aligned} t^{**} &:= t^* + |\mathrm{R}| + |\mathrm{L}| \\ &= (2\rho' - 2) \cdot |\mathrm{L}| + (\rho' - 2) \cdot |\mathrm{R}| + |\mathrm{R}| + |\mathrm{L}| \\ &= (2\rho' - 1) \cdot |\mathrm{L}| + (\rho' - 1) \cdot |\mathrm{R}|. \end{aligned}$$

We characterize the trajectory of Alg at time $t \geq |a_1^*|$ by the difference between t^{**} and the earliest possible time that Alg can still serve σ^{L} if it aborts its tour at time t and takes the shortest tour serving σ^{R} (if needed) and then σ^{L}. Formally, for $t \geq |a_1^*|$, we let

$$\mathrm{delay}(t) := \begin{cases} t + 2|\mathrm{R}| + |\mathrm{L}| - \mathrm{pos}(t) - t^{**} & \text{if by time } t \text{ request } \sigma^{\mathrm{R}} \text{ has not yet been served;} \\ t + |\mathrm{L}| + \mathrm{pos}(t) - t^{**} & \text{if by time } t \text{ request } \sigma^{\mathrm{R}} \text{ has been served, but } \sigma^{\mathrm{L}} \text{ has not; and} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

It is easy to see that the following properties hold.

Fact 1. *Consider some t such that* $\mathrm{delay}(t)$ *is defined. Let $\mathcal{T}$ be the set of tours that start in position* $\mathrm{pos}(t)$ *at time t and, if* Alg *has not served σ^{R} at time t, that do not visit* L *before* R. *The following is true:*

(i) There is no tour $T \in \mathcal{T}$ that arrives at L *earlier than* $t^{**} + \mathrm{delay}(t)$.
(ii) There is a tour $T \in \mathcal{T}$ that arrives at L *at time* $t^{**} + \mathrm{delay}(t)$.

The following two lemmata state useful properties of the delay function that will be used to define the additional requests.

Lemma 4.4. *There exists $W \geq 0$ with*

$$\mathrm{delay}\left(|T^{\mathrm{LR}}| + \frac{W}{\rho' - 1}\right) = W. \tag{33}$$

Proof. First observe that Alg has served neither σ^{L} nor σ^{R} at time $|a_1^*|$, by Definition 4.2 (ii). Since, by definition, σ^{L} is served after σ^{R}, and, hence, σ^{L} is still unserved at time $|a_1^*| + |\mathrm{L}| + |\mathrm{R}| \geq |T^{\mathrm{LR}}| = 2|\mathrm{L}| + |\mathrm{R}|$. Therefore, $\mathrm{delay}(|T^{\mathrm{LR}}|)$ is defined. Now note that we have

$$t^* = (2\rho' - 2) \cdot |\mathrm{L}| + (\rho' - 2) \cdot |\mathrm{R}| < 2.08 \cdot |\mathrm{L}| + 0.04 \cdot |\mathrm{R}| \leq 2|\mathrm{L}| + |\mathrm{R}| = |T^{\mathrm{LR}}|,$$

where the last inequality follows by Definition 4.2 (iv). Since Alg does not serve σ^{R} earlier than t^* by Definition 4.2 (iii), this implies that no tour $T \in \mathcal{T}$ (as in Fact 1) can arrive at L earlier than $t^{**} = t^* + |\mathrm{L}| + |\mathrm{R}|$, and thus

$$\mathrm{delay}(|T^{\mathrm{LR}}|) \geq 0 \tag{34}$$

by Fact 1 (ii).

If $\mathrm{delay}(|T^{\mathrm{LR}}|) = 0$, then we have $W = 0$, and we are done. Otherwise, by Inequality Equation (34), we must have $\mathrm{delay}(|T^{\mathrm{LR}}|) > 0$. Observe that, if no new requests appear, Alg must serve σ^{L} at some time to stay $(\rho - \varepsilon)$-competitive. Let W^* be chosen such that Alg serves σ^{L} at time $|T^{\mathrm{LR}}| + W^*/(\rho' - 1)$, that is,

$$\mathrm{delay}\left(|T^{\mathrm{LR}}| + \frac{W^*}{\rho' - 1} - \varepsilon'\right)$$

is defined for some sufficiently small $\varepsilon' \leq |\mathrm{L}|$. Thus, if

$$\mathrm{delay}\left(|T^{\mathrm{LR}}| + \frac{W^*}{\rho' - 1} - \varepsilon'\right) \leq \frac{W^*}{\rho' - 1} - \varepsilon',$$

then we can find W within the interval $(0, W^* - \varepsilon(\rho' - 1)]$, since $\mathrm{delay}(|T^{\mathrm{LR}}|) > 0$ and both sides of Equation (33) are continuous, and we are done. Otherwise,

$$\mathrm{delay}\left(|T^{\mathrm{LR}}| + \frac{W^*}{\rho' - 1} - \varepsilon'\right) > \frac{W^*}{\rho' - 1} - \varepsilon'.$$

So, by Fact 1 (i), Alg has not served σ^{L} at time

$$t^{**} + \frac{W^*}{\rho' - 1} - \varepsilon' = (2\rho' - 1) \cdot |\mathrm{L}| + (\rho' - 1) \cdot |\mathrm{R}| + \frac{W^*}{\rho' - 1} - \varepsilon'$$
$$> |T^{\mathrm{LR}}| + \frac{W^*}{\rho' - 1},$$

where we use the definition of t^{**} in the first step and $\rho' > 2$, $\varepsilon' \leq |\mathrm{L}|$ in the second step. We obtain a contradiction to the definition of W^*. □

Lemma 4.5. *Let W be the value given by Lemma 4.4.* Alg *serves σ^{R} no later than time* $|T^{\mathrm{LR}}| + \frac{W}{\rho' - 1}$.

Proof. Fix $t = |T^{\mathrm{LR}}| + \frac{W}{\rho'-1}$ and let $\mathcal{T}$ be as in Fact 1. Assume that

$$|T^{\mathrm{LR}}| + \frac{W}{\rho' - 1} \geq t^* + W. \tag{35}$$

Then, by definition of W and by Fact 1 (ii), there must be $T \in \mathcal{T}$ that serves σ^{L} at time $t^{**} + \mathrm{delay}(t) = t^{**} + W$. Since $t \geq t^* + W$ by assumption, this can only be the case if Alg serves σ^{R} no later than time $t^{**} + W - |\mathrm{L}| - |\mathrm{R}| = t^* + W \leq t$ as claimed.

It remains to show Equation (35). By Definition 4.2 (i), and since $|a_0^*| \leq |a_1^*|$, the tour $\mathrm{move}(a_0^*) \oplus \mathrm{move}(a_1^*)$ is optimal. By Fact 1 (i) and for Alg to be $(\rho - \varepsilon)$-competitive, we thus have

$$t^{**} + \mathrm{delay}(t) \leq (\rho - \varepsilon) \cdot (2|a_0^*| + |a_1^*|),$$

and thus

$$\mathrm{delay}(t) \leq (\rho - \varepsilon) \cdot (2|a_0^*| + |a_1^*|) - t^{**} \leq |a_0^*| + |a_1^*|. \tag{36}$$

We can now derive Inequality Equation (35) (solved for W):

$$\begin{aligned}
\frac{|T^{\mathrm{LR}}| - t^*}{1 - 1/(\rho' - 1)} &= \frac{(\rho' - 1)(|T^{\mathrm{LR}}| - t^*)}{\rho' - 2} \\
&= \frac{(\rho' - 1)(4 - 2\rho')}{(\rho' - 2)} \cdot |\mathrm{L}| + \frac{(\rho' - 1)(3 - \rho')}{(\rho' - 2)} \cdot |\mathrm{R}| \\
&> -2.08|\mathrm{L}| + 25|\mathrm{R}| \\
&> |\mathrm{L}| + |\mathrm{R}| \quad \text{(Definition 4.2 (iv))} \\
&= |a_0^*| + |a_1^*| \\
&\geq W.
\end{aligned}$$

This completes the proof. □

To prove Lemma 4.3, we show that if we present an additional request at time $|T^{\mathrm{LR}}| + W/(\rho' - 1)$ (at a distance of $W/(\rho' - 1)$ to the right of σ^{R}) and Alg decides to serve σ^{L} before the new request, the ratio between Alg's and Opt's additional costs (Inequality Equation (32)) is at least ρ'. If Alg can save time by serving the new request first and does so, then we need to present yet another request.

Proof of Lemma 4.3. Let W be the value given by Lemma 4.4. We present the request

$$\sigma_+^{\mathrm{R}} = (\mathrm{R}_+; r_+^{\mathrm{R}}) := \left(|\mathrm{R}| + \frac{W}{\rho' - 1}; |T^{\mathrm{LR}}| + \frac{W}{\rho' - 1}\right)$$

and distinguish two cases:

Case 1: *At time r_+^{R}, Alg is at least as close to* L *as to* R_+*, or it deliberately serves σ^{L} before σ_+^{R}.* We do not present additional requests. Since Alg has already served σ^{R} at time r_+^{R} (Lemma 4.5), it follows by Fact 1 (i) and the definition of W that Alg does not serve σ^{L} earlier than $t^{**} + W$. Hence, the total cost of Alg is

$$\begin{aligned}
|T^{\mathrm{alg}}| &\geq t^{**} + W + |\mathrm{L}| + |\mathrm{R}| + \frac{W}{\rho' - 1} \\
&= \rho' \cdot \left(|T^{\mathrm{LR}}| + \frac{W}{\rho' - 1}\right) = \rho' \cdot r_+^{\mathrm{R}} \geq \rho' \cdot |T^{\mathrm{opt}}|,
\end{aligned}$$

which shows the claim.

Case 2: *At time r_+^{R}, Alg is closer to R_+ than to L, and it serves σ_+^{R} first.*
The offline server first serving σ^{L} continues going to the right at time r_+^{R} (note that, in fact, it has been going to the right from time $|\mathrm{L}|$ on). For later times t, we denote by

$$M(t) := \frac{-|\mathrm{L}| + (t - 2|\mathrm{L}|)}{2} = \frac{t - 3|\mathrm{L}|}{2}$$

the midpoint between the current position of the offline server and $-|\mathrm{L}|$. We distinguish two cases.

Case 2.1: *Alg does not serve σ_+^{R} until time $M^{-1}(a_+^R)$, i.e., the midpoint reaches σ_+^{R} before Alg.*
We do not present additional requests and compute Alg's cost. Because we are in Case 2, σ^{L} is also not served before $M^{-1}(a_+^R)$. Using Equation (35) from the proof of Lemma 4.5, we have $r_+^{\mathrm{R}} \geq t^* + W$ and the resulting total cost for Alg is

$$\begin{aligned}|T^{\mathrm{alg}}| &\geq M^{-1}(\mathrm{R}_+) + |\mathrm{R}_+| + |\mathrm{L}| \\ &\overset{(35)}{\geq} t^* + W + \left(M^{-1}(\mathrm{R}_+) - r_+^{\mathrm{R}}\right) + |\mathrm{R}_+| + |\mathrm{L}| \\ &= \frac{(\rho' + 1)W}{\rho' - 1} + \rho'|\mathrm{R}| + 2\rho'|\mathrm{L}| \\ &= \rho' \cdot \left(|T^{\mathrm{LR}}| + \frac{W}{\rho' - 1}\right) + \frac{W}{\rho' - 1} \geq \rho' \cdot |T^{\mathrm{opt}}|,\end{aligned}$$

which shows the claim.

Case 2.2: *Alg serves σ_+^{R} before time $M^{-1}(a_+^R)$.*
By the definition of W, the delay function is defined at time r_+^{R}. Hence, Alg cannot have served σ^{L} before this time. Since Alg is to the right of the midpoint at time r_+^{R}, there must be a (first) time t_{mid} after Alg has served σ_+^{R} at which $M(t_{\mathrm{mid}}) = \mathrm{pos}(t_{\mathrm{mid}})$. We present a last request $\sigma_{++}^{\mathrm{R}} = (\mathrm{R}_{++}; r_{++}^{\mathrm{R}}) := (t_{\mathrm{mid}} - 2|\mathrm{L}|, t_{\mathrm{mid}})$. Because at time t_{mid} Alg is at the midpoint between L and R_{++}, at this time the tours $\mathrm{move}(\sigma_{++}^{\mathrm{R}}) \oplus \mathrm{move}(\sigma^{\mathrm{L}})$ and $\mathrm{move}(\sigma^{\mathrm{L}}) \oplus \mathrm{move}(\sigma_{++}^{\mathrm{R}})$ incur identical costs for Alg, and we have

$$|T^{\mathrm{alg}}| = t_{\mathrm{mid}} + 3\left(\frac{t_{\mathrm{mid}} - 3|\mathrm{L}|}{2} + |\mathrm{L}|\right) = \frac{5t_{\mathrm{mid}} - 3|\mathrm{L}|}{2}.$$

For Alg to not be $(\rho - \varepsilon)$-competitive, we need

$$|T^{\mathrm{alg}}| \geq \rho' \cdot |T^{\mathrm{opt}}| = \rho' \cdot |\mathrm{R}_{++}| = \rho' \cdot t_{\mathrm{mid}},$$

which is equivalent to

$$(5 - 2\rho') \cdot t_{\mathrm{mid}} \geq 3|\mathrm{L}|. \tag{37}$$

Since the coefficient $5 - 2\rho'$ of t_{mid} is positive, we may assume that t_{mid} is minimal to show Equation (37). By assumption, σ_+^{R} is already served at time t_{mid}. Hence, t_{mid} is minimized if, starting at time r_+^{R} at $\mathrm{pos}(r_+^{\mathrm{R}})$, Alg serves σ_+^{R} and then goes to the left, both at full speed. We get

$$\begin{aligned}&|\mathrm{R}_+| - \left(t_{\mathrm{mid}} - r_+^{\mathrm{R}} - \left|\mathrm{R}_+ - \mathrm{pos}(r_+^{\mathrm{R}})\right|\right) = \mathrm{pos}(t_{\mathrm{mid}}) \\ &= M(t_{\mathrm{mid}}) = \frac{t_{\mathrm{mid}} - 3|\mathrm{L}|}{2}.\end{aligned} \tag{38}$$

By Lemma 4.5, σ^{R} is already served at time $|T^{\mathrm{LR}}| + \frac{W}{\rho'-1}$, and thus we can solve Equation (33) for $\mathrm{pos}(r_+^{\mathrm{R}})$:

$$\begin{aligned}
\mathrm{pos}(r_+^{\mathrm{R}}) &= \mathrm{delay}\left(|T^{\mathrm{LR}}| + \frac{W}{\rho'-1}\right) - |\mathrm{L}| - |T^{\mathrm{LR}}| - \frac{W}{\rho'-1} + t^{**} \\
&= W - 2|\mathrm{L}| - \frac{W}{\rho'-1} + t^{*} \\
&= (2\rho'-4)\cdot|\mathrm{L}| + (\rho'-2)\cdot|\mathrm{R}| + \frac{\rho'-2}{\rho'-1}\cdot W \qquad (39)\\
&< 0.08\cdot|\mathrm{L}| + 0.04\cdot|\mathrm{R}| + \frac{\rho'-2}{\rho'-1}\cdot W \\
&\le |\mathrm{R}| + \frac{W}{\rho'-1} = \mathrm{R}_+. \quad \text{(Definition 4.2 (iv))}
\end{aligned}$$

This implies that Alg is to the left of R_+ at time r_+^{R}. Using this insight, we get $|\mathrm{R}_+ - \mathrm{pos}(r_+^{\mathrm{R}})| = \mathrm{R}_+ - \mathrm{pos}(r_+^{\mathrm{R}})$. By applying this insight to (38) and solving for t_{mid}, we obtain

$$\begin{aligned}
t_{\mathrm{mid}} &= \frac{1}{3}\left(3|\mathrm{L}| + 4|\mathrm{R}_+| + 2r_+^{\mathrm{R}} - 2\mathrm{pos}(r_+^{\mathrm{R}})\right) \\
&= \frac{1}{3}\left(7|\mathrm{L}| + 6|\mathrm{R}| + \frac{6W}{\rho'-1} - 2\mathrm{pos}(r_+^{\mathrm{R}})\right) \\
&= \frac{1}{3}\left((15-4\rho')\cdot|\mathrm{L}| + (10-2\rho')\cdot|\mathrm{R}| + \frac{(10-2\rho')\cdot W}{\rho'-1}\right),
\end{aligned}$$

where we plug in the definition of R_+ in the second step and Equation (39) in the second step. By substituting this into Inequality Equation (37) and noting that it is hardest to satisfy when $W = 0$, we get

$$\frac{|\mathrm{L}|}{|\mathrm{R}|} \le \underbrace{\frac{4\rho'^2 - 30\rho' + 50}{-8\rho'^2 + 50\rho' - 66}}_{>\,2.08},$$

which is true due to Definition 4.2 (iv). □

To prove Theorem 4.1 it remains to show that we can define a request sequence (depending on Alg) that ends with a pair of critical requests. We use the following strategy:

- W.l.o.g. $\mathrm{pos}(1) \le 0$. The first request is $\sigma_0^{\mathrm{R}} := (1, 1)$.
- Whenever, at some time, called r_i^{L} in the following, a request at $\sigma_{i-1}^{\mathrm{R}}$ gets served, we present the new request $\sigma_i^{\mathrm{L}} := (\mathrm{L}_i = -r_i^{\mathrm{L}}; r_i^{\mathrm{L}})$. Based on r_i^{L}, we define for $t \ge r_i^{\mathrm{L}}$ the two functions

$$\begin{aligned}
\ell_i^{\mathrm{L}}(t) &:= (2\rho'-3)\cdot t - (3-\rho')\cdot r_i^{\mathrm{L}}, \\
\ell_i^{\mathrm{R}}(t) &:= (4-\rho')\cdot t - (2\rho'-2)\cdot r_i^{\mathrm{L}},
\end{aligned}$$

  which can as well be viewed as lines in the time–distance diagram.
- If at some time r_i^{R} after r_i^{L} Alg crosses $\ell_i^{\mathrm{L}}(t)$ or $\ell_i^{\mathrm{R}}(t)$, then we present the request $\sigma_i^{\mathrm{R}} := (\mathrm{R}_i = r_i^{\mathrm{R}}; r_i^{\mathrm{R}})$.
- We stop the procedure when one of the following cases occurs. (The pair $(\sigma_i^{\mathrm{L}}, \sigma_i^{\mathrm{R}})$ will be shown to be critical in these cases.)

  Case 1: Alg serves σ_i^{L} before σ_i^{R} if no new requests appear.

Case 2: Alg serves σ_i^{R} not before time $(2\rho' - 2) \cdot r_i^{\mathrm{L}} + (\rho' - 2) \cdot r_i^{\mathrm{R}}$ if no new requests appear.

The intuition behind the lines ℓ_i^{L} and ℓ_i^{R} is the following: Suppose the position of Alg at r_i^{R} is on or to the right of ℓ_i^{L} and Alg decides to serve σ_i^{L} before σ_i^{R} in case no new requests appear after r_i^{R}. Then the pair $(\sigma_i^{\mathrm{L}}, \sigma_i^{\mathrm{R}})$ satisfies Definition 4.2 (iii). The symmetric statement holds for ℓ_i^{R}. The following lemma ensures that, in each iteration, we obtain a (not necessarily critical) pair of unserved requests $(\sigma_i^{\mathrm{L}}, \sigma_i^{\mathrm{R}})$ and that Definition 4.2 (iv) is fulfilled.

Lemma 4.6. *Let $i \geq 1$. At time r_i^{L}, Alg is to the right of ℓ_i^{L} and ℓ_i^{R} and crosses one of them after r_i^{L} and before it serves σ_i^{L}. We also have that $r_i^{\mathrm{R}} \leq 2r_i^{\mathrm{L}}$.*

Proof. We use induction on i. For the induction base, note that $r_1^{\mathrm{L}} \in [2, \rho')$. Thus, $\ell_1^{\mathrm{L}}(r_1^{\mathrm{L}}) = -\ell_1^{\mathrm{R}}(r_1^{\mathrm{L}}) \in [6\rho' - 12, \rho' \cdot (3\rho' - 6))$. In particular, $\ell_1^{\mathrm{L}}(r_1^{\mathrm{L}})$ and $\ell_1^{\mathrm{R}}(r_1^{\mathrm{L}})$ are both contained is contained in $(-t_1, \mathrm{pos}(r_1^{\mathrm{L}})) = (\mathrm{L}_1, 1)$. Since the coefficients of t in both $\ell_1^{\mathrm{L}}(t)$ and $\ell_1^{\mathrm{R}}(t)$ are positive, there is no way for Alg to serve σ_1^{L} before crossing the line.

We get an upper bound on r_1^{R} by neglecting the possible intersection of Alg with ℓ_1^{L}. Also note that r_1^{R} is maximized if Alg goes to the right at full speed throughout the interval (t_1, r_1^{R}). In this case,

$$\ell_1^{\mathrm{R}}(r_1^{\mathrm{R}}) = (4 - \rho') \cdot r_1^{\mathrm{R}} - (2\rho' - 2) \cdot r_1^{\mathrm{L}} = 1 + (r_1^{\mathrm{R}} - r_1^{\mathrm{L}}),$$

and, using $r_1^{\mathrm{L}} \geq 2$,

$$\frac{r_1^{\mathrm{R}}}{r_1^{\mathrm{L}}} = \frac{2\rho' - 3}{3 - \rho'} + \frac{1}{(3 - \rho')r_1^{\mathrm{L}}} \leq \frac{4\rho' - 5}{6 - 2\rho'} < 1.63,$$

for $\rho' \in (2, \rho)$, in accordance with our claim.

For the induction step, consider some $i > 1$ and assume the statement was true for all smaller i. Observe that Alg served $\sigma_{i-1}^{\mathrm{R}}$ before $(2\rho' - 2) \cdot r_{i-1}^{\mathrm{L}} + (\rho' - 2) \cdot r_{i-1}^{\mathrm{R}}$, since otherwise we would have stopped the procedure. Using $r_{i-1}^{\mathrm{R}} \geq r_{i-1}^{\mathrm{L}}$, this implies

$$r_i^{\mathrm{L}} < (3\rho' - 4) \cdot r_{i-1}^{\mathrm{R}} = (3\rho' - 4) \cdot |\mathrm{R}_{i-1}| = (3\rho' - 4) \cdot \mathrm{pos}(r_i^{\mathrm{L}}).$$

Hence, $\ell_i^{\mathrm{L}}(r_i^{\mathrm{L}}) = -\ell_i^{\mathrm{R}}(r_i^{\mathrm{L}}) = (3\rho' - 6) \cdot r_i^{\mathrm{L}} < (3\rho' - 6)(3\rho' - 4) \cdot \mathrm{pos}(r_i^{\mathrm{L}})$. Consequently $\ell_i^{\mathrm{L}}(r_i^{\mathrm{L}})$ and $\ell_i^{\mathrm{R}}(r_i^{\mathrm{L}})$ are both contained in $(-r_i^{\mathrm{L}}, \mathrm{pos}(r_i^{\mathrm{L}})) = (\mathrm{L}_i, \mathrm{pos}(r_i^{\mathrm{L}}))$, which implies, as before, that Alg crosses ℓ_i^{L} or ℓ_i^{R} before serving σ_i^{L}.

We prove the upper bound on $r_i^{\mathrm{R}}/r_i^{\mathrm{L}}$. First, observe that in step $i-1$, Alg cannot have crossed ℓ_{i-1}^{R} at time r_{i-1}^{R}, because in this case σ_i^{R} can be served no earlier than

$$r_{i-1}^{\mathrm{R}} + \mathrm{R}_{i-1} - \ell_{i-1}^{\mathrm{R}}(r_{i-1}^{\mathrm{R}}) = (2\rho' - 2) \cdot r_{i-1}^{\mathrm{L}} + (\rho' - 2) \cdot r_{i-1}^{\mathrm{R}},$$

which is the condition of Case 2. Hence,

$$\ell_{i-1}^{\mathrm{R}}(r_{i-1}^{\mathrm{R}}) < \ell_{i-1}^{\mathrm{L}}(r_{i-1}^{\mathrm{R}}) \;\Leftrightarrow\; r_{i-1}^{\mathrm{L}} > \frac{7 - 3\rho'}{3\rho' - 5} \cdot r_{i-1}^{\mathrm{R}}$$

and, using $\mathrm{pos}(r_{i-1}^{\mathrm{R}}) = \ell_{i-1}^{\mathrm{L}}(r_{i-1}^{\mathrm{R}})$,

$$\frac{\mathrm{pos}(r_{i-1}^{\mathrm{R}})}{r_{i-1}^{\mathrm{R}}} = (2\rho' - 3) - (3 - \rho')\frac{r_{i-1}^{\mathrm{L}}}{r_{i-1}^{\mathrm{R}}} < (2\rho' - 3) - \frac{(3 - \rho')(7 - 3\rho')}{3\rho' - 5}.$$

This implies

$$\frac{r_i^{\mathrm{L}}}{r_{i-1}^{\mathrm{R}}} \geq \frac{r_{i-1}^{\mathrm{R}} + \mathrm{R}_{i-1} - \mathrm{pos}(r_{i-1}^{\mathrm{R}})}{r_{i-1}^{\mathrm{R}}} > \frac{-3\rho'^2 + 9\rho' - 4}{3\rho' - 5}. \tag{40}$$

As $r^{\mathrm{R}}_{i-1} = |\mathrm{pos}(r^{\mathrm{L}}_i)|$, Inequality Equation (40) gives an upper bound to the quotient $|\mathrm{pos}(t)|/t$ at the time r^{L}_i the request σ^{R}_i is served. This inequality can be interpreted as a bound on the speed of the server and it is essential for the design and proof of Algorithm 2.

To see the claimed upper bound on $r^{\mathrm{R}}_i/r^{\mathrm{L}}_i$, we again neglect the intersection with ℓ^{L}_i, and note that, $r^{\mathrm{L}}_i, r^{\mathrm{R}}_i$ are maximized if ALG goes to the right at full speed throughout the interval $(r^{\mathrm{L}}_i, r^{\mathrm{R}}_i)$. In this case,

$$\begin{aligned}\ell^{\mathrm{R}}_i(r^{\mathrm{R}}_i) &= (4-\rho')\cdot r^{\mathrm{R}}_i - (2\rho'-2)\cdot r^{\mathrm{L}}_i \\ &= \mathrm{pos}(r^{\mathrm{L}}_i) + (r^{\mathrm{R}}_i - r^{\mathrm{L}}_i) = \mathrm{R}_{i-1} + (r^{\mathrm{R}}_i - r^{\mathrm{L}}_i) = r^{\mathrm{R}}_{i-1} + (r^{\mathrm{R}}_i - r^{\mathrm{L}}_i),\end{aligned}$$

and, using Inequality Equation (40),

$$\frac{r^{\mathrm{R}}_i}{r^{\mathrm{L}}_i} \le \frac{-6\rho'^3 + 27\rho'^2 - 32\rho' + 7}{3\rho'^3 - 18\rho'^2 + 31\rho' - 12} < 1.72,$$

for $\rho' \in (2, \rho)$, as claimed. □

In the proof of Theorem 4.1, we show that Case 1 or 2 eventually occurs, and we formalize the above intuition to show, along with Lemma 4.6, that the requests $(\sigma^{\mathrm{L}}_i, \sigma^{\mathrm{R}}_i)$ are indeed critical.

PROOF OF THEOREM 4.1. The proof has two parts: We first show that we can indeed apply Lemma 4.3 in Cases 1 and 2, and then we argue that the procedure terminates.

Consider the step i in which the procedure terminates. In both cases, we apply Lemma 4.3, that is, we have to show that we have two critical requests. We set $\sigma^*_0 := \sigma^{\mathrm{L}}_i$ and $\sigma^*_1 := \sigma^{\mathrm{R}}_i$, which are obviously of the desired form $(a; |a|)$ for some a and fulfill $\mathrm{sign}(a^*_0) \neq \mathrm{sign}(a^*_1)$ as well as $0 < |a^*_0| \le |a^*_1|$. Again let σ^*_k be the request served first by ALG in case no new requests appear. We argue that Properties (i)–(iv) of Definition 4.2 are fulfilled.

The tours $\mathrm{move}(a^*_0) \oplus \mathrm{move}(a^*_1)$ and $\mathrm{move}(a^*_1) \oplus \mathrm{move}(a^*_0)$ both serve all requests presented until time r^{R}_i as all these requests have the form $(a, |a|)$ for some a and there are no requests among them outside $[a^*_0, a^*_1]$. So Property (i) is fulfilled. Since ℓ^{L}_i has a positive slope and $\ell^{\mathrm{L}}_i(r^{\mathrm{L}}_i) > \mathrm{L}_i$, Lemma 4.6 implies that σ^{L}_i is still unserved at time r^{R}_i. From $\mathrm{pos}(1) \le 0$ it follows that $\mathrm{pos}(t) < t$ at all times, hence σ^{R}_i also remains unserved at time $r^{\mathrm{R}}_i = |\mathrm{R}_i|$. In particular $\mathrm{pos}(r^{\mathrm{R}}_i) \in (\mathrm{L}_i, \mathrm{R}_i)$, and Property (ii) is also fulfilled. Property (iv) follows by Lemma 4.6. To see Property (iii), we make a case distinction:

- Consider Case 1, that is, we have $\sigma^*_k = \sigma^{\mathrm{L}}_i$. By definition of r^{R}_i, we have
$$\begin{aligned}\mathrm{pos}(r^{\mathrm{R}}_i) \ge \ell^{\mathrm{L}}_i(r^{\mathrm{R}}_i) &= (2\rho'-3)\cdot r^{\mathrm{R}}_i + (\rho'-3)\cdot r^{\mathrm{L}}_i \\ &= (2\rho'-3)\cdot |a^*_{1-k}| + (\rho'-3)\cdot |a^*_k|.\end{aligned}$$
With $\mathrm{pos}(r^{\mathrm{R}}_i) > \mathrm{L}_i$, this implies that σ^*_k is served no earlier than
$$\begin{aligned}t^* &= r^{\mathrm{R}}_i + |\mathrm{L}_i - \mathrm{pos}(r^{\mathrm{R}}_i)| = r^{\mathrm{R}}_i + |\mathrm{L}_i| + \mathrm{pos}(r^{\mathrm{R}}_i) \\ &= (2\rho'-2)\cdot |a^*_{1-k}| + (\rho'-2)\cdot |a^*_k|.\end{aligned}$$
- Consider Case 2. We can assume that Case 1 is not fulfilled, meaning that ALG serves $\sigma^{\mathrm{R}}_i = \sigma^*_k$ first and, since we are in Case 2, not earlier than
$$t^* = (2\rho'-2)\cdot r^{\mathrm{L}}_i + (\rho'-2)\cdot r^{\mathrm{R}}_i = (2\rho'-2)\cdot |a^*_{1-k}| + (\rho'-2)\cdot |a^*_k|.$$

It remains to show that the procedure terminates. We show this by contradiction. Assume Case 1 and Case 2 both never occur. First observe that, if ℓ^{R}_i is crossed at r^{R}_i, Case 2 is fulfilled as we have

$$r^{\mathrm{R}}_i + \mathrm{R}_i - \ell^{\mathrm{R}}_i(r^{\mathrm{R}}_i) = (2\rho'-2)\cdot r^{\mathrm{L}}_i + (\rho'-2)\cdot r^{\mathrm{R}}_i.$$

Thus, we have $r^{\mathrm{R}}_i = \ell^{\mathrm{L}}_i(r^{\mathrm{R}}_i)$.

We consider the difference between the release times $r_i^{\mathrm{L}}, r_{i-1}^{\mathrm{R}}$ of the two consecutive requests $\sigma_{i-1}^{\mathrm{R}}, \sigma_i^{\mathrm{L}}$. It is at least the time to move from $\ell_{i-1}^{\mathrm{L}}(r_{i-1}^{\mathrm{R}})$ to R_{i-1}:

$$r_i^{\mathrm{L}} - r_{i-1}^{\mathrm{R}} \geq \mathrm{R}_{i-1} - \ell_{i-1}^{\mathrm{L}}(r_{i-1}^{\mathrm{R}}) \Leftrightarrow r_i^{\mathrm{L}} \geq (5 - 2\rho') \cdot r_{i-1}^{\mathrm{R}} + (3 - \rho') \cdot r_{i-1}^{\mathrm{L}}. \tag{41}$$

Similarly, the difference between the release times $r_i^{\mathrm{R}}, r_i^{\mathrm{L}}$ is at least the time to move from R_{i-1} to $\ell_i^{\mathrm{L}}(r_i^{\mathrm{R}})$. If $\ell_i^{\mathrm{L}}(r_i^{\mathrm{R}}) < \mathrm{R}_{i-1}$, then this is equivalent to the following inequality, which is true otherwise, since $r_i^{\mathrm{R}} > r_i^{\mathrm{L}}$:

$$r_i^{\mathrm{R}} - r_i^{\mathrm{L}} \geq \mathrm{R}_{i-1} - \ell_i^{\mathrm{L}}(r_i^{\mathrm{R}}) \Leftrightarrow (2\rho' - 2) \cdot r_i^{\mathrm{R}} \geq r_{i-1}^{\mathrm{R}} + (4 - \rho') \cdot r_i^{\mathrm{L}}. \tag{42}$$

As Case 2 is not triggered when $\sigma_{i-1}^{\mathrm{R}}$ is served, we have

$$\begin{aligned} r_i^{\mathrm{L}} &\overset{\neg(\text{Case } 2)}{<} (2\rho' - 2) \cdot r_{i-1}^{\mathrm{L}} + (\rho' - 2) \cdot r_{i-1}^{\mathrm{R}} \\ &\overset{(41)}{<} (2\rho' - 2) \cdot r_{i-1}^{\mathrm{L}} + \frac{\rho - 2}{5 - 2\rho'} \cdot (r_i^{\mathrm{L}} - (3 - \rho') \cdot r_{i-1}^{\mathrm{L}}) \\ \Leftrightarrow \quad r_i^{\mathrm{L}} &< \frac{-3\rho'^2 + 9\rho' - 4}{7 - 3\rho'} \cdot r_{i-1}^{\mathrm{L}} \\ \Rightarrow \quad r_i^{\mathrm{L}} &< \left(\frac{-3\rho'^2 + 9\rho' - 4}{7 - 3\rho'}\right)^{i-1} \cdot r_1^{\mathrm{L}}. \end{aligned}$$

Combining Inequality Equation (41) and the fact that Case 2 does not occur also yields

$$\begin{aligned} (3 - \rho') \cdot r_i^{\mathrm{L}} + (5 - 2\rho') \cdot r_i^{\mathrm{R}} \overset{(41)}{<} r_{i+1}^{\mathrm{L}} &\overset{\neg(\text{Case } 2)}{<} (2\rho' - 2) \cdot r_i^{\mathrm{L}} + (\rho' - 2) \cdot r_i^{\mathrm{R}} \\ \Leftrightarrow \quad r_i^{\mathrm{L}} &> \frac{7 - 3\rho'}{3\rho' - 5} \cdot r_i^{\mathrm{R}}. \end{aligned} \tag{43}$$

We use this relation between the release time of right and left requests in Inequality Equation (42)

$$\begin{aligned} (2\rho' - 2) \cdot r_i^{\mathrm{R}} &> r_{i-1}^{\mathrm{R}} + (4 - \rho') \cdot \frac{7 - 3\rho'}{3\rho' - 5} \cdot r_i^{\mathrm{R}} \\ \Leftrightarrow \quad r_i^{\mathrm{R}} &> \frac{3\rho' - 5}{3\rho'^2 + 3\rho' - 18} \cdot r_{i-1}^{\mathrm{R}} \\ \Rightarrow \quad r_i^{\mathrm{R}} &> \left(\frac{3\rho' - 5}{3\rho'^2 + 3\rho' - 18}\right)^i \cdot r_0^{\mathrm{R}}. \end{aligned}$$

Combining the bounds on r^{L} and r^{R}, we get

$$\left(\frac{3\rho' - 5}{3\rho'^2 + 3\rho' - 18}\right)^i < r_i^{\mathrm{R}} \overset{(43)}{<} \frac{3\rho' - 5}{7 - 3\rho'} \cdot r_i^{\mathrm{L}} < c \cdot \left(\frac{-3\rho'^2 + 9\rho' - 4}{7 - 3\rho'}\right)^i.$$

for some constant $c > 1$. This can only be true for all i if

$$\begin{aligned} \frac{3\rho' - 5}{3\rho'^2 + 3\rho' - 18} &\leq \frac{-3\rho'^2 + 9\rho' - 4}{7 - 3\rho'} \\ \Leftrightarrow \quad 9\rho'^4 - 18\rho'^3 - 78\rho'^2 + 210\rho' - 107 &\leq 0 \\ \Leftrightarrow \quad \rho' &\geq \rho, \end{aligned}$$

where we use $\rho' \geq 2$ in the last step. This contradicts our choice of ρ' and thus proves that the request procedure terminates. □

5 ALGORITHM FOR OPEN ONLINE TSP

In this section, we propose an algorithm for the open online TSP on the line, which achieves a competitive ratio of $\rho \approx 2.04$, matching the lower bound presented in Section 4. In the first part of this section, we describe the algorithm and give some intuition. Afterwards, in the second part, we analyze the algorithm and prove its competitive ratio.

5.1 Algorithm and Intuition

We introduce a new algorithm for the open case, listed in Algorithm 2. As in Algorithm 1 for the closed case, our algorithm is given by a subroutine that is called every time a new extreme request is released, and that takes the current time, the current position of the server, and a pair of extreme requests as input. It then computes a new tour serving the current extreme requests and thus also all other requests between these extremes. The algorithm decides the order in which it serves the extremes and possibly waits based on the current position pos(t) of the server, the current time t, and the release times and positions of the single or the two extreme requests.

ALGORITHM 2: Update$(t, \text{pos}(t), \sigma^{\text{L}}(t), \sigma^{\text{R}}(t))$ for the open online TSP Problem

```
          ▷ This function is called every time a new extreme request is released.
          Input: Current time $t$
                 Current position pos$(t)$
                 Unserved extreme requests $\sigma^{\text{L}}(t)$ and $\sigma^{\text{R}}(t)$
          Output: An open TSP tour serving all remaining requests
          $T_0 := \text{move}(0) \oplus \text{waituntil}(\infty)$
          if $\exists$ only single extreme $\sigma_1 = (a_1; r_1)$ then
(P1),(E1)     return $T_0.\text{until}(\tau + |\text{pos}(\tau) - a_1| \geq \rho r_1) \oplus \text{move}(a_1)$
          if $0 \notin [\text{L}(t), \text{R}(t)]$ then
(O)           return $\text{move}(\text{near}(t)) \oplus T_0.\text{until}(\tau + |\text{pos}(\tau) - \text{far}(t)| \geq \rho \cdot r^{\text{far}}(t)) \oplus \text{move}(\text{far}(t))$
          else
              if $t + |\text{pos}(t) - \text{early}(t)| \leq L(\sigma^{\text{early}}(t), \sigma^{\text{late}}(t))$ then
(P2)              return
                  $T_0.\text{until}(\tau + |\text{pos}(\tau) - \text{early}(t)| = L(\sigma^{\text{early}}(t), \sigma^{\text{late}}(t))) \oplus \text{move}(\text{early}(t)) \oplus$
                  $\text{move}(\text{late}(t))$
              else if $t + |\text{pos}(t) - \text{late}(t)| \leq L(\sigma^{\text{late}}(t), \sigma^{\text{early}}(t))$ and
              $|\text{late}(t)| \leq \frac{3\rho-5}{(2\rho-2)(7-3\rho)}(\rho \cdot r^{\text{early}}(t) + (\rho - 2)|\text{early}(t)|)$ then
(A2)              return
                  $T_0.\text{until}(\tau + |\text{pos}(\tau) - \text{late}(t)| = L(\sigma^{\text{late}}(t), \sigma^{\text{early}}(t))) \oplus \text{move}(\text{late}(t)) \oplus$
                  $\text{move}(\text{early}(t))$
              else
(E2)              return $\text{move}(\text{early}(t)) \oplus \text{move}(\text{late}(t))$
```

The basic strategy of the algorithm is to move to the origin whenever possible and to serve the extremes in the latest tour that still guarantees a competitive ratio of ρ. The idea is to wait for more information and delay the choice in which order to serve the extremes for as long as possible. The tour that moves to the origin and waits there is defined as $T_0 := \text{move}(0) \oplus \text{waituntil}(\infty)$. We further introduce the notation $T_0.\text{until}(\text{condition}(\tau))$ to denote the tour that follows T_0 until the first time τ that satisfies "condition(τ)." Note that this may happen before the server reaches the origin.

By moving to the origin whenever possible, the algorithm ensures that the ratio $|\mathrm{pos}(t)|/t$ is bounded by a constant significantly smaller than 1. In fact, in the lower bound construction (Section 4), we observed that a ρ-competitive algorithm may not serve a request too early, i.e., $|\mathrm{pos}(t)|/t$ cannot be above a certain threshold. The exact bound on this ratio is computed in the proof of Lemma 4.6 and coincides with the following bound in Lemma 5.1. In the next section, we see that this bound is crucial for the overall analysis of the algorithm.

Lemma 5.1. *The position of the server in a tour computed by Algorithm 2 satisfies*

$$\frac{|\mathrm{pos}(t)|}{t} \leq \frac{3\rho - 5}{-3\rho^2 + 9\rho - 4} \approx 0.583$$

for all times $t \geq 0$.

We now discuss the different cases occurring in Algorithm 2 step-by-step. Let us first consider the simplest case where only one extreme request $\sigma_1 = (a_1; r_1)$ is present (cases (P1) and (E1) in Algorithm 2). The offline optimum Opt obviously cannot finish before time r_1 in this case. To guarantee ρ-competitiveness it is therefore sufficient to serve σ_1 at time ρr_1. Hence, we can afford to move to the origin and wait until the equation $t + |\mathrm{pos}(t) - a_1| = \rho r_1$ is satisfied for the current time t and position $\mathrm{pos}(t)$. In the next section, we show that this ensures that $|\mathrm{pos}(t)|/t$ is bounded as stated in Lemma 5.1. Moving to the origin and waiting there further gives us the option to change our tour without much additional cost if an extreme on the other side is released later on. We call the resulting tour in this case, the *preferred tour* (case (P1)). It can happen, however, that we are only able to serve σ_1 later than time ρr_1, i.e., $t + |\mathrm{pos}(t) - a_1| > \rho r_1$. This means that the until-condition is already satisfied and the server serves σ_1 immediately. We call this tour the *enforced tour* (case (E1)). The makespan of Algorithm 2 in this case is $|T^{\mathrm{alg}}| = r_1 + |\mathrm{pos}(r_1) - a_1|$, and we have $|T^{\mathrm{opt}}| \geq r_1$ and also $|T^{\mathrm{opt}}| \geq |\mathrm{pos}(t) - a_1|$ as Algorithm 2 only visits points on the real line that must also be visited by Opt at some time. Overall, we have $|T^{\mathrm{alg}}| \leq 2 \cdot |T^{\mathrm{opt}}|$, implying ρ-competitiveness.

Now, consider the case where two extremes $\sigma_1 = (a_1; r_1)$ and $\sigma_2 = (a_2; r_2)$ are present. The two extremes define an interval

$$[\mathrm{L}(t), \mathrm{R}(t)] = [\min\{a_1, a_2\}, \max\{a_1, a_2\}].$$

Note that $\mathrm{L}(t) < \mathrm{pos}(t) < \mathrm{R}(t)$ holds by the definition of extreme requests. Furthermore, we define $\sigma^{\mathrm{near}}(t) = (\mathrm{near}(t), r^{\mathrm{near}}(t))$ and $\sigma^{\mathrm{far}}(t) = (\mathrm{far}(t), r^{\mathrm{far}}(t))$ to denote an arbitrary extreme request at time t that is closest to the origin and the other extreme request, respectively. Similarly, let $\sigma^{\mathrm{early}}(t) = (\mathrm{early}(t), r^{\mathrm{early}}(t))$ and $\sigma^{\mathrm{late}}(t) = (\mathrm{late}(t), r^{\mathrm{late}}(t))$ be an arbitrary extreme request at time t that has been released earliest and the other extreme request, respectively.

If $0 \notin [\mathrm{L}(t), \mathrm{R}(t)]$ (case (O) in Algorithm 2), then we immediately serve $\sigma^{\mathrm{near}}(t)$ to ensure that Lemma 5.1 holds and $|\mathrm{pos}(t)|/t$ stays small. We know that the offline optimum Opt cannot finish before time $r^{\mathrm{far}}(t)$. After serving $\sigma^{\mathrm{near}}(t)$ it is therefore safe to return to the origin and wait as long as we can to reach $\mathrm{far}(t)$ at time $\rho r^{\mathrm{near}}(t)$. We can thus follow the tour T_0 after serving $\sigma^{\mathrm{near}}(t)$ at time $t + |\mathrm{pos}(t) - \mathrm{far}(t)| \geq \rho r^{\mathrm{far}}(t)$. Possibly, this equation is already satisfied from the start, and we serve $\sigma^{\mathrm{far}}(t)$ immediately.

Next, consider the case that $0 \in [\mathrm{L}(t), \mathrm{R}(t)]$ (cases (P2), (A2) and (E2)) at time $t = r^{\mathrm{late}}(t)$. We have a lower bound of $|T^{\mathrm{opt}}| \geq r^{\mathrm{early}}(t) + |\mathrm{early}(t)| + |\mathrm{late}(t)|$, irrespective of whether Opt serves $\sigma^{\mathrm{early}}(t)$ or $\sigma^{\mathrm{late}}(t)$ first. If we serve $\sigma^{\mathrm{early}}(t)$ first, which we call the *preferred tour* (case (P2)), then we ensure that the tour produced by Algorithm 2 is not longer than $\rho|T^{\mathrm{opt}}|$ by satisfying the inequality

$$t + |\mathrm{pos}(t) - \mathrm{early}(t)| + |\mathrm{early}(t)| + |\mathrm{late}(t)| \leq \rho(r^{\mathrm{early}}(t) + |\mathrm{early}(t)| + |\mathrm{late}(t)|). \tag{44}$$

Now suppose that Opt serves $\sigma^{\text{late}}(t)$ first and a new request $\sigma' = (\text{early}(t); r^{\text{late}}(t) + |\text{early}(t)| + |\text{late}(t)|)$ appears at the same position as $\sigma^{\text{early}}(t)$ when Opt arrives there. The new request does not increase the cost for Opt, which is still only lower bounded by $|T^{\text{opt}}| \geq r^{\text{late}}(t) + |\text{early}(t)| + |\text{late}(t)|$. But our algorithm, which served $\sigma^{\text{early}}(t)$ first, may be closer to $\sigma^{\text{late}}(t)$ at the time when this new request appears and now has to go all the way back to the position early(t) after serving $\sigma^{\text{late}}(t)$. The intuition for why it is sufficient to protect against this worst-case situation is that if σ' appears at a position further away from the origin, then this additional distance has to be traveled by Opt as well, and if σ' appears closer to the origin, it only benefits our algorithm. To ensure ρ-competitiveness in this scenario, the following inequality has to be satisfied additionally:

$$\begin{aligned} &t + |\text{pos}(t) - \text{early}(t)| + 2(|\text{early}(t)| + |\text{late}(t)|) \\ &\quad \leq \rho(r^{\text{late}}(t) + |\text{early}(t)| + |\text{late}(t)|). \end{aligned} \tag{45}$$

If we now define

$$\begin{aligned} L(\sigma_1, \sigma_2) := \min\{&\rho r_1 + (\rho - 1)|a_1| + (\rho - 1)|a_2|, \\ &\rho r_2 + (\rho - 2)|a_1| + (\rho - 2)|a_2|\}, \end{aligned}$$

then Inequality Equations (44) and (45) are simultaneously satisfied if and only if $t + |\text{pos}(t) - \text{early}(t)| \leq L(\sigma^{\text{early}}(t), \sigma^{\text{late}}(t))$. If this latter inequality is satisfied with some slack, then we again follow the tour T_0 until it becomes tight, so that $|\text{pos}(t)|/t$ stays small, and we are more flexible to change our tour when new requests appear.

In case the conditions for the preferred tour are not satisfied, we try to serve $\sigma^{\text{late}}(t)$ first. This is called the *anticipated tour* (case (A2)). The inequalities that need to be satisfied in this case are the same as those for the preferred tour with $\sigma^{\text{early}}(t)$ and $\sigma^{\text{late}}(t)$ exchanged. Moreover, to ensure that $|\text{pos}(t)|/t$ is bounded as claimed in Lemma 5.1, the following inequality also needs to be satisfied if Algorithm 2 serves $\sigma^{\text{late}}(t)$ first:

$$\begin{aligned} |\text{late}(t)| &\leq \frac{3\rho - 5}{(2\rho - 2)(7 - 3\rho)}(\rho r^{\text{early}}(t) + (\rho - 2)|\text{early}(t)|) \\ &\approx 1.21 \cdot r^{\text{early}}(t) + 0.02 \cdot |\text{early}(t)|. \end{aligned} \tag{46}$$

Intuitively, the inequality ensures that $\sigma^{\text{late}}(t)$ is not too far from the origin compared to $\sigma^{\text{early}}(t)$, as otherwise we would need to start too early to serve the extreme $\sigma^{\text{late}}(t)$ and would violate the bound on $|\text{pos}(t)|/t$.

Last, in case neither the conditions for the preferred tour nor the anticipated tour are met, we immediately serve the earlier request $\sigma^{\text{early}}(t)$ first and $\sigma^{\text{late}}(t)$ directly afterwards. This is called the *enforced tour* (case (E2)). The main challenge in showing ρ-competitiveness in the analysis of the algorithm in the next section is to derive a better lower bound on Opt in this case by considering extremes that must have been released before.

5.2 Algorithm Analysis

The main goal of this section is to prove the following theorem stating that the competitiveness of Algorithm 2 indeed matches the lower bound from Theorem 4.1:

Theorem 5.2. *Algorithm 2 is ρ-competitive with $\rho \approx 2.04$ being the second-largest root of the polynomial* $9\rho^4 - 18\rho^3 - 78\rho^2 + 210\rho - 107$.

Recall that we assume without loss of generality that $r \geq |a|$ holds for all requests $\sigma = (a; r)$, because the server cannot reach σ before time $|a|$ and it only helps the algorithm to know a request earlier. Note that Algorithm 2 is called only if the newly released request is an extreme.

Lemma 3.1 (1) also holds for open online TSP, and we can therefore assume for the analysis that every request σ is an extreme at its release time. Moreover, we will frequently use the following lower bounds for an optimal offline algorithm.

LEMMA 5.3. *We have the following lower bounds for the makespan of an optimal (offline) schedule* $|T^{\text{opt}}|$. *If* $\sigma_1 = (a_1; r_1)$ *and* $\sigma_2 = (a_2; r_2)$ *are two requests such that* $r_1 \leq r_2$, *then*

$$|T^{\text{opt}}| \geq r_1 + |a_1 - a_2| \geq |a_1 - a_2|.$$

Moreover, if $\sigma_1 = (a_1; r_1)$ *is a request and* $\text{pos}(t)$ *denotes the position of the server at an arbitrary time* t *in a tour computed by Algorithm 2, then*

$$|T^{\text{opt}}| \geq |\text{pos}(t) - a_1|.$$

PROOF. The first inequality follows from the fact that OPT needs to serve both σ_1 and σ_2. For the second inequality, let $p^{\text{R}}(t)$ be the rightmost location of a request seen until time t and $p^{\text{L}}(t)$ be the leftmost location of a request seen until time t. The server following the tour computed by Algorithm 2 never moves left of $p^{\text{L}}(t)$ or right of $p^{\text{R}}(t)$, thus $\text{pos}(t) \in [p^{\text{L}}(t), p^{\text{R}}(t)]$ and $|T^{\text{opt}}| \geq |p^{\text{L}}(t) - p^{\text{R}}(t)| \geq |\text{pos}(t) - a_1|$. □

The first important step of the analysis is now to show the bound on $|\text{pos}(t)|/t$ stated in Lemma 5.1.

PROOF OF LEMMA 5.1. Observe that the server in a tour computed by Algorithm 2 always either moves with unit speed or it waits at the origin. If the server moves with unit speed toward the origin, then the ratio $|\text{pos}(t)|/t$ is decreasing, and if it moves away from the origin with unit speed toward a request, then the ratio $|\text{pos}(t)|/t$ is increasing (unless it is 1). The ratio $|\text{pos}(t)|/t$ is therefore maximized at a time t when a request is served and the server moves back to the origin afterwards. To show the claim it is thus sufficient to only consider times when an extreme request is served. Let t^* be the time at which the tour serving the extreme request is planned by Algorithm 2. We distinguish between the cases that can occur in Algorithm 2 at time t^*.

(1) *Cases (P1) and (E1)*
The server follows the preferred or enforced tour serving a single extreme request $\sigma^{\text{early}}(t^*) = (a_1; r_1)$. Let t be the time this request is served. In Case (P1) the server arrives at a_1 exactly at time ρr_1 and in Case (E1) it arrives after time ρr_1. Thus, we have $t \geq \rho r_1$ and using $\rho > 2$ and $a_1 \leq r_1$, we obtain

$$\frac{\text{pos}(t)}{t} \leq \frac{a_1}{\rho r_1} \leq \frac{a_1}{\rho a_1} = \frac{1}{\rho} < \frac{1}{2} < 0.583.$$

(2) *Case (O)*
The server follows the tour defined according to Case (O). As long as the server moves toward $\sigma^{\text{near}}(t^*)$, the ratio $|\text{pos}(t)|/t$ decreases. The request $\sigma^{\text{far}}(t^*)$ is not served until time $\rho \cdot r^{\text{far}}(t^*)$. We again apply $\rho > 2$ and $\text{far}(t^*) \leq r^{\text{far}}(t^*)$ and get

$$\frac{\text{pos}(\rho \cdot r^{\text{far}}(t^*))}{\rho \cdot r^{\text{far}}(t^*)} = \frac{\text{far}(t^*)}{\rho \cdot r^{\text{far}}(t^*)} \leq \frac{\text{far}(t^*)}{\rho \cdot \text{far}(t^*)} = \frac{1}{\rho} < \frac{1}{2} < 0.583.$$

(3) *Cases (P2) and (E2)*
The algorithm follows the preferred or enforced tour serving $\sigma^{\text{early}}(t^*)$ followed by $\sigma^{\text{late}}(t^*)$. Let t be the time when the server arrives at $\text{early}(t^*)$. In Case (P2), we have $t = L(\sigma^{\text{early}}(t^*), \sigma^{\text{late}}(t^*))$, while in Case (E2), we have $t > L(\sigma^{\text{early}}(t^*), \sigma^{\text{late}}(t^*))$. Using

$|\text{early}(t^*)| \le r^{\text{early}}(t^*) \le r^{\text{late}}(t^*)$, we obtain

$$\begin{aligned} t &\ge L(\sigma^{\text{early}}(t^*), \sigma^{\text{late}}(t^*)) \\ &= \min\{\rho r^{\text{early}}(t^*) + (\rho-1)|\text{early}(t^*)| + (\rho-1)|\text{late}(t^*)|,\ \rho r^{\text{late}}(t^*) \\ &\qquad + (\rho-2)|\text{early}(t^*)| + (\rho-2)|\text{late}(t^*)|\}, \\ &\ge (2\rho-2)|\text{early}(t^*)|. \end{aligned}$$

Thus, $|\text{pos}(t)|/t \le 1/(2\rho-2) < 1/2 < 0.583$ holds, which implies the claim for $\sigma^{\text{early}}(t^*)$. After serving $\sigma^{\text{early}}(t^*)$, the server moves to late(t^*) to serve $\sigma^{\text{late}}(t^*)$. Let t' be the time the server arrives at late(t^*). We have $|\text{late}(t^*)| \le r^{\text{late}}(t^*)$, and therefore

$$\begin{aligned} t' &\ge L(\sigma^{\text{early}}(t^*), \sigma^{\text{late}}(t^*)) + |\text{early}(t^*)| + |\text{late}(t^*)| \\ &= \min\{\rho r^{\text{early}}(t^*) + \rho|\text{early}(t^*)| + \rho|\text{late}(t^*)|, \rho r^{\text{late}}(t^*) \\ &\qquad + (\rho-1)|\text{early}(t^*)| + (\rho-1)|\text{late}(t^*)|\} \\ &\ge \rho|\text{late}(t^*)|. \end{aligned}$$

We get $|\text{pos}(t')|/t' \le 1/\rho < 1/2 < 0.583$ as claimed.

(4) *Case (A2)*

In the anticipating tour Algorithm 2 serves $\sigma^{\text{late}}(t^*) = (\text{late}(t^*); r^{\text{late}}(t^*))$ before $\sigma^{\text{early}}(t^*) = (\text{early}(t^*); r^{\text{early}}(t^*))$. Moreover, as the conditions for the anticipating tour are satisfied, we also have

$$|\text{late}(t^*)| \le \frac{3\rho-5}{(2\rho-2)(7-3\rho)}(\rho r^{\text{early}}(t^*) + (\rho-2)|\text{early}(t^*)|). \tag{47}$$

Let t be the time when the server arrives at late(t^*). By using Inequality Equation (47) and $r^{\text{early}}(t^*) \le r^{\text{late}}(t^*)$, we obtain

$$\begin{aligned} t &= L(\sigma^{\text{late}}(t^*), \sigma^{\text{early}}(t^*)) \\ &= \min\{\rho r^{\text{late}}(t^*) + (\rho-1)|\text{late}(t^*)| + (\rho-1)|\text{early}(t^*)|, \rho r^{\text{early}}(t^*) \\ &\qquad + (\rho-2)|\text{late}(t^*)| + (\rho-2)|\text{early}(t^*)|\} \\ &= \rho r^{\text{early}}(t^*) + (\rho-2)|\text{late}(t^*)| + (\rho-2)|\text{early}(t^*)| \\ &\overset{(47)}{\ge} |\text{late}(t^*)|\left((\rho-2) + \frac{(2\rho-2)(7-3\rho)}{3\rho-5}\right) \\ &= |\text{late}(t^*)|\frac{-3\rho^2+9\rho-4}{3\rho-5}. \end{aligned}$$

Thus, Inequality Equation (56) is satisfied with equality.

Now, let t' be the time the server arrives at early(t^*) after having served $\sigma^{\text{late}}(t^*)$. Using $r^{\text{early}}(t^*) \ge \text{early}(t^*)$, we obtain

$$\begin{aligned} t' &= L(\sigma^{\text{late}}(t^*), \sigma^{\text{early}}(t^*)) + |\text{late}(t^*)| + |\text{early}(t^*)| \\ &= \rho r^{\text{early}}(t^*) + (\rho-1)|\text{late}(t^*)| + (\rho-1)|\text{early}(t^*)| \\ &\ge (2\rho-1)|\text{early}(t^*)|. \end{aligned}$$

Therefore, $|\text{pos}(t')|/t' \le 1/(2\rho-2) < 1/2 < 0.583$ holds, as claimed. □

We are now ready to show the main result.

Proof of Theorem 5.2. At time t a new request is released and Algorithm 2 computes a new tour T^{alg}. We show that this new tour is ρ-competitive, i.e., $|T^{\text{alg}}| \leq \rho|T^{\text{opt}}|$. In the proof, we first distinguish between the cases that can occur at time t and also in previous calls of Algorithm 2.

Without loss of generality, we assume that the new extreme request released at time t is a rightmost request denoted by $\sigma^{\text{R}}(t) = (\text{R}(t); t)$. If at the call of Algorithm 2 at time t Case (P1), (P2), or (A2) occurs, i.e., the newly computed tour T^{alg} is preferred or anticipating, then we know that T^{alg} is ρ-competitive. This is because the request served last will be served at time ρt at the latest and obviously $|T^{\text{opt}}| \geq t$. Thus, we only need to analyze the other cases that can occur at time t. If at time t a leftmost extreme $\sigma^{\text{L}}(t)$ exists, then we denote its release time by $r^{\text{L}} := r^{\text{L}}(t)$.

(1) *Case (E1) at time t*
In this case $\sigma^{\text{R}}(t) = (\text{R}(t); t)$ is the only extreme request at time t and the server immediately moves toward $\sigma^{\text{R}}(t)$ as $t + |\text{pos}(t) - \text{R}(t)| > \rho t$. We have $|T^{\text{alg}}| = t + |\text{pos}(t) - \text{R}(t)|$ and Lemma 5.3 implies $|T^{\text{opt}}| \geq t$ as well as $|T^{\text{opt}}| \geq |\text{pos}(t) - \text{R}(t)|$. Thus, $|T^{\text{alg}}| = t + |\text{pos}(t) - \text{R}(t)| \leq 2|T^{\text{opt}}| < \rho|T^{\text{opt}}|$ holds, as desired.

(2) *Case (O) at time t*
In this case two extremes $\sigma^{\text{L}}(t)$ and $\sigma^{\text{R}}(t)$ are present such that $0 \notin [\text{L}(t), \text{R}(t)]$. Recall that the position of the server is strictly between both extremes, because $\text{R}(t)$ is right of the server and $\text{L}(t)$ left of the server by definition and both requests are unserved at time t. If there is no unserved request on one side of the server, then the algorithm would be in Case (P1) or (E1).

If after serving $\sigma^{\text{early}}(t^*)$ the server follows the tour T_0 for some time, then it serves $\sigma^{\text{late}}(t^*)$ by time $\rho r^{\text{late}}(t^*)$ and the tour computed by Algorithm 2 is ρ-competitive. Thus, from now on, we suppose that the server moves toward $\sigma^{\text{early}}(t^*)$ and then $\sigma^{\text{late}}(t^*)$ without interruption. We distinguish the two cases $|\text{R}(t)| < |\text{L}(t)|$ and $|\text{R}(t)| > |\text{L}(t)|$.

We first consider the case where $|\text{R}(t)| < |\text{L}(t)|$. As we are in the Case (O), this implies that $\text{R}(t), \text{L}(t) < 0$. The tour in this case chooses to serve $\sigma^{\text{R}}(t) = \sigma^{\text{early}}(t^*)$ first. Using $|\text{L}(t) - \text{R}(t)| \leq |\text{L}(t)| \leq r^{\text{L}}$, $\text{pos}(t) \in (\text{L}(t), \text{R}(t))$ and the bounds $|T^{\text{opt}}| \geq t$ and $|\text{L}(t) - \text{R}(t)| + r^{\text{L}} \leq |T^{\text{opt}}|$ by Lemma 5.3 ($t \geq r^{\text{L}}$), we obtain

$$\begin{aligned}|T^{\text{alg}}| &= t + |\text{pos}(t) - \text{R}(t)| + |\text{R}(t) - \text{L}(t)| \\ &\leq t + |\text{pos}(t) - \text{R}(t)| + r^{\text{L}} \\ &\leq t + |\text{L}(t) - \text{R}(t)| + r^{\text{L}} \\ &\leq 2|T^{\text{opt}}|.\end{aligned}$$

Next suppose that $|\text{R}(t)| > |\text{L}(t)|$ and hence $\text{L}(t), \text{R}(t) > 0$. In this case $\sigma^{\text{L}}(t)$ is served first, and we obtain

$$|T^{\text{alg}}| = t + |\text{pos}(t) - \text{L}(t)| + |\text{L}(t) - \text{R}(t)|.$$

Recall that $\text{pos}(t) \in [\text{L}(t), \text{R}(t)]$. If $\text{pos}(t) = \text{L}(t)$ this implies that $|T^{\text{alg}}| = t + |\text{L}(t) - \text{R}(t)| \leq 2|T^{\text{opt}}|$. If $\text{pos}(t) > \text{L}(t)$, we have to take the situation during the time interval $[r^{\text{L}}, t]$ into account. As $\sigma^{\text{L}}(t)$ was released at r^{L}, only rightmost extremes are released during $(r^{\text{L}}, t]$, which means that Case (O) occurs whenever a new request is released during this time frame, because $\text{L}(t) > 0$. At time r^{L} only the Cases (P1), (E1), or (O) can occur, again because $\text{L}(t) > 0$. In all these cases the server will immediately move toward $\text{L}(t)$ or 0, i.e., to the left, and $\text{pos}(t) > \text{L}(t)$ implies that the server does not reach $\text{L}(t)$ by time t. Thus, the server moves uninterrupted to the left during $[r^{\text{L}}, t]$. Applying Lemma 5.3 and

$\mathrm{L}(t) < \mathrm{pos}(t)$, we get

$$|T^{\mathrm{alg}}| = r^{\mathrm{L}} + |\mathrm{pos}(r^{\mathrm{L}}) - \mathrm{L}(t)| + |\mathrm{L}(t) - \mathrm{R}(t)| \overset{\text{Lem. 5.3}}{\leq} 2|T^{\mathrm{opt}}|.$$

(3) *Case (E2) at time t*
In this case, we have $\mathrm{L}(t) \leq 0$ and $\mathrm{R}(t) \geq 0$ at time t. Hence, from time t on, the tour computed by Algorithm 2 serves $\sigma^{\mathrm{L}}(t)$ and $\sigma^{\mathrm{R}}(t)$ without interruption in this order provided no new requests are released after time t. If $\mathrm{pos}(t) \leq 0$, then we obtain using Lemma 5.3

$$|T^{\mathrm{alg}}| \leq t + |\mathrm{L}(t)| + |\mathrm{L}(t) - \mathrm{R}(t)| \leq t + (r^{\mathrm{L}} + |\mathrm{L}(t) - \mathrm{R}(t)|) \overset{\text{Lem. 5.3}}{\leq} 2|T^{\mathrm{opt}}|.$$

Thus, from now on, we suppose that $\mathrm{pos}(t) > 0$. In this case, we get

$$|T^{\mathrm{alg}}| = t + |\mathrm{pos}(t)| + |\mathrm{L}(t)| + |\mathrm{L}(t) - \mathrm{R}(t)|.$$

It is not possible to directly show ρ-competitiveness in this case. Instead, we need to take the behavior of the server during times $[r^{\mathrm{L}}, t]$ into account. We distinguish two cases. Either the server only moves to the left toward $\sigma^{\mathrm{L}}(t)$ without interruption during this interval or the server gets delayed, because it waits at the origin or moves to the right for some time. Let $p^{\mathrm{R}}(t)$ be the rightmost point of a request seen until time t. In the first case, we know that $0 < |\mathrm{pos}(t)| \leq |\mathrm{pos}(r^{\mathrm{L}})| \leq p^{\mathrm{R}}(t)$ as the server only moves left during $[r^{\mathrm{L}}, t]$ and $\mathrm{pos}(t) > 0$ by assumption. Applying Lemma 5.3 and and using $|\mathrm{pos}(r^{\mathrm{L}})| \leq |p^{\mathrm{R}}(t)|$, we obtain

$$\begin{aligned} |T^{\mathrm{alg}}| &\leq r^{\mathrm{L}} + |\mathrm{pos}(r^{\mathrm{L}})| + |\mathrm{L}(t)| + |\mathrm{L}(t) - \mathrm{R}(t)| \\ &\leq (|p^{\mathrm{R}}(t)| + |\mathrm{L}(t)|) + (r^{\mathrm{L}} + |\mathrm{L}(t) - \mathrm{R}(t)|) \overset{\text{Lem. 5.3}}{\leq} 2|T^{\mathrm{opt}}|. \end{aligned}$$

We now consider the case where the server does not move toward $\sigma^{\mathrm{L}}(t)$ without interruption during $[r^{\mathrm{L}}, t]$. For this to happen, there has to be a call of the Algorithm 2 at some time $\hat{t} \in [r^{\mathrm{L}}, t)$ such that the tour followed by the server in $[\hat{t}, t]$ waits at the origin or moves to the right for some time. We choose $\hat{t}$ maximal with this property. By Lemma 5.4, only the Cases (A2), (P2), and (E2) can occur at time $\hat{t}$. The Cases (A2) and (P2) are both treated in Lemma 5.6. Finally, Case (E2) is treated in Lemma 5.7. This concludes the proof. □

5.3 Remaining Lemmas in the Proof of Theorem 5.2

We still have to show ρ-competitiveness of Algorithm 2 in the following setting (last case of the proof of Theorem 5.2.)

Setting 3. *Consider the following setting of requests in an instance for Algorithm 2.*

- *At time t the request $\sigma^{\mathrm{R}}(t) = (\mathrm{R}(t), t)$ appears with $\mathrm{R}(t) \geq 0$ and Case (E2) occurs.*
- *At time t there is a leftmost extreme $\sigma^{\mathrm{L}}(t) = (\mathrm{L}(t), r^{\mathrm{L}})$ present with $\mathrm{L}(t) \leq 0$.*
- *The position of the server satisfies $\mathrm{pos}(t) > 0$.*
- *Time $\hat{t} \in [r^{\mathrm{L}}, t)$ is the last time Algorithm 2 is called such that the tour followed by the server in $[\hat{t}, t]$ waits at the origin or moves to the right for some time, i.e., is not identical to "move($\mathrm{L}(t)$)".*

In the next lemma, we limit the cases that can occur at time $\hat{t}$ in the setting above.

Lemma 5.4. *In Setting 3 the only cases that can occur when Algorithm 2 is called at time $\hat{t}$ are Case (A2) if $\hat{t} > r^{\mathrm{L}}$, and Cases (P2) and (E2) if $\hat{t} = r^{\mathrm{L}}$.*

Proof. We first establish the following two claims:

- *The tour computed at $\hat{t}$ does not serve the extreme request $\sigma^{\mathrm{L}}(t)$ first.*
  Aiming for a contradiction assume that $\sigma^{\mathrm{L}}(t)$ is served first. Note that by the choice of $\hat{t}$ any tour computed after $\hat{t}$ immediately moves left. If $\mathrm{pos}(\hat{t}) \leq 0$, then also $\mathrm{pos}(t) \leq 0$ holds, because the computed tour does not move right of the origin before serving $\sigma^{\mathrm{L}}(t)$ in this case. However, this contradicts $\mathrm{pos}(t) > 0$. If $\mathrm{pos}(\hat{t}) > 0$, then the server will move toward the origin first. If the server arrives at the origin and waits there during $[r^{\mathrm{L}}, t)$, then we have $\mathrm{pos}(t) \leq 0$, which is again a contradiction. If it does reach the origin or does not wait there, then the server moves toward $\mathrm{L}(t)$ during $[\hat{t}, t]$ without interruption, contradicting our assumptions about $\hat{t}$ in Setting 3.
- *Case (O) cannot occur at time $\hat{t}$.*
  As $\mathrm{L}(t) \leq 0$ holds, Case (O) would imply that $\mathrm{pos}(\hat{t}) \leq \sigma^{\mathrm{R}}(\hat{t}) < 0$. But this contradicts $\mathrm{pos}(t) > 0$, because the tour computed at $\hat{t}$ only moves to $\sigma^{\mathrm{R}}(\hat{t})$ and then left and any tour computed after $\hat{t}$ immediately moves to the left by the choice of $\hat{t}$.

First consider the case $r^{\mathrm{L}} < \hat{t} < t$. This implies that Alg is forced to divert from the direct tour toward $\sigma^{\mathrm{L}}(t)$ caused by the release of a rightmost extreme at time $\hat{t}$. The release of a leftmost extreme would contradict that $\sigma^{\mathrm{L}}(r^{\mathrm{L}})$ is still the leftmost extreme at time t. The Cases (P1) and (E1) cannot occur at time $\hat{t}$, because two different extremes exists at this point in time. By the first claim above, (P2) and (E2) cannot occur and by the second claim, Case (O) cannot occur. This leaves Case (A2) as the only possibility.

Next, let us consider the case $\hat{t} = r^{\mathrm{L}}$. This means that Algorithm 2 computes a tour at the release of $\sigma^{\mathrm{L}}(t)$ at time r^{L} that waits at the origin or moves to the right for some time. By the first claim above, Case (P1), (E1), and (A2) cannot occur. Moreover, Case (O) cannot occur by the second claim. Thus, only Case (P2) and Case (E2) are possible for $\hat{t} = r^{\mathrm{L}}$. □

Later, we will also consider the following analogous version of the setting above where the role of the leftmost and rightmost extreme are exchanged, the server is on the other side of the origin, and we again consider the time interval between the release times of both extremes.

Setting 4. *Consider the following setting of requests of an instance for Algorithm 2.*

- *At time r^{L} the request $\sigma^{\mathrm{L}}(t) = (\mathrm{L}(t), r^{\mathrm{L}})$ appears with $\mathrm{L}(t) \leq 0$ and Case (E2) occurs.*
- *At time r^{L} there is a rightmost extreme $\sigma^{\mathrm{R}}(r^{\mathrm{L}}) = (\mathrm{R}(r^{\mathrm{L}}), r^{\mathrm{R}})$ present with $\mathrm{R}(r^{\mathrm{L}}) \geq 0$.*
- *The position of the server satisfies $\mathrm{pos}(r^{\mathrm{L}}) < 0$.*
- *Time $\tilde{t} \in [r^{\mathrm{R}}, r^{\mathrm{L}})$ is the last time Algorithm 2 is called such that the tour followed by the server in $[\tilde{t}, r^{\mathrm{L}}]$ waits at the origin or moves to the left for some time, i.e., is not identical to "$\mathrm{move}(\mathrm{R}(r^{\mathrm{L}}))$."*

With a completely analogous proof, we obtain the same result as in Lemma 5.4 for the setting above.

Corollary 5.5. *In Setting 4, the cases that can occur when Algorithm 2 is called at time $\tilde{t}$ are Case (A2) when $\tilde{t} > r^{\mathrm{R}}$, and Cases (P2) and (E2) when $\tilde{t} = r^{\mathrm{R}}$.*

The following lemma now treats the Cases (P2) and (A2) at $\hat{t}$.

Lemma 5.6. *In Setting 3, if Case (P2) or Case (A2) occur at time $\hat{t}$, no additional requests appear after $\sigma^{\mathrm{R}}(t) = (\mathrm{R}(t), t)$ is released at time t and the tour computed by Algorithm 2 at $\hat{t}$ serves $\sigma^{\mathrm{R}}(\hat{t})$ before $\sigma^{\mathrm{L}}(t)$, then $|T^{\mathrm{alg}}| \leq \rho |T^{\mathrm{opt}}|$.*

Proof. By the choice of $\hat{t}$, we know that the server immediately moves toward $\mathrm{L}(t)$ without waiting if any new extreme arrives in $(\hat{t}, t]$. From then on it will serve $\sigma^{\mathrm{L}}(t)$ and afterwards $\sigma^{\mathrm{R}}(t)$

without interruption, because Case (E2) occurs at time t. Thus, the latest point in time after $\hat{t}$ at which the server starts moving to the left is when it reaches $\mathrm{R}(\hat{t})$. Hence, we have $\mathrm{pos}(t) \leq \mathrm{R}(\hat{t})$ and additionally using $\mathrm{pos}(t) > 0$, we obtain

$$\begin{aligned} |T^{\mathrm{alg}}| &\leq t + |\mathrm{pos}(t)| + 2|\mathrm{L}(t)| + |\mathrm{R}(t)| \\ &\leq t + |\mathrm{R}(\hat{t})| + 2|\mathrm{L}(t)| + |\mathrm{R}(t)|. \end{aligned} \tag{48}$$

By the definition of Cases (P2) and (A2) the server reaches $\mathrm{R}(\hat{t})$ at the latest at time $L(\sigma^{\mathrm{R}}(\hat{t}), \sigma^{\mathrm{L}}(t))$. From then on it will serve $\sigma^{\mathrm{L}}(t)$ and afterwards $\sigma^{\mathrm{R}}(t)$ without interruption. Using $L(\sigma^{\mathrm{R}}(\hat{t}), \sigma^{\mathrm{L}}(t)) \leq \rho r^{\mathrm{L}} + (\rho - 2)\mathrm{R}(\hat{t}) + (\rho - 2)\mathrm{L}(t)$, we also have the following bound:

$$\begin{aligned} |T^{\mathrm{alg}}| &\leq L(\sigma^{\mathrm{R}}(\hat{t}), \sigma^{\mathrm{L}}(t)) + |\mathrm{R}(\hat{t})| + 2|\mathrm{L}(t)| + |\mathrm{R}(t)| \\ &\leq \rho r^{\mathrm{L}} + (\rho - 1)|\mathrm{R}(\hat{t})| + \rho|\mathrm{L}(t)| + |\mathrm{R}(t)|. \end{aligned}$$

For $|\mathrm{R}(\hat{t})| \leq |\mathrm{R}(t)|$, we immediately obtain $|T^{\mathrm{alg}}| \leq \rho(r^{\mathrm{L}} + |\mathrm{L}(t)| + |\mathrm{R}(t)|) \leq \rho|T^{\mathrm{opt}}|$. Otherwise, we have $|\mathrm{R}(t)| < |\mathrm{R}(\hat{t})|$ and get

$$|T^{\mathrm{alg}}| \leq \rho(r^{\mathrm{L}} + |\mathrm{L}(t)| + |\mathrm{R}(\hat{t})|). \tag{49}$$

To show ρ-competitiveness in this case, we consider the six possible orders in which the requests $\sigma^{\mathrm{L}}(t)$, $\sigma^{\mathrm{R}}(\hat{t})$ and $\sigma^{\mathrm{R}}(t)$ can be served in an optimal tour.

- In the three cases where the optimal tour serves $\sigma^{\mathrm{L}}(t)$ before $\sigma^{\mathrm{R}}(\hat{t})$, we have $|T^{\mathrm{opt}}| \geq r^{\mathrm{L}} + |\mathrm{R}(\hat{t})| + |\mathrm{L}(t)|$ and Inequality Equation (49) ensures ρ-competitiveness.
- If the optimal tour serves $\sigma^{\mathrm{R}}(t)$ before $\sigma^{\mathrm{L}}(t)$, then we obtain $|T^{\mathrm{opt}}| \geq t + |\mathrm{R}(t)| + |\mathrm{L}(t)|$. By the bound in Equation (48), we obtain

$$\begin{aligned} |T^{\mathrm{alg}}| &\overset{(48)}{\leq} t + |\mathrm{R}(\hat{t})| + 2|\mathrm{L}(t)| + |\mathrm{R}(t)| \\ &= (t + |\mathrm{L}(t)| + |\mathrm{R}(t)|) + (|\mathrm{L}(t)| + |\mathrm{R}(\hat{t})|) \leq 2|T^{\mathrm{opt}}|. \end{aligned}$$

- In the remaining case where the optimum uses the order $\sigma^{\mathrm{R}}(\hat{t}), \sigma^{\mathrm{L}}(t), \sigma^{\mathrm{R}}(t)$, we have that $|T^{\mathrm{opt}}| \geq 2|\mathrm{R}(\hat{t})| + 2|\mathrm{L}(t)| + |\mathrm{R}(t)|$. Using Equation (48) together with $|T^{\mathrm{opt}}| \geq t$, we can conclude that

$$\begin{aligned} |T^{\mathrm{alg}}| &\overset{(48)}{\leq} t + |\mathrm{R}(\hat{t})| + 2|\mathrm{L}(t)| + |\mathrm{R}(t)| \\ &\leq t + (2|\mathrm{R}(\hat{t})| + 2|\mathrm{L}(t)| + |\mathrm{R}(t)|) \leq 2|T^{\mathrm{opt}}|. \end{aligned}$$ □

Finally, we treat the remaining Case (E2) at time $\hat{t}$.

Lemma 5.7. *In Setting 3, if Case (E2) occurs at time $\hat{t} = r^{\mathrm{L}}$ and no additional requests appear after $\sigma^{\mathrm{R}}(t) = (\mathrm{R}(t), t)$, then $|T^{\mathrm{alg}}| \leq \rho|T^{\mathrm{opt}}|$.*

Proof. By assumption Case (E2) occurs at time $r^{\mathrm{L}} = \hat{t}$ and the server immediately moves toward $\mathrm{R}(r^{\mathrm{L}})$ by Lemma 5.4. It either serves $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$ and then moves left to serve $\sigma^{\mathrm{L}}(t)$ or it moves to $\sigma^{\mathrm{L}}(t)$ earlier when a new extreme arrives in $(\hat{t}, t]$. By time t the server has not reached $\mathrm{L}(t)$ as $\sigma^{\mathrm{L}}(t)$ is still unserved at time t. Now Case (E2) occurs and the server serves $\sigma^{\mathrm{L}}(t)$ and afterwards $\sigma^{\mathrm{R}}(t)$ without interruption. We denote the release time of $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$ by r^{R}. We first consider the case where $\mathrm{pos}(r^{\mathrm{L}}) \geq 0$. Using $0 \leq \mathrm{pos}(r^{\mathrm{L}}) \leq |\mathrm{R}(r^{\mathrm{L}})| \leq r^{\mathrm{R}}$ and applying Lemma 5.3 twice, we obtain

$$\begin{aligned} |T^{\mathrm{alg}}| &\leq r^{\mathrm{L}} + 2|\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}(t)| + |\mathrm{R}(t)| \\ &\leq (r^{\mathrm{R}} + |\mathrm{R}(r^{\mathrm{L}})| + |\mathrm{L}(t)|) + (r^{\mathrm{L}} + |\mathrm{L}(t)| + |\mathrm{R}(t)|) \overset{\text{Lem. 5.3}}{\leq} 2|T^{\mathrm{opt}}|. \end{aligned}$$

So, from now on, we suppose $\text{pos}(r^{\text{L}}) < 0$. In this case, we have

$$|T^{\text{alg}}| \le |r^{\text{L}} + |\text{pos}(r^{\text{L}})| + 2|\text{R}(r^{\text{L}})| + 2|\text{L}(t)| + |\text{R}(t)| \tag{50}$$

$$\begin{aligned} &= (r^{\text{L}} + |\text{L}(t)| + |\text{R}(t)|) + (|\text{pos}(r^{\text{L}})| + 2|\text{R}(r^{\text{L}})| + |\text{L}(t)|) \\ &\overset{\text{Lem. 5.3}}{\le} |T^{\text{opt}}| + (|\text{pos}(r^{\text{L}})| + 2|\text{R}(r^{\text{L}})| + |\text{L}(t)|). \end{aligned} \tag{51}$$

The remaining proof of the lemma proceeds along the following key claims. Note that Claims 2 and 3 together show that Algorithm 2 is ρ-competitive, because either Inequality Equation (53) or Inequality Equation (54) is satisfied.

Claim 1: We can assume the following lower bound for Opt (otherwise Algorithm 2 is ρ-competitive):

$$|T^{\text{opt}}| \ge r^{\text{L}} + |\text{L}(t)| + \max(|\text{R}(r^{\text{L}})|, |\text{R}(t)|). \tag{52}$$

Claim 2: We can show $|T^{\text{alg}}| \le \rho|T^{\text{opt}}|$ provided that

$$|\text{L}(t)| \le \frac{3\rho - 5}{(2\rho - 2)(7 - 3\rho)}(\rho r^{\text{R}} + (\rho - 2)|\text{R}(r^{\text{L}})|). \tag{53}$$

Claim 3: We can show $|T^{\text{alg}}| \le \rho|T^{\text{opt}}|$ provided that

$$|\text{L}(t)| > \frac{3\rho - 5}{(2\rho - 2)(7 - 3\rho)}(\rho r^{\text{R}} + (\rho - 2)|\text{R}(r^{\text{L}})|). \tag{54}$$

The claims are proven below. This concludes the proof of the lemma. □

It remains to prove the claims made in the proof of the Lemma 5.7. We start with the auxiliary claim.

Proof of Claim 1. Suppose at first that $|\text{R}(r^{\text{L}})| > |\text{R}(t)|$. In this case, we know that by time t Algorithm 2 has served $\sigma^{\text{R}}(r^{\text{L}})$, since at that time $\sigma^{\text{R}}(t)$ is the extreme rightmost of $\text{pos}(t)$ rather than $\sigma^{\text{R}}(r^{\text{L}})$. As Case (E2) occurs at time t, we have $\text{pos}(t) \in [\text{L}(t), \text{R}(t)]$ and therefore $t \ge r^{\text{L}} + |\text{pos}(r^{\text{L}})| + |\text{R}(r^{\text{L}})| + (|\text{R}(r^{\text{L}})| - |\text{R}(t)|)$. Using this inequality and $\text{pos}(r^{\text{L}}) < 0$, we obtain

$$\begin{aligned} |T^{\text{alg}}| &\overset{(50)}{\le} r^{\text{L}} + |\text{pos}(r^{\text{L}})| + 2|\text{R}(r^{\text{L}})| + 2|\text{L}(t)| + |\text{R}(t)| \\ &= r^{\text{L}} + |\text{pos}(r^{\text{L}})| + |\text{R}(r^{\text{L}})| + (|\text{R}(r^{\text{L}})| - |\text{R}(t)|) + 2|\text{L}(t)| + 2|\text{R}(t)| \\ &\le t + 2|\text{L}(t)| + 2|\text{R}(t)| \\ &= (t + |\text{R}(t)| + |\text{L}(t)|) + (|\text{L}(t)| + |\text{R}(t)|) \\ &\le (t + |\text{R}(t)| + |\text{L}(t)|) + |T^{\text{opt}}|. \end{aligned}$$

If the optimal tour serves $\sigma^{\text{R}}(t)$ before $\sigma^{\text{L}}(t)$, then we have $|T^{\text{opt}}| \ge t + |\text{L}(t)| + |\text{R}(t)|$, which implies that $|T^{\text{alg}}| \le 2|T^{\text{opt}}|$. If, however, the optimal tour serves the requests in the order $\sigma^{\text{R}}(r^{\text{L}}), \sigma^{\text{L}}(t), \sigma^{\text{R}}(t)$, then we know that $|T^{\text{opt}}| \ge 2|\text{R}(r^{\text{L}})| + 2|\text{L}(t)| + |\text{R}(t)| \ge 2|\text{L}(t)| + 2|\text{R}(t)| + |\text{R}(r^{\text{L}})|$. This implies that

$$|T^{\text{alg}}| \le t + (2|\text{R}(t)| + 2|\text{L}(t)|) \le 2|T^{\text{opt}}|.$$

Next suppose that $|\text{R}(r^{\text{L}})| > |\text{R}(t)|$ holds. Then in the optimal tour $\sigma^{\text{L}}(t)$ is served before $\sigma^{\text{R}}(r^{\text{L}})$. Of course, it might still be the case that $|\text{R}(r^{\text{L}})| \le |\text{R}(t)|$; however, in both cases we obtain the claimed lower bound $|T^{\text{opt}}| \ge r^{\text{L}} + |\text{L}(t)| + \max(|\text{R}(r^{\text{L}})|, |\text{R}(t)|)$. □

Next, we prove the two remaining claims, establishing that the algorithm has the claimed competitive ratio.

Proof of Claim 2. Using $r^{\mathrm{R}} \geq \mathrm{R}(r^{\mathrm{L}})$, the assumption implies

$$|\mathrm{L}(t)| \leq \frac{3\rho - 5}{(2\rho - 2)(7 - 3\rho)}(\rho r^{\mathrm{R}} + (\rho - 2)|\mathrm{R}(r^{\mathrm{L}})|) \leq \frac{3\rho - 5}{7 - 3\rho} r^{\mathrm{R}}. \tag{55}$$

Note that the assumed inequality is exactly the second condition for Case (A2) at time r^{L}. Because Case (E2) occurs at time r^{L} by assumption, we know that the first condition of Case (A2) does not hold, i.e., $r^{\mathrm{L}} + |\mathrm{pos}(r^{\mathrm{L}}) - \mathrm{L}(t)| > L(\sigma^{\mathrm{L}}(t), \sigma^{\mathrm{R}}(r^{\mathrm{L}}))$. We have $r^{\mathrm{R}} \leq r^{\mathrm{L}}$ and thus $L(\sigma^{\mathrm{L}}(t), \sigma^{\mathrm{R}}(r^{\mathrm{L}})) = \rho r^{\mathrm{R}} + (\rho - 2)|\mathrm{L}(t)| + (\rho - 2)|\mathrm{R}(r^{\mathrm{L}})|$. Moreover, $|\mathrm{pos}(r^{\mathrm{L}}) - \mathrm{L}(t)| = |\mathrm{L}(t)| - |\mathrm{pos}(r^{\mathrm{L}})|$ holds, because $\mathrm{pos}(r^{\mathrm{L}}) < 0$. Hence, we obtain

$$\begin{aligned} |\mathrm{pos}(r^{\mathrm{L}})| &< r^{\mathrm{L}} + |\mathrm{L}(t)| - L(\sigma^{\mathrm{L}}(t), \sigma^{\mathrm{R}}(r^{\mathrm{L}})) \\ &= r^{\mathrm{L}} + (3 - \rho)|\mathrm{L}(t)| + (2 - \rho)|\mathrm{R}(r^{\mathrm{L}})| - \rho r^{\mathrm{R}}. \end{aligned} \tag{56}$$

Using Inequality Equation (56), $2r^{\mathrm{L}} \leq \rho r^{\mathrm{L}} + (2 - \rho)|\mathrm{L}(t)|$ and Inequality Equation (55), we obtain

$$\begin{aligned} |T^{\mathrm{alg}}| &\overset{(50)}{\leq} r^{\mathrm{L}} + |\mathrm{pos}(r^{\mathrm{L}})| + 2|\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}(t)| + |\mathrm{R}(t)| \\ &\overset{(56)}{\leq} 2r^{\mathrm{L}} + (5 - \rho)|\mathrm{L}(t)| + (4 - \rho)|\mathrm{R}(r^{\mathrm{L}})| + |\mathrm{R}(t)| - \rho r^{\mathrm{R}} \\ &\leq \rho r^{\mathrm{L}} + (7 - 3\rho)|\mathrm{L}(t)| + \rho|\mathrm{L}(t)| + (4 - \rho)|\mathrm{R}(r^{\mathrm{L}})| + |\mathrm{R}(t)| - \rho r^{\mathrm{R}} \\ &\overset{(55)}{\leq} \rho r^{\mathrm{L}} + (3\rho - 5)r^{\mathrm{R}} + \rho|\mathrm{L}(t)| + (4 - \rho)|\mathrm{R}(r^{\mathrm{L}})| + |\mathrm{R}(t)| - \rho r^{\mathrm{R}} \\ &= \rho r^{\mathrm{L}} - (5 - 2\rho)r^{\mathrm{R}} + \rho|\mathrm{L}(t)| + (4 - \rho)|\mathrm{R}(r^{\mathrm{L}})| + |\mathrm{R}(t)| \\ &\leq \rho r^{\mathrm{L}} - (5 - 2\rho)|\mathrm{R}(r^{\mathrm{L}})| + \rho|\mathrm{L}(t)| + (4 - \rho)|\mathrm{R}(r^{\mathrm{L}})| + |\mathrm{R}(t)| \\ &= \rho r^{\mathrm{L}} + (\rho - 1)|\mathrm{R}(r^{\mathrm{L}})| + \rho|\mathrm{L}(t)| + |\mathrm{R}(t)| \\ &\leq \rho\left(r^{\mathrm{L}} + |\mathrm{L}(t)| + \max(|\mathrm{R}(r^{\mathrm{L}})|, |\mathrm{R}(t)|)\right) \\ &\overset{(52)}{\leq} \rho|T^{\mathrm{opt}}|. \end{aligned}$$

This completes the proof of the claim. □

Proof of Claim 3. To show $|T^{\mathrm{alg}}| \leq \rho|T^{\mathrm{opt}}|$ in this case, we have to take the situation during $[r^{\mathrm{R}}, r^{\mathrm{L}}]$ into account. Let us first suppose that during this interval Algorithm 2 moves toward $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$ without interruption. In particular, this implies that $\mathrm{pos}(r^{\mathrm{R}}) < \mathrm{pos}(r^{\mathrm{L}}) < 0$. Let us additionally assume that the following inequality holds:

$$|\mathrm{pos}(r^{\mathrm{R}})| \leq (2\rho - 2)|\mathrm{L}(t)| - (3 - \rho)|\mathrm{R}(r^{\mathrm{L}})| - r^{\mathrm{R}}. \tag{57}$$

Recall that at time r^{L} Case (E2) occurs, $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$ is still the rightmost extreme and the server continues moving toward $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$. The server then moves toward $\sigma^{\mathrm{L}}(t)$ at the latest by the time it reaches $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$. From then on it serves $\sigma^{\mathrm{L}}(t)$ and then $\sigma^{\mathrm{R}}(t)$ without interruption. Using Inequality Equation (57) and Claim 1, we therefore obtain

$$\begin{aligned} |T^{\mathrm{alg}}| &\leq r^{\mathrm{R}} + |\mathrm{pos}(r^{\mathrm{R}})| + 2|\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}(t)| + |\mathrm{R}(t)| \\ &\overset{(57)}{\leq} (2\rho - 2)|\mathrm{L}(t)| - (3 - \rho)|\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{L}(t)| + |\mathrm{R}(t)| \\ &= 2\rho|\mathrm{L}(t)| + (\rho - 1)|\mathrm{R}(r^{\mathrm{L}})| + |\mathrm{R}(t)| \\ &\leq \rho|r^{\mathrm{L}}| + \rho|\mathrm{L}(t)| + \rho\max(|\mathrm{R}(t)|, |\mathrm{R}(r^{\mathrm{L}})|) \\ &\overset{(52)}{\leq} \rho|T^{\mathrm{opt}}|. \end{aligned}$$

Thus, it is sufficient to show that Inequality Equation (57) is satisfied. Using Inequality Equation (54), Lemma 5.1, $|\mathrm{R}(r^{\mathrm{L}})| \le r^{\mathrm{R}}$ and the definition of ρ, we get

$$\begin{aligned}&(2\rho - 2)|\mathrm{L}(t)| - (3-\rho)|\mathrm{R}(r^{\mathrm{L}})| - r^{\mathrm{R}}\\ &\overset{(54)}{>} \frac{3\rho-5}{7-3\rho}(\rho r^{\mathrm{R}} + (\rho-2)|\mathrm{R}(r^{\mathrm{L}})|) - (3-\rho)|\mathrm{R}(r^{\mathrm{L}})| - r^{\mathrm{R}}\\ &= \frac{3\rho^2-2\rho-7}{7-3\rho}r^{\mathrm{R}} - \frac{11-5\rho}{7-3\rho}|\mathrm{R}(r^{\mathrm{L}})|.\end{aligned}$$

The latter term is at least

$$\begin{aligned}&\ge \frac{3\rho^2-2\rho-7}{7-3\rho}r^{\mathrm{R}} - \frac{11-5\rho}{7-3\rho}r^{\mathrm{R}}\\ &= \frac{3\rho^2+3\rho-18}{7-3\rho}r^{\mathrm{R}}\\ &\overset{\text{Lem. 5.1}}{\ge} \frac{3\rho^2+3\rho-18}{7-3\rho}\cdot\frac{-3\rho^2+9\rho-4}{3\rho-5}|\mathrm{pos}(r^{\mathrm{R}})|.\end{aligned}$$

Hence, it suffices to show

$$\frac{3\rho^2+3\rho-18}{7-3\rho}\cdot\frac{-3\rho^2+9\rho-4}{3\rho-5} \ge 1,$$

which is equivalent to $-9\rho^4 + 18\rho^3 + 78\rho^2 - 210\rho + 107 \ge 0$. The left-hand side is zero by the definition of ρ. This completes the case that the server moves constantly toward $\mathrm{R}(r^{\mathrm{L}})$ during $[r^{\mathrm{R}}, r^{\mathrm{L}}]$.

Next, we consider the case in which the server does not constantly move toward $\mathrm{R}(r^{\mathrm{L}})$ during $[r^{\mathrm{R}}, r^{\mathrm{L}}]$. For this to happen, there has to be a call of Algorithm 2 at some time $\tilde{t} \in [r^{\mathrm{R}}, r^{\mathrm{L}})$ such that the tour followed by the server in $[\tilde{t}, r^{\mathrm{L}})$ waits at the origin or goes to the left for some time. We choose $\tilde{t}$ maximal with this property. This setting is stated in Setting 4 and analogous to the Setting 3 we considered before. By Corollary 5.5, we know that only Case (A2) can occur when $\tilde{t} > r^{\mathrm{R}}$ and Cases (P2) and (E2) can occur at time $\tilde{t} = r^{\mathrm{R}}$.

We first establish that in all three cases the following inequality holds:

$$r^{\mathrm{L}} + |\mathrm{pos}(r^{\mathrm{L}}) - \mathrm{L}(\tilde{t})| \ge L(\sigma^{\mathrm{L}}(\tilde{t}), \sigma^{\mathrm{R}}(r^{\mathrm{L}})). \tag{58}$$

Let $f(\tau) := \tau + |\mathrm{pos}(\tau) - \mathrm{L}(\tilde{t})|$, then we need to show $f(r^{\mathrm{L}}) \ge L(\sigma^{\mathrm{L}}(\tilde{t}), \sigma^{\mathrm{R}}(r^{\mathrm{L}}))$. Observe that the function f is monotonously increasing in τ, because the server moves with at most unit speed. Thus, if $f(t') \ge L(\sigma^{\mathrm{L}}(\tilde{t}), \sigma^{\mathrm{R}}(r^{\mathrm{L}}))$ holds for some $t' < r^{\mathrm{L}}$, then Inequality Equation (58) follows. If Case (E2) occurs at time $\tilde{t} = r^{\mathrm{R}}$, then $f(r^{\mathrm{R}}) > L(\sigma^{\mathrm{L}}(\tilde{t}), \sigma^{\mathrm{R}}(r^{\mathrm{L}}))$, because the condition for Case (P2) is not satisfied. Thus, we get Inequality Equation (58) in Case (E2) by monotonicity. In the Cases (P2) and (A2), we know that the tour computed at time $\tilde{t}$ by Algorithm 2 serves $\sigma^{\mathrm{L}}(\tilde{t})$ first and then $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$. More precisely, the computed tour is

$$T_0.\,\mathrm{until}(\tau + |\mathrm{pos}(\tau) - \mathrm{L}(\tilde{t})| = L(\sigma^{\mathrm{L}}(\tilde{t}), \sigma^{\mathrm{R}}(r^{\mathrm{L}}))) \oplus \mathrm{move}(\mathrm{L}(\tilde{t})) \oplus \mathrm{move}(\mathrm{R}(r^{\mathrm{L}})).$$

If the until-condition becomes satisfied at a time $t' \le r^{\mathrm{L}}$, i.e., $f(t') = L(\sigma^{\mathrm{L}}(\tilde{t}), \sigma^{\mathrm{R}}(r^{\mathrm{L}}))$, then Inequality Equation (58) holds by the monotonicity of f. Otherwise, there is a request appearing before the until-condition is satisfied. Let t' the first time after $\tilde{t}$ at which an extreme appears. By the choice of $\tilde{t}$, we know that the server will immediately move toward $\mathrm{R}(r^{\mathrm{L}})$ at t'. Thus, if the server is waiting at the origin at t', i.e., $\mathrm{pos}(t') = 0$, then we have $\mathrm{pos}(r^{\mathrm{L}}) \ge 0$ contradicting that $\mathrm{pos}(r^{\mathrm{L}}) < 0$ by assumption. Next suppose that the server is still moving toward the origin at time t'. If additionally $\mathrm{pos}(t') < 0$ holds, then the server moves continuously to the right in $[\tilde{t}, r^{\mathrm{L}}]$ contradicting the choice of $\tilde{t}$. If $\mathrm{pos}(t') > 0$ holds, then we also get $\mathrm{pos}(r^{\mathrm{L}}) > 0$, because the server moves to the right from time t' onwards and does not reach $\sigma^{\mathrm{R}}(r^{\mathrm{L}})$ as this request is still the rightmost extreme at r^{L}. Thus, the until-condition must become satisfied as all other cases were contradictions.

We have shown that Inequality Equation (58) holds in any of the three Cases (A2), (P2), and (E2). Now, we analyze the cases $\tilde{t} > r^{\mathrm{R}}$ (Case (A2)) and $\tilde{t} = r^{\mathrm{R}}$ (Cases (P2) and (E2)) separately.

(1) *Case (A2) at time $\tilde{t} > r^{\mathrm{R}}$.*
Note that we cannot use the same argumentation as in Lemma 5.6, because there we rely on the fact that after t (here it would be after r^{L}) no new requests appear.
We have $r^{\mathrm{R}} < \tilde{t}$ and $r^{\mathrm{R}} \geq |\mathrm{R}(r^{\mathrm{L}})|$, and therefore

$$\begin{aligned} L(\sigma^{\mathrm{L}}(\tilde{t}), \sigma^{\mathrm{R}}(r^{\mathrm{L}})) &= \rho r^{\mathrm{R}} + (\rho - 2)|\mathrm{L}(\tilde{t})| + (\rho - 2)|\mathrm{R}(r^{\mathrm{L}})| \\ &\geq (\rho - 2)|\mathrm{L}(\tilde{t})| + (2\rho - 2)|\mathrm{R}(r^{\mathrm{L}})|. \end{aligned} \tag{59}$$

Using $|\mathrm{pos}(r^{\mathrm{L}}) - \mathrm{L}(\tilde{t})| = |\mathrm{L}(\tilde{t})| - |\mathrm{pos}(r^{\mathrm{L}})|$, since $\mathrm{L}(\tilde{t}) \leq \mathrm{pos}(r^{\mathrm{L}}) < 0$, we can simplify Inequality Equation (58) as follows:

$$\begin{aligned} |\mathrm{pos}(r^{\mathrm{L}})| &\leq r^{\mathrm{L}} + |\mathrm{L}(\tilde{t})| - L(\sigma^{\mathrm{L}}(\tilde{t}), \sigma^{\mathrm{R}}(r^{\mathrm{L}})) \\ &\overset{(59)}{\leq} r^{\mathrm{L}} + (3 - \rho)|\mathrm{L}(\tilde{t})| + (2 - 2\rho)|\mathrm{R}(r^{\mathrm{L}})|. \end{aligned} \tag{60}$$

As Case (A2) occurred at time $\tilde{t}$, we know that

$$|\mathrm{L}(\tilde{t})| \leq \frac{3\rho - 5}{(2\rho - 2)(7 - 3\rho)}(\rho r^{\mathrm{R}} + (\rho - 2)|\mathrm{R}(r^{\mathrm{L}})|) \leq \frac{3\rho - 5}{7 - 3\rho} r^{\mathrm{R}}. \tag{61}$$

By Inequality Equation (54), this implies that $|\mathrm{L}(\tilde{t})| < |\mathrm{L}(t)|$. Using the last two statements together with Equation (60) and Claim 1, we obtain

$$\begin{aligned} &|\mathrm{pos}(r^{\mathrm{L}})| + 2|\mathrm{R}(r^{\mathrm{L}})| \\ \overset{(60)}{\leq}& r^{\mathrm{L}} - \rho r^{\mathrm{R}} + (3 - \rho)|\mathrm{L}(\tilde{t})| + (4 - \rho)|\mathrm{R}(r^{\mathrm{L}})| \\ =& r^{\mathrm{L}} - \rho r^{\mathrm{R}} + (7 - 3\rho)|\mathrm{L}(\tilde{t})| + (2\rho - 4)|\mathrm{L}(\tilde{t})| + (4 - \rho)|\mathrm{R}(r^{\mathrm{L}})| \\ \overset{(61)}{\leq}& r^{\mathrm{L}} - \rho r^{\mathrm{R}} + (7 - 3\rho)\frac{3\rho - 5}{7 - 3\rho} r^{\mathrm{R}} + (2\rho - 4)|\mathrm{L}(\tilde{t})| + (4 - \rho)|\mathrm{R}(r^{\mathrm{L}})| \\ \leq& r^{\mathrm{L}} - \rho r^{\mathrm{R}} + (7 - 3\rho)\frac{3\rho - 5}{7 - 3\rho} r^{\mathrm{R}} + (2\rho - 4)|\mathrm{L}(t)| + (4 - \rho)|\mathrm{R}(r^{\mathrm{L}})| \\ =& r^{\mathrm{L}} - (5 - 2\rho) r^{\mathrm{R}} + (2\rho - 4)|\mathrm{L}(t)| + (4 - \rho)|\mathrm{R}(r^{\mathrm{L}})| \\ \leq& r^{\mathrm{L}} - (5 - 2\rho)|\mathrm{R}(r^{\mathrm{L}})| + (2\rho - 4)|\mathrm{L}(t)| + (4 - \rho)|\mathrm{R}(r^{\mathrm{L}})| \\ =& r^{\mathrm{L}} + (\rho - 2)|\mathrm{L}(t)| + (\rho - 1)|\mathrm{L}(t)| + (\rho - 1)|\mathrm{R}(r^{\mathrm{L}})| - |\mathrm{L}(t)| \\ \leq& ((\rho - 1) r^{\mathrm{L}} + (\rho - 1)|\mathrm{L}(t)| + (\rho - 1)|\mathrm{R}(r^{\mathrm{L}})|) - |\mathrm{L}(t)| \\ \overset{(52)}{\leq}& (\rho - 1)|T^{\mathrm{opt}}| - |\mathrm{L}(t)|. \end{aligned}$$

This implies $|T^{\mathrm{alg}}| \leq \rho|T^{\mathrm{opt}}|$ by Inequality Equation (51).

(2) *Cases (P2) or (E2) at time $r^{\mathrm{R}} = \tilde{t}$*

Using $r^{\mathrm{R}} \geq |\mathrm{L}(r^{\mathrm{R}})|$, $r^{\mathrm{R}} \geq |\mathrm{R}(r^{\mathrm{L}})|$ and $\rho \geq 2$, we can bound $L(\sigma^{\mathrm{L}}(r^{\mathrm{R}}), \sigma^{\mathrm{R}}(r^{\mathrm{L}}))$ as follows

$$\begin{aligned} L(\sigma^{\mathrm{L}}(r^{\mathrm{R}}), \sigma^{\mathrm{R}}(r^{\mathrm{L}})) &= \min\{\rho r^{\mathrm{L}}(r^{\mathrm{R}}) + (\rho-1)|\mathrm{L}(r^{\mathrm{R}})| + (\rho-1)|\mathrm{R}(r^{\mathrm{L}})|, \\ &\qquad \rho r^{\mathrm{R}} + (\rho-2)|\mathrm{L}(r^{\mathrm{R}})| + (\rho-2)|\mathrm{R}(r^{\mathrm{L}})|\} \\ &\geq \min\{|\mathrm{L}(r^{\mathrm{R}})| + |\mathrm{R}(r^{\mathrm{L}})|, \rho r^{\mathrm{R}}\} \\ &\geq |\mathrm{L}(r^{\mathrm{R}})| + |\mathrm{R}(r^{\mathrm{L}})|. \end{aligned} \tag{62}$$

Recall that $\mathrm{L}(r^{\mathrm{R}}) \leq \mathrm{pos}(r^{\mathrm{L}}) < 0$ and hence $|\mathrm{pos}(r^{\mathrm{L}}) - \mathrm{L}(r^{\mathrm{R}})| = |\mathrm{L}(r^{\mathrm{R}})| - |\mathrm{pos}(r^{\mathrm{L}})|$. This means that Inequality Equation (58) for $\tilde{t} = r^{\mathrm{R}}$ can be simplified to $r^{\mathrm{L}} + |\mathrm{L}(r^{\mathrm{R}})| - |\mathrm{pos}(r^{\mathrm{L}})| \geq L(\sigma^{\mathrm{L}}(r^{\mathrm{R}}), \sigma^{\mathrm{R}}(r^{\mathrm{L}}))$. Applying Inequality Equation (62) above yields

$$\begin{aligned} |\mathrm{pos}(r^{\mathrm{L}})| &\leq r^{\mathrm{L}} + |\mathrm{L}(r^{\mathrm{R}})| - L(\sigma^{\mathrm{L}}(r^{\mathrm{R}}), \sigma^{\mathrm{R}}(r^{\mathrm{L}})) \\ &\overset{(62)}{\leq} r^{\mathrm{L}} + |\mathrm{L}(r^{\mathrm{R}})| - |\mathrm{L}(r^{\mathrm{R}})| - |\mathrm{R}(r^{\mathrm{L}})| = r^{\mathrm{L}} - |\mathrm{R}(r^{\mathrm{L}})|. \end{aligned} \tag{63}$$

Finally, using Inequality Equation (51) and Lemma 5.3, we obtain

$$\begin{aligned} |T^{\mathrm{alg}}| &\overset{(51)}{\leq} |T^{\mathrm{opt}}| + |\mathrm{pos}(r^{\mathrm{L}})| + 2|\mathrm{R}(r^{\mathrm{L}})| + |\mathrm{L}(t)| \\ &\overset{(63)}{\leq} |T^{\mathrm{opt}}| + r^{\mathrm{L}} - |\mathrm{R}(r^{\mathrm{L}})| + 2|\mathrm{R}(r^{\mathrm{L}})| + |\mathrm{L}(t)| \\ &= |T^{\mathrm{opt}}| + r^{\mathrm{L}} + |\mathrm{R}(r^{\mathrm{L}})| + |\mathrm{L}(t)| \\ &\leq 2|T^{\mathrm{opt}}|. \end{aligned}$$

This concludes the proof of the claim. □

This finishes the proof of Theorem 5.2.

6 ONLINE DIAL-A-RIDE ON THE LINE

In this section, we give a $(1+\sqrt{2})$-competitive algorithm for (preemptive) open online Dial-a-Ride on the line. Our algorithm repeatedly computes a tour for serving a set of requests optimally and executes it with some delay. We adapt it for both the capacitated and the uncapacitated version of our problem.

6.1 The capacitated case

The following lemma allows us to use optimal tours that do not negatively affect future tours.

Lemma 6.1. *Among the optimal tours to serve any set of requests S, there exists one in which the server never moves any request away from its destination.*

Proof. Consider a feasible tour T. We modify T to a tour T' in which the server never moves any request away from its destination, without changing its makespan. In T' the server moves identically to T, but we change the times at which any request σ is picked up and dropped off, respectively. Let $(\ell_1, r_1), (\ell_2, r_2), \ldots$ be the maximum time intervals in increasing order during which, according to T, σ is closer to its destination than at any previous point in time (thus, during these time periods the server transports σ toward its destination). In T', for any i, the server picks up σ at time ℓ_i and drops it off at time r_i. Note that, by induction, σ is indeed at the position of the server at each time ℓ_i. Further, in T', the server delivers each request at the same time as in T, so that T' has the same makespan as T. Note that, at every point in time, in T', the server carries a subset of the requests that it carries in T, thus T' obeys the capacity constraint. □

To serve any set S of requests, there thus exists an optimal tour starting at 0 in which the server never moves requests away from their destinations. We denote an arbitrary such tour by T_S^{opt}. It could be computed, e.g., using an algorithm given by Guan [16]. Further, at any time t, we let $R(t)$ denote the set of released but not yet delivered requests.

Our algorithm works as follows (cf. Algorithm 3): The server stays at position 0 until the first request arrives. Whenever a new request arrives at some time t, the server stops its current tour and unloads all requests that it was carrying. The corresponding operation is denoted by "unload" in the following, and it formally results in changing the source of each unloaded request to $\text{pos}(t)$. The server then returns to 0 and stays there until time $\sqrt{2} \cdot |T^{\text{opt}}_{R(t)}|$. Importantly, we will prove that the server can always be back at the origin by this time. Then, the server starts the tour $T^{\text{opt}}_{R(t)}$.

ALGORITHM 3: For the open online DIAL-A-RIDE problem with $c \geq 0$ fixed.

```
▷ This function is called upon receiving a new request.
Input: Current time $t$, current position $\text{pos}(t)$
       Unserved requests $R(t)$, new request $\sigma$
Output: An open tour starting at $\text{pos}(t)$ and serving all requests in $R(t)$
return unload ⊕ move(0) ⊕ waituntil($\sqrt{2} \cdot |T^{\text{opt}}_{R(t)}|$) ⊕ $T^{\text{opt}}_{R(t)}$
```

Toward the analysis, for any time t, define $R^0(t)$ to be the set of requests in $R(t)$ with their original sources and destinations (before being moved). Note that, by Lemma 6.1, each tour serving $R^0(t)$ is a tour serving $R(t)$, and thus the following fact holds.

FACT 2. *At every time t during the execution of Algorithm 3, we have $|T^{\text{opt}}_{R(t)}| \leq |T^{\text{opt}}_{R^0(t)}|$.*

We are now ready to show the competitiveness result.

THEOREM 6.2. *Algorithm 3 is $(1 + \sqrt{2}) \approx 2.41$-competitive for the preemptive open online DIAL-A-RIDE problem with capacity $c \geq 1$.*

PROOF. Let $|T^{\text{opt}}|$ be the length of an optimal tour and $|T^{\text{alg}}|$ be the length of the tour produced by our algorithm. Suppose that the server can always return to 0 until time $\sqrt{2} \cdot |T^{\text{opt}}_{R(r_i)}|$ upon receiving a new request σ_i. Then, after the last request σ_n is received, the server stays at 0 until time $\sqrt{2} \cdot |T^{\text{opt}}_{R(t_n)}|$ and then finishes all requests within time $|T^{\text{opt}}_{R(t_n)}|$. Thus, $|T^{\text{alg}}| \leq \sqrt{2} \cdot |T^{\text{opt}}_{R(t_n)}| + |T^{\text{opt}}_{R(t_n)}|$. Using Fact 2, we have $|T^{\text{opt}}_{R(t_n)}| \leq |T^{\text{opt}}_{R^0(t_n)}| \leq |T^{\text{opt}}|$, and the algorithm is $(1 + \sqrt{2})$-competitive, as claimed.

It remains to show that the server can always return to 0 until time $\sqrt{2} \cdot |T^{\text{opt}}_{R(r_i)}|$ when receiving a new request σ_i at time r_i. We prove this by induction on the number i of released requests. Let t^* be the first time when the server left 0. Then, the statement clearly holds if $r_i \leq t^*$, and, in particular, if $i = 1$. Now consider a request σ_i with $i \geq 2$, $r_i > t^*$, and suppose that, at time t_j the server would still have been able to return to 0 by time $\sqrt{2} \cdot |T^{\text{opt}}_{R(t_j)}|$, for all $j \in \{1, \ldots, i-1\}$. As an optimal tour $T^{\text{opt}}_{R(r_i)}$ over the set of requests $R(r_i)$ cannot finish before all requests are released, we have $|T^{\text{opt}}_{R(r_i)}| \geq r_i$. For the sake of contradiction, suppose the server cannot return to the origin until time $\sqrt{2} \cdot |T^{\text{opt}}_{R(r_i)}| \geq \sqrt{2} \cdot r_i$. This implies that the distance $|\text{pos}(r_i)|$ is larger than $\sqrt{2} \cdot r_i - r_i$. To reach position $\text{pos}(r_i)$ at time r_i, the server cannot have been at 0 after time $r_i - |\text{pos}(r_i)| = r_i - (\sqrt{2} \cdot r_i - r_i) = 2r_i - \sqrt{2} \cdot r_i$. Consider the last request σ_j before σ_i for which the server started to actually execute the tour $T^{\text{opt}}_{R(t_j)}$ (such a request exists, because $r_i > t^*$). By induction, the server was at 0 at time $\sqrt{2} \cdot |T^{\text{opt}}_{R(t_j)}|$, and from before, we know that it cannot have been at 0 after time $2t_i - \sqrt{2} \cdot r_i$. We thus have $\sqrt{2}|T^{\text{opt}}_{R(t_j)}| \leq 2r_i - \sqrt{2}r_i$, or

$$\left|T^{\text{opt}}_{R(t_j)}\right| \leq \sqrt{2}r_i - r_i. \tag{64}$$

By the choice of σ_j, at time r_i, the algorithm is either following $T^{\text{opt}}_{R(t_j)}$, moving back to the origin, or already waiting there. In either case, we know that $T^{\text{opt}}_{R(t_j)}$ visits $\text{pos}(r_i)$, which implies $|T^{\text{opt}}_{R(t_j)}| \geq |\text{pos}(r_i)|$. However, by our assumption above, we have $|\text{pos}(r_i)| > \sqrt{2} \cdot r_i - r_i$, which contradicts Equation (64). □

Note that the proof of Theorem 6.2 does not use that we are in the open variant of Dial-A-Ride, and, consequently, carries over to the closed variant (we omitted this in the statement of Theorem 6.2, since better bounds are known for this case; see Table 1).

6.2 The uncapacitated case

Using the above techniques, we prove that a slightly modified version of Algorithm 3 works in the uncapacitated case, even if the server does not move on a line but in the Euclidean space M with Euclidean metric $\|\cdot\|$. The server moves again with unit speed, i.e., $\|\text{pos}(t) - \text{pos}(t')\| \leq \|t - t'\|$ for all $t, t' \geq 0$. A series of requests $\sigma_1, \ldots, \sigma_n$ arrives over time with $\sigma_i = (a_i, b_i; r_i)$, where $r_i \geq 0$ denotes the release time of the request and $a_i, b_i \in M$ denote its source and destination position, respectively. The other preconditions remain the same.

The algorithm differs from Algorithm 3 in that currently loaded requests are not unloaded when a new request appears. In terms of notation, whenever a request is picked up, we keep it in $R(t)$ for future t, but we change its source to 0. We provide a formal description in Algorithm 4.

ALGORITHM 4: For uncapacitated open/closed online Dial-A-Ride.

```
▷ This function is called upon receiving a new request.
Input: Current time $t$, current position $\text{pos}(t)$
         Unserved requests $R(t)$, new request $\sigma$
Output: A tour starting at $\text{pos}(t)$ and serving all requests in $R(t)$
return move(0) $\oplus$ waituntil($\sqrt{2} \cdot |T^{\text{opt}}_{R(t)}|$) $\oplus$ $T^{\text{opt}}_{R(t)}$
```

To analyze this algorithm, we need a fact analogous to Fact 2. Recall that $R^0(t)$ is the set of requests in $R(t)$ with their original sources and destinations (before being moved). Again, we use that each tour serving $R^0(t)$ serves $R(t)$ as well.

Fact 3. *When we run Algorithm 4, for each t, we have $|T^{\text{opt}}_{R(t)}| \leq |T^{\text{opt}}_{R^0(t)}|$.*

We now get a competitive result that is analogous Theorem 6.2.

Theorem 6.3. *Algorithm 4 is $(1 + \sqrt{2}) \approx 2.41$-competitive for the uncapacitated, non-preemptive open or closed online Dial-A-Ride problem in Euclidean space.*

Proof. This proof is identical to the proof of Theorem 6.2 if we replace Fact 2 by Fact 3 and $|\cdot|$ by $\|\cdot\|$ throughout. Recall that the proof of Theorem 6.2 carries over to the closed variant (cf. note below the proof). □

Remark 1. Note that Algorithm 3 relies on the observation (Lemma 6.1 and Fact 2) that we can drop off requests anywhere closer to their destinations without increasing the length of an optimum tour. For Euclidean space this is no longer true, as illustrated by the following example in $\mathbb{R}^2$: Let $\sigma_1 = ((1,2),(1,0);0)$, $\sigma_2 = ((0,0),(1,2);t)$, $\sigma_3 = ((1,2),(2,1);t)$, $\sigma_4 = ((2,1),(1,0);t)$ and $\sigma_5 = ((0,0),(0,0);t + \sqrt{5} + 2\sqrt{2} + 1)$ for some $t \geq 0$. Clearly, waiting until time t and then following the trajectory $(0,0) \to (1,2) \to (2,1) \to (1,0) \to (0,0)$ yields an optimum (open or closed) tour of length $t + \sqrt{5} + 2\sqrt{2} + 1$ for the original set of requests. Now suppose that we started to serve σ_1 at some point before t and dropped it off at $(1,1)$, i.e., along the shortest connection between $(1,2)$

and $(1,0)$, thus modifying σ_1 to $\sigma_1' = ((1,1),(1,0);0)$. This increases the length of an optimum tour starting at time t, e.g., $(0,0) \to (1,2) \to (2,1) \to (1,1) \to (1,0) \to (0,0)$, to $t + \sqrt{5} + \sqrt{2} + 3$. Algorithm 4 avoids this issue by never dropping off requests before reaching their destinations.

We also provide a lower bound for non-preemptive closed Dial-A-Ride on the line that improves the lower bound of 1.70 from Reference [3].

Theorem 6.4. *No algorithm for the non-preemptive closed Dial-A-Ride problem on the line with fixed capacity $c \geq 1$ has competitive ratio lower than $\rho = 1.75$.*

Proof. Consider any ρ-competitive online algorithm Alg. We define an instance with three types of requests $\sigma^{(0)} = (0,0;1)$, $\sigma_i^{(1)} = (t,0;t)$, $\sigma_i^{(2)} = (-t,0;t)$, where $i = 1,\dots,c$ and $t \geq 1$ is the time when Alg serves $\sigma^{(0)}$. First, suppose that Alg serves the c requests $\sigma_1^{(1)},\dots,\sigma_c^{(1)}$ not together in one tour from t to 0. As Alg can pick up any $\sigma_i^{(1)}$ at the earliest at time $2t$ and Alg has to return to t to pick up the remaining requests $\sigma_i^{(1)}$ it did not bring to 0 in the first trip, we have $|T^{\text{alg}}| \geq 7t$ in this case. We have $|T^{\text{opt}}| = 4t$ and hence obtain $|T^{\text{alg}}|/|T^{\text{opt}}| \geq 7/4$ in this case. An analogous argument shows that Alg has to take the c requests $\sigma_1^{(2)},\dots,\sigma_c^{(2)}$ together to the origin or we have $|T^{\text{alg}}|/|T^{\text{opt}}| \geq 7/4$.

Thus, from now on, we assume that Alg picks up all c requests $\sigma_1^{(j)},\dots,\sigma_c^{(j)}$ before going to the origin for $j = 1,2$. Let $t' \geq 2t$ be the first time when Alg picks up all c requests $\sigma_i^{(1)}$ or all c requests $\sigma_i^{(2)}$. We have $|T^{\text{opt}}| = 4t$ and $|T^{\text{alg}}| \geq t' + 3t$. If $t' \geq 4t$, then $\rho \geq |T^{\text{alg}}|/|T^{\text{opt}}| \geq 7/4$ as claimed.

Otherwise, without loss of generality, we assume that the c requests $\sigma_i^{(1)}$ are picked up before the c requests $\sigma_i^{(2)}$. We introduce a new request $\sigma^{(3)} = (t,t;t' + 1/7)$. For the new instance, we have $|T^{\text{alg}}| \geq t' + 5t$, since at time $t' + 1/7$ Alg still needs to return to the origin to finish serving the c requests it is currently transporting (no preemption), then serve the remaining requests $\sigma_i^{(2)}$, $\sigma^{(3)}$, and finally return to the origin. If $t' \leq 3t - 1/7$, then we have $|T^{\text{opt}}| = 4t$ and, using $t' \geq 2t$, we get

$$\rho \geq \frac{|T^{\text{alg}}|}{|T^{\text{opt}}|} \geq \frac{t' + 5t}{4t} \geq \frac{7}{4},$$

as claimed. Now if $t' > 3t - 1/7$, then $|T^{\text{opt}}| = t' + 1/7 + t$ and $\rho \geq (t' + 5t)/(t' + 1/7 + t)$, which is monotonically decreasing in t' for $t, t' \geq 1$. Since we have $t' < 4t$ from above, we get $\rho > 9/(5 + 1/7) = 1.75$. □

7 THE OFFLINE PROBLEM

Psaraftis et al. [26] show that open offline TSP on the line with release dates can be solved in quadratic time. For the closed variant they claim that the optimal tour has the structure [26, pp. 215–216]:

$$\text{waituntil}(t) \oplus \text{move}(p^{\text{R}}) \oplus \text{move}(p^{\text{L}}) \oplus \text{move}(0)$$

or

$$\text{waituntil}(t) \oplus \text{move}(p^{\text{L}}) \oplus \text{move}(p^{\text{R}}) \oplus \text{move}(0).$$

Here the waiting time t at the origin is chosen maximally such that all requests are still served. We contradict this claim by showing that an optimal server tour may need to turn around arbitrarily many times.

Theorem 7.1. *For every $k \in \mathbb{N}$, there is an instance of closed TSP on the line such that any optimal solution turns around at least $2k$ times.*

Proof. We analyze the following instance consisting of $2k + 1$ requests with $M := 2k(k + 1)$.

$$\sigma_i = (a_i; r_i) \text{ with } a_i = 0, 1, -1, 2, -2, 3, -3, \ldots, k, -k$$
$$\text{and } r_i = M, M - 1, M - 3, M - 6, M - 10, M - 15, \ldots, k.$$

We show this instance has a unique optimal server tour with $2k$ turnarounds. Observe first, that the difference between two consecutive release times $r_i - r_{i-1}$ is exactly the travel time of the server between the two requested positions $|a_i - a_{i-1}|$. Now consider the server tour T serving each of the requests exactly at its release time, which is obviously feasible and also optimal, as it ends at time $M = 2k(k + 1)$, which is the release time of the request to position 0. By construction, the server alternatively serves requests left and right of position 0 and thus turns around $2k$ times.

Now suppose there is another optimal server tour T'. Then T' must serve the request at position 0 at time M, as this is its release time as well as the makespan of T'. As we observed $r_i - r_{i-1} = |a_i - a_{i-1}|$, the tour T' also serves the previous request exactly at its release time. Iteratively the same must hold for all other positions. This shows any request is served exactly at its release time and hence, the tour is exactly the one described above. □

Next, we derive a dynamic program that solves closed offline TSP on the line in quadratic time. It is inspired by the one of Psaraftis et al. [26] for the open variant and a dynamic program of Tsitsiklis [28] for the same problem with deadlines instead of release times. To describe the dynamic program for the closed offline TSP problem on the line, we make the following non-restrictive assumptions on the requests:

- Each position is requested at most once, as several requests to the same position can all be fulfilled when the one with the largest release time is served.
- There is a request $\sigma_0 = (0; 0)$.
- The requests $\sigma_i = (a_i; r_i)$ are labeled by increasing position, i.e., $a_i > a_{i-1}$, and requests with $a_i < 0$ have negative indices, while the other ones have a positive index. This yields a sequence $a_{-\ell} < \cdots < a_{-1} < a_0 = 0 < a_1 < \cdots < a_r$.

Our dynamic program (Algorithm 5) relies on the fact that an optimal server tour has a "zigzag shape" with decreasing amplitude. This holds, because we can assume any request is served the last time the server visits this position. Then, an increase in amplitude means that on the smaller amplitude part of the tour no request is served and thus this part of the tour can be omitted.

We compute for each index pair $-\ell - 1 \leq i < j \leq r + 1$ the completion time of two tours. We use $C^+_{i,j}$ for the best tour serving all requests σ_k with $k \leq i$ or $k \geq j$ and ending at position a_j. This means the tour serves all requests to position a_j and larger positions as well as all requests to position a_i and smaller positions. The time $C^-_{i,j}$ is the completion time of the best tour that serves the same set of requests and ends at position a_i.

We start by considering the two request sets $\{\sigma_{-\ell}\}$ and $\{\sigma_r\}$. By assumption, the release time r_i exceeds the travel time between start position 0 and requested position a_i for each request. Thus, the initial values are $C^+_{-\ell-1,r} = r_r$ and $C^-_{-\ell,r+1} = r_{-\ell}$. We then compute by recursion the completion time of the tours $C^-_{i,r+1}$ and $C^+_{-\ell-1,j}$ for the request sets $\{\sigma_{-\ell}, \ldots, \sigma_i\}$ and $\{\sigma_j, \ldots, \sigma_r\}$, respectively. Then, we recursively compute $C^+_{i,j}$ and $C^-_{i,j}$, starting with large difference $d = j - i$ and iteratively decreasing it. We output $C^+_{0,0}$ as the minimum completion time of a feasible server tour.

To prove the correctness of Algorithm 5, we use two lemmas. We first prove a structural result about feasible server tours and then show that the recurrence we use in the dynamic program is correct. Contrary to before, we assume without loss of generality that each request is served when its requested position is visited for the last time in the tour.

ALGORITHM 5: Dynamic Program for Closed Offline TSP on the line.

```
**Input**: A set of requests $\sigma_i = (a_i; r_i)$ with $r_i \geq |a_i|$ for $-\ell \leq i \leq r$ and $a_i < a_{i+1}$ for $-\ell \leq i < r$.
**Output**: The minimum completion time $C_{\max}$ of a tour.
$C^+_{-\ell-1,r} \leftarrow r_r$.
$C^-_{-\ell,r+1} \leftarrow r_{-\ell}$.
for $i = -\ell, \ldots, r$ do
    $C^-_{i,r+1} \leftarrow \max\{r_i, C^-_{i-1,r+1} + a_i - a_{i-1}\}$.
for $j = r, \ldots, \ell$ do
    $C^+_{-\ell-1,j} \leftarrow \max\{r_j, C^+_{\ell-1,j+1} + a_{j+1} - a_j\}$.
for $d = r + \ell, \ldots, 0$ do
    for $i = -\ell, \ldots, r - d$ do
        $j \leftarrow i + d$.
        $C^-_{i,j} \leftarrow \max\{r_i, \min\{C^+_{i-1,j} + a_j - a_i, C^-_{i-1,j} + a_i - a_{i-1}\}\}$.
        $C^+_{i,j} \leftarrow \max\{r_j, \min\{C^+_{i,j+1} + a_{j+1} - a_j, C^-_{i,j+1} + a_j - a_i\}\}$.
return $C_{\max} = C^+_{0,0}$.
```

Lemma 7.2. *At any time in a feasible server tour ending at position p, the set of served requests is the union of two disjoint sets $S_1 = \{\sigma_\ell, \ldots, \sigma_i\}, a_i \leq p$ and $S_2 = \{\sigma_j, \ldots, \sigma_r\}, p \leq a_j$, both of which are contiguous.*

Proof. Suppose there is a time t, at which the set of requests does not have the claimed structure. Then there is a request σ_1 served until time t and a request σ_2 served after t with either $p \leq a_1 < a_2$ or $a_1 < a_2 \leq p$. Without loss of generality assume the first to be true. At time $t' > t$, when request σ_2 is served, the server is at position a_2. The server tour ends at position p and thus passes position a_1 at some time after t. This contradicts that request σ_1 has been served until time t, which was our assumption. □

Lemma 7.3. *Given an instance of TSP on the line with requests $\sigma_\ell, \ldots, \sigma_r$ to positions $a_\ell < a_{\ell+1} < \cdots < a_r$ and completion times $C^+_{i,j+1}, C^-_{i,j+1}, C^+_{i-1,j}$ and $C^-_{i-1,j}$ for some indices $\ell \leq i \leq j \leq r$; then the minimal completion times $C^+_{i,j}$ and $C^-_{i,j}$ are given by the following recurrence:*

$$C^+_{i,j} = \max\{r_j, \min\{C^+_{i,j+1} + a_{j+1} - a_j, C^-_{i,j+1} + a_j - a_i\}\},$$
$$C^-_{i,j} = \max\{r_i, \min\{C^+_{i-1,j} + a_j - a_i, C^-_{i-1,j} + a_i - a_{i-1}\}\}.$$

Proof. The completion time $C^+_{i,j}$, we give is feasible, as it can be achieved by executing the tour attaining $C^+_{i,j+1}$ or the tour attaining $C^-_{i,j+1}$ and then moving to position a_j, waiting there until the release time r_j has passed.

By Lemma 7.2, the tour serving requests $\sigma_\ell, \ldots, \sigma_i$ and $\sigma_j, \ldots, \sigma_r$ serves request σ_j last and request σ_{j+1} or σ_i immediately before. Thus, the completion time exceeds the minimum of $C^+_{i,j+1} + a_{j+1} - a_j$ and $C^-_{i,j+1} + a_j - a_i$. Furthermore, we cannot serve request σ_j before its release time r_j. Symmetrically the recursion for $C^-_{i,j}$ is constructed. □

These two lemmas allow us to prove the correctness of Algorithm 5.

Theorem 7.4. *Algorithm 5 computes the minimum completion time of a server tour for offline TSP on the line in time $O(n^2)$.*

Proof. By Lemma 7.2, the optimal server tour has the structure that it has served the union of two disjoint contiguous sets at any fixed point in time. We start computing the best completion

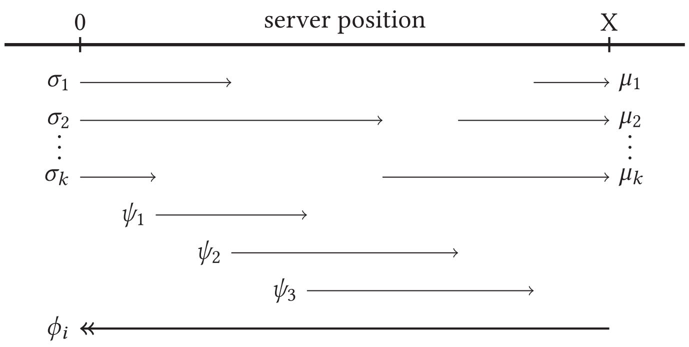

Fig. 1. Construction of a Dial-A-Ride instance for Theorem 7.6.

time of a tour for pairs of request sets containing one request each and increase the number of contained requests $r - \ell - d$ in each iteration by decreasing the parameter d. We end with $d = 0$ and compute the earliest completion time of tours serving the complete request set in this iteration. The recursion we use is correct by Lemma 7.3 and the computation order is feasible, as the recursion formula only contains request set pairs of strictly smaller size.

The algorithm uses quadratic time for the two nested loops and all other steps take linear or constant time. Hence the algorithm runs in time $O(n^2)$. □

Observation 3. *The algorithm of Reference [26] for open TSP on the line can be obtained from Algorithm 5 by changing the computation order of the completion times and returning the minimum completion time of all possible end positions* $\min_{-\ell \leq i \leq r} C^+_{i,i}$.

For the non-preemptive Dial-A-Ride problem on the line, we show that the open and closed variant with release times are NP-hard. Without release times, we prove they are NP-hard for capacity $c \geq 2$. Our reductions are from the Circular Arc Coloring problem [14], which is also used in a reduction for minimizing the sum of completion times of Dial-A-Ride on the line with capacity $c = 1$ [11].

Definition 7.5 (Circular Arc Coloring). Let $\mathcal{I}$ be a family of intervals on a circle, and let $k \in \mathbb{Z}_{>0}$ be a fixed parameter. Decide, if a coloring of all intervals $I \in \mathcal{I}$ with k colors exists, such that no two intervals of the same color overlap.

Theorem 7.6. *The non-preemptive closed offline Dial-A-Ride problem on the line with capacity $c = 1$ is NP-complete.*

Proof. The problem is in NP, as we can decide whether or not a server tour completes before a fixed deadline in polynomial time.

Let an instance of the Circular Arc Coloring problem be given by a circle with circumference X, a set $\mathcal{I}$ of intervals $I = [\ell, r)$, $\ell, r \in [0, X)$, on the circle and coloring number k. Without loss of generality, we can assume that there are exactly k intervals $I^0_1, \dots, I^0_k$ overlapping 0. If there are fewer, then we can add intervals $[0, \varepsilon)$ to the instance for sufficiently small ε. If there are more, then the instance is trivial. We define the following set of requests for the Dial-A-Ride problem: For each interval $I^0_j = [\ell_j, r_j)$, $j \leq k$, we create two requests $\sigma_j = (a_j, b_j; r_j) = (0, r_j; 2(j-1)X)$ and $\mu_j = (\ell_j, X; 2(j-1)X)$. For any other interval $I = [\ell, r)$ define a request $\psi = (\ell, r; 0)$ and furthermore define k requests $\phi_i = (X, 0; 0)$ for $i = \{1, \dots, k\}$. The construction of requests with their start and end position is displayed in Figure 1. Note that the server position increases from left to right and that the release times are not displayed. The bold arrow with a double tip represents the k identical requests ϕ_i.

Any feasible server tour for this instance with start position 0 travels at least k times from 0 to X and back to serve the k requests ϕ_i. Hence, any feasible tour has length at least $2kX$. A tour of exactly that length is closed and partitions the requests into k sets, each served during one trip from 0 to X and back. Observe that requests σ_k and μ_k must be served on the last trip, because they are released at time $2(k-1)X$. Then requests σ_{k-1} and μ_{k-1} must be served on the previous trip. They cannot be served before because of their release time and they cannot be served on the last trip as the server has capacity 1 and serves requests σ_k whose interval overlaps that of σ_{k-1} and also serves request μ_k whose interval overlaps that of μ_{k-1}. Iteratively, we deduce that requests σ_j and μ_j, $j \leq k$, must be served on the same trip. This shows, that if we use the k-partitioning induced by the tour to color the intervals on the circle, we get a feasible circular arc coloring. Conversely, we can transform any circular arc coloring into a feasible server tour of length $2kX$ by scheduling the requests corresponding to one color on the same trip from O to X. □

Remark 2. Note that we can extend this proof to the case $c \geq 1$ if we add $c-1$ requests from 0 to X for each release time $0, 2X, 4X, \dots, 2(k-1)X$. Then any feasible tour finishing at time $2kX$ serves exactly $c-1$ of these additional requests on each trip from 0 to X and thus still partitions the other requests in the desired fashion. However, this result is subsumed by the next theorem.

THEOREM 7.7. *The non-preemptive closed offline* DIAL-A-RIDE *problem on the line with capacity $c \geq 2$ is NP-complete, even when all release times are 0.*

PROOF. The problem is in NP, as we can decide whether or not a server tour completes before a fixed deadline in polynomial time.

Let an instance of the CIRCULAR ARC COLORING problem be given by a circle with circumference X, a set $\mathcal{I}$ of intervals $I = [\ell, r), \ell, r \in [0, X)$, on the circle and coloring number k. Without loss of generality, we can assume that there are exactly k intervals overlapping any point p on the circle. If there are less, then we can add intervals $[p, p+\varepsilon \bmod X)$ to the instance. If there are more, then the instance is trivial. Let the k intervals that overlap the point 0 be $I_1^0, \dots, I_k^0$. We define the following set of requests for the DIAL-A-RIDE problem: For each interval $I_j^0 = [\ell_j, r_j), j \in \{1, \dots, k\}$, we create two requests $\sigma_j = (-j, r_j; 0)$ and $\mu_j = (\ell_j, X+j; 0)$. For any other interval $I = [\ell, r)$ define a request $\psi = (\ell, r; 0)$. Furthermore define for each $j \in \{1, \dots, k\}$ exactly $c-1$ identical requests $\nu_j = (-j, X+j; 0)$, for $j \in \{1, \dots, k-1\}$ exactly c identical requests $\phi_j = (X+j, -j-1; 0)$ and another c identical requests $\phi_k = (X+k, -1; 0)$. Contrary to before, let the start position of the server be -1 for this instance. The set of requests with their start and end position is displayed in Figure 2. In the figure, the server position increases from left to right and release times are not displayed. We depict a set of $c-1$ identical requests by a dashed thick arrow with a double tip, and a set of c identical requests by a thick arrow with a double tip.

Let d be given by $d = 2kX + 2\sum_{j=1}^{k} j$. This is $2c$ times the total length of requests in positive direction, because exactly k intervals overlap each point of the circle and there is one request starting at $-j$ and one ending at $X+j$ for all $j \leq k$. Furthermore for each $j \in \{1, \dots, k\}$ there are $c-1$ requests from $-j$ to $X+j$, so it is also $2c$ times the total length of requests in negative direction. Thus, d is the minimum length of any feasible server tour. A server tour meeting this length bound must travel with full capacity at any time. As the server capacity c is at least 2, the tour must start with serving the $c-1$ requests ν_1, continue with serving requests ϕ_1 and then serve ν_j and ϕ_j for increasing indices $j = 2, \dots, k$. This tour leaves capacity 1 on each trip from $-j$ to $X+j$ for requests from the set $\bigcup_{j=1}^{k} \{\sigma_j, \mu_j\}$ and the requests of type ψ. Hence, any tour of length d yields a partition of the requests σ, μ and ψ into k sets of non-overlapping requests. Moreover, requests σ_j and μ_j occur in the same set of the partition. Thus, assigning one color to each set of the partition yields a feasible coloring of the circular arc instance. Conversely, we can use any circular arc coloring

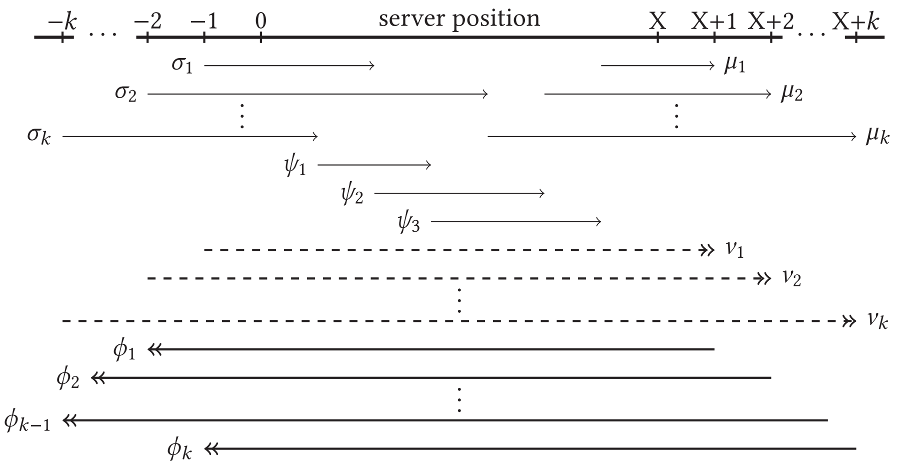

Fig. 2. Construction of a Dial-A-Ride instance for Theorem 7.7.

to design a server tour of length d. We serve requests ν and ϕ as described above and use for the $(j+1)$-st trip in positive direction all requests with the same color as request σ_j in the arc coloring. □

Theorem 7.8. *The non-preemptive open and closed offline Dial-A-Ride problem on the line are NP-complete. For capacity $c \geq 2$ this even holds when all release times are 0.*

Proof. Theorems 7.6 and 7.7 show the statement for the closed problem variant. For the open case, we show in the proof of Theorem 7.6, that deciding if there is a tour of length $2kX$ is hard. Also for the open problem variant, any feasible tour has at least this length as a feasible server tour starts at position 0 and travels at least k times from X to 0 to serve the k requests ϕ_i. This also shows that any tour attaining this bound is a closed tour.

In the proof of Theorem 7.7, we consider closed tours of full capacity. Again this shows that there cannot be a smaller open tour and that all minimal open tours are closed. □

Remark 3. It is particularly interesting, that in the hardness reduction the server's start position is at one of the two extreme positions occurring in the tour. For the same problem allowing preemption, Karp showed that it is solvable in polynomial time [19], while the hardness of the problem with an arbitrary start position is not known.

Remark 4. Non-preemptive offline Dial-A-Ride on the line is known to be easy without release dates [11] and becomes hard if each request comes with a deadline in addition to its release time [28]. Thus, the only complexity question remaining open is the case with unbounded capacity.

In the classification of closed dial-a-ride problems in Reference [11], offline TSP on the line is the problem $1|s=t, d_j|\text{line}|C_{\max}$ (release dates are the symmetric case to deadlines). De Paepe et al. [11] claim Tsitsiklis [28] shows a polynomial algorithm, but this is for the open version. We solve the closed variant by giving a polynomial algorithm (Theorem 7.4) as well as a counterexample to the algorithm of Psaraftis et al. [26]. Our Theorem 7.8 shows the problem $1, \text{cap}1|d_j|\text{line}|C_{\max}$ in the same classification scheme is NP-hard. While this is implicitly claimed in Reference [11], no proof is given. For the generalization to arbitrary capacity but without release dates, $1||C_{\max}$, Guan [16] showed hardness for capacity $c = 2$ and our new hardness proof handles any capacity $c \geq 2$.

ACKNOWLEDGMENTS

The authors thank the anonymous referees for their effort and valuable comments.

REFERENCES

[1] Foto Afrati, Stavros Cosmadakis, Christos H. Papadimitriou, George Papageorgiou, and Nadia Papakostantinou. 1986. The complexity of the travelling repairman problem. *Inform. Theor. Appl.* 20, 1 (1986), 79–87.

[2] Norbert Ascheuer, Martin Grötschel, Sven Oliver Krumke, and Jörg Rambau. 1998. Combinatorial on-line optimization. In *Proceedings of the International Conference of Operations Research (OR'98).* 21–37.

[3] Norbert Ascheuer, Sven Oliver Krumke, and Jörg Rambau. 2000. Online dial-a-ride problems: Minimizing the completion time. *Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science (STACS'00).* 639–650.

[4] Mikhail J. Atallah and S. Rao Kosaraju. 1988. Efficient solutions to some transportation problems with applications to minimizing robot arm travel. *SIAM J. Comput.* 17, 5 (1988), 849–869.

[5] Giorgio Ausiello, Esteban Feuerstein, Stefano Leonardi, Leen Stougie, and Maurizio Talamo. 1994. Serving requests with on-line routing. In *Proceedings of the 4th Scandinavian Workshop on Algorithm Theory Aarhus on Algorithm Theory (SWAT'94).* 37–48.

[6] Giorgio Ausiello, Esteban Feuerstein, Stefano Leonardi, Leen Stougie, and Maurizio Talamo. 1995. Competitive algorithms for the on-line traveling salesman. In *Proceedings of the 4th International Workshop on Algorithms and Data Structures (WADS'95).* 206–217.

[7] Giorgio Ausiello, Esteban Feuerstein, Stefano Leonardi, Leen Stougie, and Maurizio Talamo. 2001. Algorithms for the on-line travelling salesman. *Algorithmica* 29, 4 (2001), 560–581.

[8] Antje Bjelde, Yann Disser, Jan Hackfeld, Christoph Hansknecht, Maarten Lipmann, Julie Meißner, Kevin Schewior, Miriam Schlöter, and Leen Stougie. 2017. Tight bounds for online TSP on the line. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA'17).* 994–1005.

[9] Michiel Blom, Sven O. Krumke, Willem de Paepe, and Leen Stougie. 2001. The online TSP against fair adversaries. *INFORMS J. Comput.* 13, 2 (2001), 138–148.

[10] Moses Charikar and Balaji Raghavachari. 1998. The finite capacity dial-a-ride problem. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS'98).* 458–467.

[11] Willem E. de Paepe, Jan Karel Lenstra, Jiri Sgall, René A. Sitters, and Leen Stougie. 2004. Computer-aided complexity classification of dial-a-ride problems. *INFORMS J. Comput.* 16, 2 (2004), 120–132.

[12] Esteban Feuerstein and Leen Stougie. 2001. On-line single-server Dial-a-Ride problems. *Theor. Comput. Sci.* 268, 1 (2001), 91–105.

[13] Greg N. Frederickson and Dih Jiun Guan. 1993. Nonpreemptive ensemble motion planning on a tree. *J. Algor.* 15, 1 (1993), 29–60.

[14] Michael R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, New York.

[15] Paul C. Gilmore and Ralph E. Gomory. 1964. Sequencing a one state-variable machine: A solvable case of the traveling salesman problem. *Oper. Res.* 12, 5 (1964), 655–679.

[16] Dih Jiun Guan. 1998. Routing a vehicle of capacity greater than one. *Discrete Appl. Math.* 81, 1–3 (1998), 41–57.

[17] Dawsen Hwang and Patrick Jaillet. 2018. Online scheduling with multi-state machines. *Networks* 71, 3 (2018), 209–251.

[18] Patrick Jaillet and Michael R. Wagner. 2008. Generalized online routing: New competitive ratios, resource augmentation, and asymptotic analyses. *Oper. Res.* 56, 3 (2008), 745–757.

[19] Richard M. Karp. 1972. Two combinatorial problems associated with external sorting. In *Proceedings of the Combinatorial Algorithms, Courant Computer Science Symposium.* 17–29.

[20] Sven O. Krumke. 2001. Online Optimization Competitive Analysis and Beyond. Habilitation thesis.

[21] Sven O. Krumke, Willem E. de Paepe, Diana Poensgen, and Leen Stougie. 2003. News from the online traveling repairman. *Theor. Comput. Sci.* 295, 1–3 (2003), 279–294.

[22] Sven O. Krumke, Luigi Laura, Maarten Lipmann, Alberto Marchetti-Spaccamela, Willem de Paepe, Diana Poensgen, and Leen Stougie. 2002. 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'02).* 200–214.

[23] Eugene L. Lawler, Jan Karel Lenstra, Alexander H. G. Rinnooy Kan, and David B. Shmoys (Eds.). 1985. *The Traveling Salesman Problem; A Guided Tour of Combinatorial Optimization.* Wiley, Chichester.

[24] Maarten Lipmann. 2003. *On-Line Routing.* Ph.D. Dissertation. Technical University Eindhoven.

[25] Mark S. Manasse, Lyle A. McGeoch, and Daniel D. Sleator. 1990. Competitive algorithms for server problems. *J. Algor.* 11, 2 (1990), 208–230.

[26] Harilaos N. Psaraftis, Marius M. Solomon, Thomas L. Magnanti, and Tai-Up Kim. 1990. Routing and scheduling on a shoreline with release times. *Manage. Sci.* 36, 2 (1990), 212–223.
[27] René Sitters. 2002. The minimum latency problem is np-hard for weighted trees. In *Proceedings of the 9th International Conference on Integer Programming and Combinatorial Optimization.* 230–239.
[28] John N. Tsitsiklis. 1992. Special cases of traveling salesman and repairman problems with time windows. *Networks* 22, 3 (1992), 263–282.

Received July 2018; revised April 2020; accepted July 2020