Competitive analysis of the online dial-a-ride problem

Zur Erlangung des Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Dissertation von Alexander Marcel Birx aus Lich, Hessen, Deutschland
Fachbereich 04 – Mathematik
Tag der Einreichung: 8. Juli 2020, Tag der Prüfung: 9. Oktober 2020

1. Gutachten: Prof. Dr. Yann Disser
2. Gutachten: Prof. Dr. Leen Stougie
Darmstadt, 2020

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Mathematik
AG Optimierung
Diskrete Optimierung

Competitive analysis of the online dial-a-ride problem

genehmigte Dissertation von Alexander Marcel Birx

1. Gutachten: Prof. Dr. Yann Disser
2. Gutachten: Prof. Dr. Leen Stougie

Tag der Einreichung: 8. Juli 2020
Tag der Prüfung: 9. Oktober 2020

Darmstadt, 2020

Darmstadt, Technische Universität Darmstadt
Jahr der Veröffentlichung auf TUprints: 2020
URN: urn:nbn:de:tuda-tuprints-141341
URL: http://tuprints.ulb.tu-darmstadt.de/14134

Dieses Dokument wird bereitgestellt von tuprints,
E-Publishing-Service der TU Darmstadt
http://tuprints.ulb.tu-darmstadt.de
tuprints@ulb.tu-darmstadt.de

Acknowledgements

First and foremost, I would like to thank my supervisor Prof. Dr. Yann Disser for giving me the opportunity to work in his research group and suggesting this interesting research topic.

I am very grateful to Yann for his guidance and support from which I profited immensely over the past years. I thank Prof. Dr. Christian Bischof for acting as a co-supervisor of this thesis within the Graduate School of Computational Engineering. I am thankful to Prof. Dr. Leen Stougie for acting as a referee of this thesis. Furthermore, I thank Prof. Dr. Marc Pfetsch and Prof. Dr. Christian Bischof for their willingness to participate in my examination committee. I thank Dr. Kevin Schewior for the fruitful collaboration on our publication concerning the open version of online DIAL-A-RIDE.

Many thanks go to my colleagues at the Research Group Optimization and the Graduate School of Computational Engineering at Technische Universität Darmstadt for creating a great working atmosphere. In particular, I want to thank Alexander V. Hopp and Christina Karousatou for the productive collaboration in the Wednesday Project that resulted in a publication concerning online Graph Exploration.

Furthermore, I am grateful to my sister Katharina and my colleague Alexander V. Hopp for proof-reading parts of this thesis and for their valuable suggestions. Last but certainly not least I would like to thank my wife Viviane for her patience and constant support.

I gratefully acknowledge the support of the Excellence Initiative of the German Federal and State Governments and the Graduate School of Computational Engineering at Technische Universität Darmstadt.

Abstract

Online optimization, in contrast to classical optimization, deals with optimization problems whose input data is not immediately available, but instead is revealed piece by piece. An *online algorithm* has to make irrevocable optimization decisions based on the arriving pieces of data to compute a solution of the online problem. The quality of an online algorithm is measured by the *competitive ratio*, which is the quotient of the solution computed by the online algorithm and the optimum offline solution, i.e., the solution computed by an optimum algorithm that has knowledge about all data from the start.

In this thesis we examine the online optimization problem online DIAL-A-RIDE. This problem consists of a server starting at a distinct point of a metric space, called origin, and serving transportation requests that appear over time. The goal is to minimize the makespan, i.e., to complete serving all requests as fast as possible. We distinguish between a closed version, where the server is required to return to the origin, and an open version, where the server is allowed to stay at the destination of the last served request.

In this thesis, we provide new lower bounds for the competitive ratio of online DIAL-A-RIDE on the real line for both the open and the closed version by expanding upon the approach of [13]. In the case of the open version, the improved lower bound separates online DIAL-A-RIDE from its special case online TSP, where starting position and destination of requests coincide.

To produce improved upper bounds for the competitive ratio of online DIAL-A-RIDE, we generalize the design of the IGNORE algorithm and the SMARTSTART algorithm [5] into the class of *schedule-based algorithms*. We show lower bounds for the competitive ratios of algorithms of this class and then provide a thorough analysis of IGNORE and SMARTSTART. Identifying and correcting a critical weakness of SMARTSTART gives us the improved SMARTERSTART algorithm. This schedule-based algorithm attains the best known upper bound for open online DIAL-A-RIDE on the real line as well as on arbitrary metric spaces.

Finally, we provide an analysis of the REPLAN algorithm [5] improving several known bounds for the algorithm's competitive ratio.

Zusammenfassung

Im Kontrast zur klassischen Optimierung, handelt Online Optimierung von Optimierungsproblemen, deren Parameter nicht unmittelbar bekannt sind, sondern stattdessen nach und nach verfügbar werden. Ein *online Algorithmus* muss unwiderrufliche Optimierungsentscheidungen basierend auf den gerade vorhandenen Daten treffen, um eine Lösung des online Optimierungsproblems zu berechnen. Die Güte eines online Algorithmus wird als *kompetitiver Faktor* angegeben. Dieser ist der Quotient einer Lösung, welche von einem online Algorithmus berechnet wurde, und der optimalen offline Lösung, d. h., der Lösung, die von einem optimalen Algorithmus berechnet wurde, der alle Daten bereits zu Beginn zur Verfügung hat.

In dieser Dissertation wird das online Problem online Dial-a-Ride behandelt. In diesem online Problem muss ein Zusteller, welcher in einem ausgezeichneten Punkt, Ursprung genannt, eines metrischen Raumes startet, Transportanfragen, welche zu verschiedenen Zeitpunkten erscheinen, bedienen. Ziel ist es, die Gesamtzeit für das Bedienen aller Anfragen zu minimieren. Wir unterscheiden zwischen einer geschlossenen Version des Problems, in der der Zusteller wieder zum Ursprung zurückkehren muss und einer offenen Version des Problems.

In dieser Dissertation beweisen wir verbesserte untere Schranken für den kompetitiven Faktor von online Dial-a-Ride auf der reellen Achse, sowohl für die offene, als auch für die geschlossene Variante des Problems. Die Schranke für die offene Version baut auf einer Konstruktion aus [13] auf und separiert online Dial-a-Ride von seinem Spezialfall online TSP, in welchem jede Anfrage jeweils den gleichen Start- und Endpunkt hat.

Im Weiteren verallgemeinern wir das Design des Ignore Algorithmus und des Smartstart Algorithmus [5] und führen die Klasse der *schedulebasierten Algorithmen* ein. Wir zeigen untere Schranken an den kompetitiven Faktor von Algorithmen dieser Klasse und führen eine umfassende Analyse von Ignore und Smartstart durch. Durch das Identifizieren und Beheben einer kritischen Schwäche von Smartstart, erhalten wir den Algorithmus SmarterStart. Dieser schedulebasierte Algorithmus ist der beste bekannte Algorithmus für online Dial-a-Ride sowohl auf der reellen Achse als auch in beliebigen metrischen Räumen.

Abschließend analysieren wir den Algorithmus Replan [5] und verbessern mehrere bekannte Schranken an seinen kompetitiven Faktor.

Contents

1 Introduction

In classical optimization we deal with problems that consist of given input data, and we need to make optimization decisions based on this data to optimize a certain objective. While this kind of framework allows to model a large variety of problems, it is too restrictive for problems where a part of the input data is revealed over time and optimization decisions need to be made before all input data is available.

Consider for example an elevator and the problem of bringing every arriving person to their desired destination as fast as possible. Since people arrive over time, we do not know when and on which floor they will arrive and the elevator needs to move before the last person has arrived, i.e., before all input data becomes available, if it wants to achieve a good completion time. Another example would be a robot with the goal of exploring an unknown cave network as efficiently as possible. Since the layout of the cave becomes only available by exploring it, an optimum traversal cannot be computed without making irrevocable decisions about the path to follow.

In comparison to an algorithm for a classical optimization problem that computes an optimum solution based on the given input data, an *online algorithm* gets its input data item by item and has to make an irrevocable optimization decision based on the piece of data that arrives. Examples for input items would be the arrival of a new person at some floor that wants to use the elevator or the knowledge gained by the robot by making a step in a certain direction. The online algorithm needs to find a balance between acting efficiently regarding the known data and protecting itself against future input data. Imagine for example an elevator located at the first floor and a person arriving at the second floor that wants to go the third floor. Of course, it is optimal to serve the request of this person right away – at least as long as no other requests appear. But if the algorithm acts accordingly and is unlucky, a person in the basement might appear that also wants to go to the third floor. Now the elevator has to go all the way down from the third floor to the basement and up to the third floor again, while it would have been much more efficient to serve both requests in one go.

The example above shows that no algorithm controlling an elevator can act optimally: If the elevator acts prematurely as described above, it might end up covering more distance than necessary. However, if it decides to wait and no further requests appear, it incurs unnecessary waiting time. This unavailability of the full amount of information puts online algorithms in a disadvantage compared to a hypothetical *optimum offline algorithm* that has access to all information from the start.

To measure the quality of an online algorithm, i.e., how much its result differs from the offline optimum, we use *competitive analysis* based on the algorithm's worst case behavior: The algorithm is challenged by an *adversary* that answers every choice of the algorithm by creating worst-case future input items. In the example of the elevator above, a simple adversary strategy would be to introduce the request at the basement, if the algorithm acts prematurely and to do nothing in the case that the algorithm decides to wait. Once all input items have been revealed, we compare the algorithm's result with the *optimum offline solution*, i.e., the solution provided by an optimum offline algorithm. The quotient of the algorithm's result and the optimum offline solution is called the *competitive ratio* of the algorithm.

The competitive ratio is a quite harsh measurement for the quality of an online algorithm and for many online optimization problems it is known that no online algorithm with constant competitive ratio exists. However, in this thesis we will only discuss online problems with existing but not tight constant bounds for the competitive ratio.

Outline

After introducing the necessary tools and formal definitions in this introductory chapter, we introduce the problem online Dial-a-Ride and its special case online TSP in Chapter 2. In the second chapter, we also state the currently known results for the problem, give a brief summary of the results that are presented in this thesis and discuss related work.

In the third chapter, we present two new lower bounds for the competitive ratio of online Dial-a-Ride by expanding upon the approach of [13]. One of the improved lower bound separates online Dial-a-Ride from its special case online TSP.

Starting from Chapter 4, we focus more on upper bounds for the competitive ratio of online Dial-a-Ride. We introduce the class of *schedule-based algorithms*, analyze several properties of algorithms belonging to this class and give a thorough analysis of the Ignore algorithm [5].

In Chapter 5, we give a detailed analysis of the Smartstart algorithm, providing its exact competitive ratio for the case that the underlying metric space is the real line and improving upper bounds for its competitive ratio for arbitrary metric spaces.

Identifying and correcting a critical weakness of Smartstart gives us the improved SmarterStart algorithm, which is examined in Chapter 6. This schedule-based algorithm attains the best known upper bound for the competitive ratio of open online Dial-a-Ride on the real line as well as on arbitrary metric spaces.

Finally, in Chapter 7, we provide an analysis of the REPLAN algorithm [5] improving several known bounds for the algorithm's competitive ratio.

1.1 Offline Optimization and Offline Algorithms

In the following, we formalize the notions introduced above and start with the definition of classical offline optimization problems. The notations and definitions are inspired by [9].

Definition 1.1. An *offline optimization problem* $\mathfrak{P}$ is a tuple $(\mathcal{I}, \mathcal{S}, f, c)$ where:

- $\mathcal{I}$ is the set of input instances,
- $\mathcal{S}$ is the set of solutions,
- $f : \mathcal{I} \to 2^{\mathcal{S}}$ maps an instance $I \in \mathcal{I}$ to a set of feasible solutions $F_I \subseteq \mathcal{S}$,
- $c : \mathcal{I} \times \mathcal{S} \to \mathbb{R}$ maps a solution of an instance to a cost.

The set of *optimum solutions* of an instance $I \in \mathcal{I}$ is $\operatorname{argmin}_{x \in F_I} c(I, x)$.

An example for a combinatorial offline optimization problem would be finding the chromatic number of a graph with $k \in \mathbb{N}$ vertices: A coloring of an undirected graph $G = (V, E)$ with vertex set V and edge set E is a function $g \colon V \to \mathbb{N}$ with $g(v) \neq g(u)$ for every $\{v, u\} \in E$. The chromatic number is $\chi(G) := \min_g |\mathrm{im}(g)|$ with g being a coloring of G. In this case, every graph $G = (V, E)$ with $|V| = k$ is an input instance and every function mapping from the set $\{1, \ldots, k\}$ to $\mathbb{N}$ is a solution. Furthermore, the set of coloring functions of G, i.e., $\{g \colon V \to \mathbb{N} \mid g \text{ coloring of } G\} = F_G$, is the set of feasible solutions of the instance G and $c(G, g) := |\mathrm{im}(g)|$ is the cost function.

Another classical example for an offline optimization problem is an $(m \times n)$-dimensional linear program of the form

$$\min c^\top x \qquad \text{subject to} \qquad Ax \leq b.$$

In this case $(A, b) \in \mathbb{R}^{m \times n} \times \mathbb{R}^m = \mathcal{I}$ is an input instance, the vector $x \in \mathbb{R}^n = \mathcal{S}$ is a solution, $f(A, b) = \{x \in \mathbb{R}^n \mid Ax \leq b\} = F_{(A,b)}$ is the set of feasible solutions and the vector $c \in \mathbb{R}^n$ is the cost function.

Definition 1.2. An *offline algorithm* ALG computes a feasible solution $\text{ALG}[I] \in F_I$ (if existent) of cost $\text{ALG}(I) = c(I, \text{ALG}[I])$ for every $I \in \mathcal{I}$.

Note that offline algorithms do not need to produce optimum results. Therefore, the algorithm that just returns a function that maps every vertex $v \in V$ to an unique natural number is an offline algorithm for the coloring problem. In the case of a linear program, every algorithm that produces a feasible solution, i.e., for example every Phase I for the simplex algorithm is an offline algorithm.

Definition 1.3. By $\textsc{Opt}$ we denote an *optimum offline algorithm* that computes an optimum solution $\textsc{Opt}[I] \in F_I$ (if existent) of cost $\textsc{Opt}(I) = \min_{F \in F_I} c(I, F)$ for every instance $I \in \mathcal{I}$.

Examples for optimum offline algorithms for linear programs would be the simplex algorithm or the ellipsoid method. Note that the word "optimum" in Definition 1.3 does not refer to the complexity of the algorithm, but just to the optimality of the result produced by the algorithm. Therefore, the algorithm that checks the value $|\mathrm{im}(f)|$ for every coloring function $f : V \to \{1, \dots, |V|\}$ and returns a function with the lowest value is an optimum offline algorithm for the coloring problem – albeit being very inefficient.

1.2 Online Algorithms and Competitiveness

In contrast to a classical offline optimization problem, we define an online optimization problem as a *request-response-game*: Instead of having all input available at the start, the input arrives item by item, i.e., request by request, and needs to be responded to item by item, always incurring an irrevocable cost.

Definition 1.4. An *online optimization problem* is a tuple $\mathfrak{O} = (\mathcal{R}, \mathcal{A}, \Sigma, F, C)$ where:

- $\mathcal{R}$ is the set of input items,
- $\mathcal{A}$ is the set of responses,
- $\Sigma \subseteq \bigcup_{i \in \mathbb{N}} \mathcal{R}^i$ is the set of all input sequences,
- $F = (f_i)_{i \in \mathbb{N}}$ is a sequence of functions $f_i : \mathcal{R}^i \times \mathcal{A}^{i-1} \to \mathcal{A}$ mapping the first i input items $(s_1, \dots, s_i)$ and the first $i - 1$ responses $(a_1, \dots, a_{i-1})$ to the set of feasible responses $f_i(s_1, \dots, s_i, a_1, \dots, a_{i-1})$ in step i,
- $C = (c_i)_{i \in \mathbb{N}}$ is a sequence of cost functions $c_i : \mathcal{R}^i \times \mathcal{A}^i \to \mathbb{R}_{\geq 0}$ with $c_0 := 0$.

For an input sequence $\sigma = (s_1, \dots, s_n) \in (\Sigma \cap R^n)$ and $i \leq n$, we define the subsequence $\sigma_{\leq i} := (s_1, \dots, s_i)$.

Throughout this thesis, σ will denote an arbitrary input sequence of length $n \in \mathbb{N}$. Whenever we talk about a specific input sequence, it will be given a specific identifier.

An example for an online optimization problem that is strongly connected to integer programming is the online knapsack problem. In the classical knapsack problem, we have a set of items that have weights $w_i \geq 0$ and values $v_i \geq 0$. The goal is to choose a subset of items such that the total weight of the chosen items is bounded by a constant W and the total value is maximized.

We can turn this problem into an online problem by changing the set of items into a sequence of items, introducing them one by one and demanding an irrevocable decision whether the item is taken or not, every time an item arrives. Then, the set of input items is $\mathbb{R}^2_{\geq 0}$, i.e., the set of all possible tuples of weights and values. The set of responses is $\{0, 1\}$, where 1 means that the current item is chosen and 0 that it is not chosen. The set of input sequences is $\bigcup_{n \in \mathbb{N}} \mathbb{R}^{2 \times n}$, i.e., the set of all sequences of tuples of weights and values. The set of feasible responses in step i is

$$f_i(\sigma_{\leq i}, a_1, \dots, a_{i-1}) = \begin{cases} \{0, 1\}, & \text{if } \sum_{j=1}^{i-1} a_j w_j \leq W - w_i, \\ \{0\}, & \text{otherwise.} \end{cases}$$

Last but not least, the cost functions are $c_i((w_j, v_j, a_j)_{j \in \{1,\dots,i\}}) := \sum_{j=1}^{i} (1 - a_j) v_j$. Note that we take the difference of the total value $\sum_{j=1}^{i} v_j$ and the sum $\sum_{j=1}^{i} a_j v_j$ as cost function since we require the cost function to be positive and only regard minimization problems.

Definition 1.5. An *online algorithm* $\textsc{Alg}$ computes a sequence of response functions $(g_i : \mathcal{R}^i \to f_i(\sigma_{\leq i}, g_1(s_1), \dots, g_{i-1}(s_1, \dots, s_{i-1})))_{i \in \mathbb{N}}$. The *solution* of $\textsc{Alg}$ for an input sequence $\sigma = (s_1, \dots, s_n) \in \Sigma$ is

$$\textsc{Alg}[\sigma] := (a_1, \dots, a_n)$$

with $a_i := g_i(\sigma_{\leq i})$. $\textsc{Alg}$'s total cost is $\textsc{Alg}(\sigma) := c_n(\sigma, \textsc{Alg}[\sigma])$.

Note that an online algorithm only uses $\sigma_{\leq i}$ for the computation of the answer a_i, i.e., answer a_i has to be given irrevocably before input item s_{i+1} becomes available. In consequence, an online algorithm usually does not compute an optimum solution. In fact, for the online problems discussed in this thesis there are no online algorithms that compute optimum solutions for all input sequences.

If we fix the number n of input items, every online problem can be interpreted as an offline problem: We set $\mathcal{I} = \mathcal{R}^n$ and $\mathcal{S} = \mathcal{A}^n$. The set of feasible solutions F_σ of an instance σ consists of all response sequences $(a_1, \dots, a_n)$ with $a_i \in f_i(\sigma_{\leq i}, a_1, \dots, a_{i-1})$

for all $i \in \{1, \dots, n\}$. Last but not least, we set $c(\sigma, (a_i)_{i\in\{1,\dots,n\}}) = c_n(\sigma, (a_i)_{i\in\{1,\dots,n\}})$. If interpreted like that, the set of *optimum offline solutions* for an input sequence σ is defined as $\textsc{Opt}[\sigma] \in \operatorname{argmin}_{a\in F_\sigma} c_n(\sigma, a)$. This is identical to an optimum solution as defined in Definition 1.1.

From now on let $\textsc{Opt}$ always be a fixed optimum offline algorithm that computes an optimum solution $\textsc{Opt}[\sigma]$ of cost $\textsc{Opt}(\sigma) = c_n(\sigma, \textsc{Opt}[\sigma])$. Observe that an online algorithm $\textsc{Alg}$ has to obey some consistency between its steps, i.e., for an input $\sigma = (s_1, \dots, s_n)$ and for every $i < j \le n$, the solution $\textsc{Alg}[\sigma_{\le i}]$ is a prefix of the solution $\textsc{Alg}[\sigma_{\le j}]$. $\textsc{Opt}$ in contrast does not need to obey this consistency, i.e., $\textsc{Opt}[\sigma_{\le i}]$ can completely differ from $\textsc{Opt}[\sigma_{\le j}]$. We compare the performances of $\textsc{Alg}$ and $\textsc{Opt}$.

Definition 1.6. An online algorithm $\textsc{Alg}$ is (strictly) *ρ-competitive* if, for all instances $\sigma \in \Sigma$, we have

$$\textsc{Alg}(\sigma) \le \rho\textsc{Opt}(\sigma).$$

Note that we have put no restriction on ρ, i.e., ρ can be constant, but also a function of σ or a function of every other problem parameter. Since Definition 1.6 demands that $\textsc{Alg}(\sigma) \le \rho\textsc{Opt}(\sigma)$ holds for every input instance σ, the competitiveness of an online algorithm measures how well it competes against the optimum offline solution in a worst-case scenario. Note that the solution $\textsc{Alg}[\sigma]$ cannot be better than the offline solution $\textsc{Opt}[\sigma]$, i.e., we always have $\rho \ge 1$. The lack of information puts the online algorithm $\textsc{Alg}$ at severe disadvantage in comparison to the optimum offline algorithm $\textsc{Opt}$. For example, in the case of the online knapsack problem, there is no ρ-competitive online algorithm for a constant $\rho \ge 1$ [38].

Definition 1.7. The *competitive ratio* of an online algorithm $\textsc{Alg}$ is the infimum over all $\rho \ge 1$, such that $\textsc{Alg}$ is ρ-competitive.

Note that an online algorithm with competitive ratio ρ is also ρ-competitive, while the converse is not necessarily true. While the competitive ratio describes $\textsc{Alg}$'s worst-case quality precisely, ρ-competitiveness only gives an upper bound for the quality of the worst-case behavior of $\textsc{Alg}$. The competitive ratio allows us to compare the worst-case performances of online algorithms. The competitive ratio of an online optimization problem is defined as the competitive ratio of the best-possible online algorithm for it.

Definition 1.8. The *competitive ratio* of an online optimization problem is the infimum over all $\rho \ge 1$ for which a ρ-competitive algorithm exists.

The goal of this thesis is to analyze several online problems and give new improved upper and lower bounds for their competitive ratios. An upper bound for the competitive ratio of an online problem can be proven by developing and analyzing an online algorithm for it. A lower bound for the competitive ratio can be proven by competitive analysis: We let a hypothetical online algorithm compete against an adversary, i.e., we construct a worst-possible input sequence for every possible sequence of actions of the online algorithm. In the next chapter we introduce the main online problems that are discussed in this thesis: online DIAL-A-RIDE and its special case online TSP.

History of Online Optimization

The first publication that studied online optimization and online algorithms was [44]. In this paper Sleator and Tarjan studied the problem online LIST UPDATE, in which elements of a *linked list* are requested over time. Accessing an element located closer to the back of the list is more costly than accessing an element located closer to the front, however, accessed elements are allowed to be moved further to the front for free. The authors realized that for this problem it is more interesting to measure the worst-case performance of an algorithm than the expected performance. This led them to prove that the online algorithm MOVETOFRONT for online LIST UPDATE is 2-competitive – even though they did not use the term "competitiveness" in their publication.

Formally, the terms "competitiveness" and "online algorithm" were first intruduced in [31]. In this paper, Karlin et al. studied the problem online PAGING. In this problem, the content of a cache with limited memory is accessed over time. Loading content to the cache incurs a cost and an element must be loaded to the cache if it is requested, while not being in the cache.

Ben-David et al. were the first to give a more rigorous description of online optimization problems [9]. They formally defined online optimization problems as request-response-games the way it is presented in Definition 1.4 and introduced the notion of the "adversary".

Over the years, a large variety of online optimization problems have been studied. Albers et al. studied the problem online LIST UPDATE from a randomized perspective [1, 3]: In comparison to a deterministic online algorithm, a randomized algorithm adds random choices into its computation. Consequently, instead of the costs $\text{ALG}(\sigma)$, the expected value of all possible costs is compared with the optimum offline result to compute the *randomized competitiveness* of the randomized online algorithm. Albers et al. introduced two 2-competitive, deterministic online algorithms for online LIST UPDATE, TIMESTAMP [3] and BIT, which they combined into the 1.6-competitive, randomized algorithm COMB [1].

The already mentioned online KNAPSACK problem is an example for an online problem that has no bounded competitive ratio [38]. However, a randomized version of it admits a bounded randomized competitive ratio: Albers et al. provided a $(1/6.65)$-competitive randomized algorithm for the randomized problem in which the adversary is allowed to choose the item set – but not its order [2].

In the beginning of this chapter we used the example of a robot with the goal of exploring an unknown cave network as efficiently as possible to introduce online optimization problems. This problem is called online GRAPH EXPLORATION and has been studied extensively [4, 7, 15, 21, 22, 23, 24, 26, 30, 39, 42]. Miyazaki et al. [40] gave a $(1+\sqrt{3})/2$-competitive algorithm for cycles and showed that this is best-possible. For graphs with bounded genus g Megow et al. provided the $16(1+2g)$-competitive BLOCKING algorithm, which is a modification of the depth-first search algorithm. For general graphs Rosenkrantz et al. published a $\Theta(\log n)$-competitive algorithm [42], i.e., no online algorithm with constant competitive ratios is known until now. The best known lower bound for general graphs is $10/3$ [11]

Another widely studied problem is online k-SERVER, where k servers are located in a metric space. Every time a request is served a new request appears. An incoming request needs to be served by one of the servers incuring a cost dependent on the distance between the server and the request. Manasse et al. showed that there is no online algorithm for k-SERVER with a competitive ratio lower than k on any metric space with at least $k+1$ points [37]. Even though Chrobak et al. provided several k-competitive algorithms for k-SERVER on a large variety on metric spaces [19, 20, 18], it remains open, if there is a k-competitive online algorithm for every metric space.

The online optimization problem online DIAL-A-RIDE, that is studied in this thesis, was first examined by Ascheuer et al. in [5]. In comparison to online k-SERVER the problem online DIAL-A-RIDE consists of a single server and requests appear over time independently of the server's actions. The goal is to minimize the total completion time. A detailed discussion of results follows in the next chapter.

2 Online Dial-a-Ride and Online TSP

In the opening chapter of this thesis we briefly discussed an elevator as an example for an online optimization problem. In this chapter we will provide a more detailed look at this problem, including a precise mathematical representation of the problem and an expansion to a more general version of it.

For a start let us take a more abstract look at the elevator: People arrive over time at a certain floor and want to be carried to another floor. Therefore, the request of an arriving person is defined by the floor, on which the person arrives, the floor, the person wants to be carried to and the time of the person's arrival. The elevator starts at the ground floor and moves with unit speed. For simplicity we assume that our elevator has no braking or accelaration time, i.e., at every point of time the elevator is either moving with unit speed or not moving at all. Furthermore, our elevator has a capacity limit, i.e., only a fixed number of people can be inside elevator at the same time. Our goal is to minimize the total completion time of carrying all people to their desired floors.

To put this problem in a more general and more mathematical framework: Our elevator is a *server* moving with unit speed in a metric space (X, d) and starting at some unique position $0 \in X$ called *origin*. Over time, *requests* $s = (a, b; r)$ are revealed, where $a \in X$ is the *starting position* of the request, $b \in X$ is the *destination* of the request and $r \in \mathbb{R}_{\geq 0}$ is the *release time* of the request. These requests need to be loaded by the server at their starting positions, after their release time and need to be transported to their destinations. The server has a *capacity* $c \in \mathbb{N} \cup \{\infty\}$, i.e., only c requests can be loaded at the same time. The goal is to minimize the *makespan*, i.e., to minimize the total completion time. This problem is called *online* Dial-a-Ride.

In the case of the elevator, the metric space (X, d) is the real line $\mathbb{R}$ equipped with the euclidean distance function $d(p, q) := |p - q|$. Then, the basement floors are represented by negative numbers and floors above the ground are represented by the positive numbers. It is clear that in general the metric space X needs to be equipped with a positive and symmetric distance function $d\colon X^2 \to \mathbb{R}_{\geq 0}$ that obeys the triangle inequality. However, we also have to ensure that the server can move continuously with unit speed. Therefore, we only allow metric spaces satisfying a smoothness property: For all pairs of positions $(p, q) \in X^2$, there needs to be a rectifiable path $\gamma : [0, 1] \to X$, with $\gamma(0) = p$ and $\gamma(1) = q$ of length $d(p, q)$ (see e.g. [8] or [32]). This property ensures that, if the server starts moving from p to q at time t, for all $t' \in [t, t + d(p, q)]$ there is a $x \in X$ such that the server is at position x at time t'. We call the class of metric spaces satisfying this property *continuous metric spaces*. Examples for continuous metric spaces are the real line and euclidean vector spaces $\mathbb{R}^k$ with $k \in \mathbb{N}$. Furthermore, every edge-weighted undirected graph induces a continuous metric space.

Of course different metric spaces allow to model different real-world problems. Indeed, online Dial-a-Ride is by far not limited to model the problem of conducting an elevator efficiently. If we, for example, take a street network as underlying metric space, Dial-a-Ride models the problem of a taxi driver efficiently serving customer requests.

We distinguish between an open and a closed variant of the problem: In the closed version of online Dial-a-Ride, the server is required to return to the origin after serving the last request of the request sequence, while in the open version, the server is allowed to stay at the destination of the last served request. Note that the information that all requests are released is only given implicitely by not releasing new requests anymore, i.e., in the closed version, the server might need to return to the origin prematurely to cover the case that the request sequence is fully released and to stay competitive.

An offline solution for an instance of Dial-a-Ride is defined by the behavior of the server including its trajectory inside the metric space and at which times requests are loaded and unloaded. At every point of time, the server either waits or moves with unit speed towards a certain position. Therefore, we can model the trajectory of the server as sequence of M tuples (q_i, x_i), where $q_i \in X$ is a position and $x_i \in \mathbb{R}_{\geq 0}$ is a waiting time at that position. We always have $q_1 = 0$ since the server starts at the origin and $x_M = 0$ since there is no waiting time after reaching the final position. In the closed variant of the problem, we additionally have $q_M = 0$. The server needs to serve all requests. To ensure this, we introduce the *loading matrix* $(L_j, U_j)_{j \in \{1,...,n\}}$ consisting of the loading and unloading times of the requests. This matrix has to obey some consistency rules: For all $j \in \{1, ..., n\}$ the inequality $L_j \leq U_j$ needs to hold, i.e., requests can only be unloaded if loaded prior. Furthermore, for all $j \in \{1, ..., n\}$ we need $|\{k \in \{1, ..., n\} \setminus \{j\} : L_j \in [L_k, U_k)\}| < c$, i.e., no request can be loaded, while c requests are already loaded. Finally, the matrix (L_j, U_j) needs to be compatible with the trajectory (q_i, x_i), i.e., the server needs to be at position a_j at time L_j and at position b_j at time U_j for all $j \in \{1, \ldots, n\}$. An offline solution for Dial-a-Ride is a trajectory with compatible loading matrix and we call offline solutions of Dial-a-Ride from now on *walks*. The completion time of a walk $(q_i, x_i)_{i \in \{1,...,M\}}$ is defined as

$$c((q_i, x_i)_{i \in \{1,...,M\}}) := \sum_{i=1}^{M-1} x_i + d(q_i, q_{i+1}).$$

Whenever it is clear from the context which request is served at which time, we omit the loading matrix (L_j, U_j) in our analysis. Furthermore, we denote by $q_1 \to \ldots \to q_m$ a walk that does not contain waiting times, i.e., that visits the points $q_1, \ldots, q_m \in X$ in the order defined by their indices, always taking the shortest way from q_k to q_{k+1} for $k \in \{1, \ldots, m-1\}$. We write

$$D(q_1 \to \ldots \to q_m) := d(q_1, q_2) + \cdots + d(q_{m-1}, q_m).$$

for the length of the walk $q_1 \to \ldots \to q_m$.

We start the examination of online Dial-a-Ride by showing that it can be modeled as an online optimization problem as defined in Definition 1.4:

- The set of input items $\mathcal{R}$ is the set of all requests $s = (a, b; r)$.
- The set of responses $\mathcal{A}$ is the set of all walks starting at the origin.
- The set of input sequences Σ is the set of all sequences of requests.
- The set of feasible responses $f_i(\sigma_{\leq i}, a_1, \ldots, a_{i-1})$ in step i is the set of all walks W that serve all requests of the subsequence $\sigma_{\leq i}$ while respecting the capacity limit c and the release times and being compatible with the actions of the server until step i, i.e., until time r_i. Compatible means that the walk the server has performed until time r_i is identical to the walk W and requests have been loaded and unloaded by the server exactly at the same times as in W.
- The cost function c_i of step i is mapping the currently chosen response, i.e., the currently chosen walk to its completion time.

In this thesis we will examine several versions of the online Dial-a-Ride problem. We will distinguish between online TSP and online Dial-a-Ride: Online TSP is a special case of online Dial-a-Ride that only allows requests where the starting position is identical to the destination, i.e., requests are served by just visiting their positions instead of having to transport them. Consequently, requests are tuples $(a; r)$ consisting only of a position $a \in X$ and a release time $r \in \mathbb{R}_{\geq 0}$ and there is no need to define a capacity. Note that, since online TSP is a special case of online Dial-a-Ride, every lower bound established for the competitive ratio of online TSP also holds for the more general online Dial-a-Ride. Similarly, every upper bound established for the competitive ratio of online Dial-a-Ride also holds for its special case online TSP.

Furthermore, we will make a distinction based on the underlying metric space: We will distinguish between the real line $\mathbb{R}$ and the general setting, i.e., arbitrary continuous metric spaces. Note that, since the real line is a continuous metric space, every lower bound established on the real line also holds for the general setting. Similarly, every upper bound established in the general setting in particualar holds on the real line. We orient the real line $\mathbb{R}$ from left to right and denote by

$$x^{\min} := \min\{0, a_1, \ldots, a_n, b_1, \ldots, b_n\}$$

the leftmost and by

$$x^{\max} := \max\{0, a_1, \ldots, a_n, b_1, \ldots, b_n\}$$

the rightmost position that needs to be visited by the server to serve the request sequence σ. Obviously, there is an optimum walk that only visits positions in $[x^{\min}, x^{\max}]$, and we let Opt be such a walk. Throughout this thesis, we assume the server to be *non-preemptive*, i.e., the server is allowed to unload requests only at their destinations.

2.1 State of the Art

The currently known best upper and lower bounds for the competitive ratios of the problems are given in Table 2.1. For online TSP, Bjelde et al. present conclusive results on the real line: For the closed version they provide a 1.6404-competitive online algorithm (see [13, Thm 3]) that matches the lower bound shown by Ausiello et al. in [8, Thm 3.3]. Furthermore, Bjelde et al. present a 2.0346-competitive algorithm for the open version of online TSP on the line (see [13, Thm 10]) that they complement with a matching lower bound (see [13, Thm 4]). Therefore, online TSP on the line is fully understood in terms of competitiveness. Moreover, the lower bound of 2.0346 for open online TSP on the line provided by Bjelde et al. is the best known lower bound for online Dial-a-Ride and online TSP in the general setting. In the same paper, Bjelde et al. also present a lower bound of 1.75 for the competitive ratio of closed online Dial-a-Ride on the line with $c < \infty$ (see [13, Thm 13]) and a 2.4142-competitive algorithm for open online Dial-a-Ride on general continuous metric spaces with infinite capacity (see [13, Thm 12]). Interestingly, the lower bound of 1.75 separates online Dial-a-Ride on the line with $c < \infty$ from online TSP on the line in terms of competitiveness.

The 2.4142-competitive algorithm returns to the origin and restarts a new optimum walk serving all unserved requests upon the release of a new request. It is presented in the paper as online algorithm for the preemptive version of open online Dial-a-Ride on the line (i.e., for the version of online Dial-a-Ride that allows to unload requests at any position and reload them at a later time). A close inspection of the proof however shows that their algorithm achieves the same result on general continuous metric spaces and for open online Dial-a-Ride with capacity $c = \infty$. The closed version of the problem on

	General Bounds	open		closed	
		upper bound	lower bound	upper bound	lower bound
line	Dial-a-Ride ($c < \infty$)	3.4142	2.0346	2	$\underline{1.75}$ [13, Thm 13]
line	Dial-a-Ride ($c = \infty$)	2.4142	2.0346	2	1.6404
line	TSP	$\underline{2.0346}$ [13, Thm 10]	$\underline{2.0346}$ [13, Thm 4]	$\underline{1.6404}$ [13, Thm 3]	$\underline{1.6404}$ [8, Thm 3.3]
general	Dial-a-Ride ($c < \infty$)	$\underline{3.4142}$ [32, Thm 2.30]	2.0346	$\underline{2}$ [5, Thm 6]	2
general	Dial-a-Ride ($c = \infty$)	$\underline{2.4142}$ [13, Thm 12]	2.0346	$\underline{2}$ [25, Thm 2.3]	2
general	TSP	2.4142	2.0346	$\underline{2}$ [8, Thm 4.2]	$\underline{2}$ [8, Thm 3.2]

Table 2.1: Overview of the best known bounds for the competitive ratios of online Dial-a-Ride on the line (top), and online Dial-a-Ride on general continuous metric spaces (bottom) excluding results of this thesis. Results are split into Dial-a-Ride with capacities $c < \infty$ and $c = \infty$ and TSP. Underlined results are original, all other results follow immediately.

continuous metric spaces is also fully understood in terms of competitiveness: Ausiello et al. provided a lower bound of 2 for the competitive ratio of online TSP on continuous metric spaces using cyclic structure as underlying metric space (see [8, Thm 3.2]). They complemented their lower bound with a 2-competitive algorithm for online TSP (see [8, Thm 4.2]). Feuerstein and Stougie provided a 2-competitive algorithm for closed online Dial-a-Ride with unlimited capacity in [25, Thm 2.3].

Algorithm Smartstart

The best known upper bounds for the competitive ratio of closed and open online Dial-a-Ride with arbitrary capacity are provided by the algorithm Smartstart that was published by Ascheuer et al. in [5]. The algorithm is 2-competitive for closed online Dial-a-Ride in the general setting with arbitrary capacity (see [5, Thm 6]). Since this upper bound matches the general lower bound for online TSP in the general setting, Smartstart is the best possible online algorithm for closed online Dial-a-Ride and closed online TSP in the general setting. Additionally, in contrast to the real line, closed online TSP and Dial-a-Ride in the general setting cannot be separated in terms of competitiveness. Krumke

	SMARTSTART	open		closed	
		upper bound	lower bound	upper bound	lower bound
line	DIAL-A-RIDE ($c < \infty$)	3.4142	2.0346	2	$\underline{1.75}$ [13, Thm 13]
	DIAL-A-RIDE ($c = \infty$)	3.4142	2.0346	2	1.6404
	TSP	3.4142	$\underline{2.0346}$ [13, Thm 4]	2	$\underline{1.6404}$ [8, Thm 3.3]
general	DIAL-A-RIDE ($c \in \mathbb{N} \cup \{\infty\}$)	$\underline{3.4142}$ [32, Thm 2.30]	2.0346	$\underline{2}$ [5, Thm 6]	2
	TSP	3.4142	2.0346	2	$\underline{2}$ [8, Thm 3.2]

Table 2.2: Overview of the best known bounds for the competitive ratios of SMARTSTART for online DIAL-A-RIDE on the line (top), and online DIAL-A-RIDE on general continuous metric spaces (bottom) excluding the results of this thesis. Results are split into DIAL-A-RIDE with capacities $c < \infty$ and $c = \infty$ and TSP. Underlined results are original, all other results follow immediately.

showed in [32, Thm 2.30] that the open version of the SMARTSTART algorithm is roughly 3.4142-competitive, which is the best known upper bound for the competitive ratio of open online DIAL-A-RIDE on the line as well as in the general setting. The currently best known bounds for the competitive ratios of algorithm SMARTSTART are summarized in Table 2.2. Note that the best known lower bounds for the competitive ratio of SMARTSTART are identical to the best known general lower bounds, i.e., no thorough analysis of SMARTSTART's lower bounds has been conducted yet. The best known upper bounds for the competitive ratio of SMARTSTART are the same for the real line as well as the general setting and the same for online DIAL-A-RIDE as well as for online TSP. However, since none of the best known upper bounds (except the closed version in the general setting) match their corresponding lower bounds, it is currently unclear if restricting the setting to the real line or disallowing transportation requests has an impact on SMARTSTART's competitive ratio. Furthermore, it is not clear if choosing a specific capacity has an impact on SMARTSTART's competitive ratio.

Algorithm IGNORE

The algorithm IGNORE is similar to SMARTSTART, but much more simple in design. This algorithm was first published in [5]. In [5, Thm 4], Ascheuer et al. showed that IGNORE

	Ignore	open upper bound	open lower bound	closed upper bound	closed lower bound
line	Dial-a-Ride ($c < \infty$)	4	2.0346	2.5	$\underline{1.75}$ [13, Thm 13]
line	Dial-a-Ride ($c = \infty$)	4	2.0346	2.5	1.6404
line	TSP	4	$\underline{2.0346}$ [13, Thm 4]	2.5	$\underline{1.6404}$ [8, Thm 3.3]
general	Dial-a-Ride ($c \in \mathbb{N} \cup \{\infty\}$)	$\underline{4}$ [32, Thm 2.29]	2.0346	$\underline{2.5}$ [5, Thm 4]	2
general	TSP	4	2.0346	2.5	$\underline{2}$ [8, Thm 3.2]

Table 2.3: Overview of the best known bounds for the competitive ratios of Ignore for online Dial-a-Ride on the line (top), and online Dial-a-Ride on general continuous metric spaces (bottom) excluding the results of this thesis. Results are split into Dial-a-Ride with capacities $c < \infty$ and $c = \infty$ and TSP. Underlined results are original, all other results follow immediately.

is 2.5-competitive for closed online Dial-a-Ride in the general setting. Later, Krumke, one of the authors of [5], analyzed Ignore for the open version of online Dial-a-Ride in the general setting in his PhD thesis: He showed that the algorithm is 4-competitive [32, Thm 2.29]. Similar to Smartstart, no lower bound analysis for Ignore's competitive ratio has been conducted yet. Consequently, the lower bounds for Ignore's competitive ratio are identical to the best known general lower bounds. Furthermore, again similar to Smartstart, the upper bounds for the competitive ratio of Ignore are the same for the real line as well as the general setting and the same for online Dial-a-Ride as well as for online TSP. Additionally, the best known upper bounds for Ignore's competitive ratio are independent of the capacity of the server. However, since none of the upper bounds match their corresponding lower bounds, it is currently unclear if restricting the setting to the real line, disallowing transportation requests or choosing a specific capacity has an impact on Ignore's competitive ratio. The currently best known upper and lower bounds for the competitive ratios of algorithm Ignore are summarized in Table 2.3.

Algorithm Replan

The last algorithm, we examine in this thesis, is the algorithm Replan. This algorithm was first presented in [5]. Ascheuer et al. showed that the algorithm is 2.5-competitive for

	REPLAN	open		closed	
		upper bound	lower bound	upper bound	lower bound
line	DIAL-A-RIDE ($c = 1$)	3	2.5	2.5	$\underline{1.75}$ [13, Thm 13]
	DIAL-A-RIDE ($1 < c < \infty$)	4.5	2.5	3.5	$\underline{1.75}$ [13, Thm 13]
	DIAL-A-RIDE ($c = \infty$)	4.5	2.5	3.5	1.6404
	TSP	2.5	$\underline{2.5}$ [8, Thm 4.1]	2.5	$\underline{1.6404}$ [8, Thm 3.3]
general	DIAL-A-RIDE ($c = 1$)	$\underline{3}$ [32, Thm 2.27]	2.5	$\underline{2.5}$ [5, Thm 3]	2
	DIAL-A-RIDE ($c > 1$)	$\underline{4.5}$ [32, Thm 2.28]	2.5	$\underline{3.5}$ [32, Thm 2.14]	2
	TSP	$\underline{2.5}$ [8, Thm 4.1]	2.5	2.5	$\underline{2}$ [8, Thm 3.2]

Table 2.4: Overview of the best known bounds for the competitive ratios of REPLAN for online DIAL-A-RIDE on the line (top), and online DIAL-A-RIDE on general continuous metric spaces (bottom) excluding results of this thesis. Results are split into DIAL-A-RIDE with capacities $c = 1$, $1 < c < \infty$ and $c = \infty$ and TSP. Underlined results are original, all other results follow immediately.

closed online DIAL-A-RIDE with capacity $c = 1$ (see [5, Thm 3]). Krumke, one of the authors of [5], examined the algorithms more thoroughly in his PhD Thesis [32]. He showed that the algorithm is 3.5-competitive for closed online DIAL-A-RIDE with capacity $c > 1$ (see [32, Thm 2.14]) and 3-competitive for open online DIAL-A-RIDE with capacity $c = 1$ (see [32, Thm 2.27]) as well as 4.5-competitive for open online DIAL-A-RIDE with capacity $c > 1$ (see [32, Thm 2.28]). For open online TSP, Ausiello et al. showed that the algorithm has a tight competitive ratio of 2.5 (see [8, Thm 4.1]). REPLAN was the best known online algorithm for open online TSP in the general setting until Bjelde et al. published a 2.4142-competitive algorithm in [13]. No lower bound analysis for REPLAN's competitive ratio in the closed version has been conducted yet. Consequently, the lower bounds for REPLAN's competitive ratio in the closed version are identical to the general lower bounds. Even though we have different upper bounds for different capacities, it is currently unclear if choosing a specific capacity has an impact on REPLAN's competitive ratio. Similarly, it is currently unclear if restricting to the real line or disallowing transportation requests

	Schedule-based	open		closed	
		upper bound	lower bound	upper bound	lower bound
line	DIAL-A-RIDE ($c \in \mathbb{N} \cup \{\infty\}$)	**2.6662** Thm 6.5 [12]	**2.5** Thm 4.1	2	**2**
line	TSP	**2.6288** Thm 6.23	**2.3333** Thm 4.2	2	**2** Thm 4.3
general	DIAL-A-RIDE ($c \in \mathbb{N} \cup \{\infty\}$)	**2.6956** Thm 6.36	**2.5**	2 [5, Thm 6]	2
general	TSP	**2.6625** Thm 6.37	**2.3333**	2	2 [8, Thm 3.2]

Table 2.5: Overview of the best known bounds for the competitive ratios of schedule-based algorithms for online DIAL-A-RIDE on the line (top), and online DIAL-A-RIDE on general continuous metric spaces (bottom). Results are split into DIAL-A-RIDE and TSP. Bold results are our contribution. Underlined results are original and imply the other results.

has an impact on REPLAN's competitive ratio. The currently best known bounds for the competitive ratios of algorithm REPLAN are summarized in Table 2.4.

2.2 Our Contribution

In this thesis, we provide a thorough analysis of the algorithms mentioned above. In particular, we take a more abstract point of view on the algorithms IGNORE and SMARTSTART and identify similarities between them. Using this insight, we define the class of *schedule-based algorithms* and identify both algorithms as elements of this class. We provide a lower bound of 2.5 for the competitive ratio of schedule-based algorithms for open online DIAL-A-RIDE on the line (see Thm 4.1) and a lower bound of roughly 2.3333 for open online TSP on the line (see Thm 4.2). For the closed online TSP on the line we provide a lower bound of 2 (see Thm 4.3). A detailed examination of the properties of schedule-based algorithms is given in the first two sections of Chapter 4. A summary of results for schedule-based algorithms is given in Table 2.5.

Algorithm IGNORE

For the algorithm IGNORE, we complement the upper bound of 4 for its competitive ratio for open online DIAL-A-RIDE (see [32, Thm 2.29]) with a matching lower bound (see

	IGNORE	open		closed	
		upper bound	lower bound	upper bound	lower bound
line & gen.	DIAL-A-RIDE ($c \in \mathbb{N} \cup \{\infty\}$)	4 [32, Thm 2.29]	**4** **Thm 4.12 [10]**	2.5 [5, Thm 4]	**2.5**
	TSP	**3.5** **Thm 4.13**	**3** **Thm 4.14**	2.5	**2.5** **Thm 4.11**

Table 2.6: Overview of the best known bounds for the competitive ratios of IGNORE for online DIAL-A-RIDE and online TSP. Bold results are our contribution. Underlined results are original and imply the other results.

Thm 4.12). We published this lower bound also in [10, Prop A.1]. For closed online TSP, we provide a lower bound of 2.5 (see Thm 4.11), matching the already known upper bound for closed online DIAL-A-RIDE (see [5, Thm 4]). Note that this proves that restricting the underlying metric space to the real line or choosing a specific capacity of the server has no impact on the competitive ratio of IGNORE for online DIAL-A-RIDE. Furthermore, the closed version of IGNORE attains the same competitive ratio for online DIAL-A-RIDE as for online TSP. However, for the open version of online TSP we provide an improved upper bound of 3.5 (see Thm 4.13) for the competitive ratio of IGNORE, which is strictly lower than the lower bound for the open version of online DIAL-A-RIDE. We complement this upper bound with a lower bound of 3 for the competitive ratio of IGNORE for open online TSP (see Thm 4.14). The exact competitive ratio of IGNORE for open online TSP remains unknown. Consequently, it also remains unclear whether restricting the setting to the real line or choosing a specific capacity for the server has an impact on the competitive ratio of IGNORE for open online TSP. A detailed examination of IGNORE is given in Section 4.3 and summary of all results for algorithm IGNORE is given in Table 2.6.

Algorithm SMARTSTART

As IGNORE before, SMARTSTART is also a schedule-based algorithm. Therefore, the already mentioned lower bounds for schedule-based algorithms are also lower bounds for SMARTSTART's competitive ratio. In particular, the lower bound of 2 for the competitive ratio of schedule-based algorithms for closed online TSP on the line (see Thm 4.3) matches the known upper bound for the competitive ratio of SMARTSTART for closed online DIAL-A-RIDE (see [5, Thm 6]). Consequently, SMARTSTART is a best-possible schedule-based algorithm for the closed version of online TSP and online DIAL-A-RIDE even on the real line. For open online DIAL-A-RIDE on the line we provide an improved upper bound for the competitive

	SMARTSTART	open		closed	
		upper bound	lower bound	upper bound	lower bound
line	DIAL-A-RIDE ($c \in \mathbb{N} \cup \{\infty\}$)	**2.9377** Thm 5.5 [10]	**2.9377** Thm 5.25 [10]	2	**2**
line	TSP	**2.7604** Thm 5.29	**2.7604** Thm 5.40	2	**2** Thm 4.3
general	DIAL-A-RIDE ($c \in \mathbb{N} \cup \{\infty\}$)	**3** Thm 5.43	**2.9377**	2 [5, Thm 6]	2
general	TSP	**2.8229** Thm 5.44	**2.7604**	2	2 [8, Thm 3.2]

Table 2.7: Overview of the best known bounds for the competitive ratios of SMARTSTART for online DIAL-A-RIDE on the line (top), and online DIAL-A-RIDE on general continuous metric spaces (bottom). Results are split into DIAL-A-RIDE and TSP. Bold results are our contribution. Underlined results are original and imply the other results.

ratio of SMARTSTART of roughly 2.9377 (see Thm 5.5), which we complement with a matching lower bound (see Thm 5.25). We published these results also in [10, Thm 3.8, Thm 4.9]. Interestingly, for open online TSP on the line, we provide a slightly stronger upper bound for the competitive ratio of SMARTSTART of roughly 2.7604 (see Thm 5.29), which we also complement with a matching lower bound (see Thm 5.40). This proves that choosing a specific capacity for the server has no impact on SMARTSTART's competitive ratio in the open setting on the line, while disallowing transportation request has. For arbitrary continuous metric spaces, we obtain slightly weaker upper bounds: We show that SMARTSTART is 3-competitive for the open version of online DIAL-A-RIDE (see Thm 5.43) and 2.8229-competitive for the open version of online TSP (see Thm 5.44). The lower bounds obtained on the real line remain for the general setting, i.e., it remains unclear if restricting the underlying metric space to the real line has an impact on SMARTSTART's competitive ratio. A detailed examination of SMARTSTART is presented in Chapter 5 and a summary of results is given in Table 2.7.

Algorithm SMARTERSTART

During the analysis of SMARTSTART, we obtain conclusive insights on its strengths and weaknesses. By avoiding a critical weakness of SMARTSTART, we are able to develop an improved schedule-based algorithm called SMARTERSTART. We show that SMARTERSTART is 2-competitive for closed online DIAL-A-RIDE (see Thm 6.40), matching the lower bound

	SMARTERSTART	open upper bound	open lower bound	closed upper bound	closed lower bound
line	DIAL-A-RIDE ($c \in \mathbb{N} \cup \{\infty\}$)	**2.6662** Thm 6.5 [12]	**2.6662** Thm 6.19 [12]	**2**	**2**
line	TSP	**2.6288** Thm 6.23	**2.6288** Thm 6.33	**2**	**2** Thm 4.3
general	DIAL-A-RIDE ($c \in \mathbb{N} \cup \{\infty\}$)	**2.6956** Thm 6.36	**2.6662**	**2** Thm 6.40	2
general	TSP	**2.6625** Thm 6.37	**2.6288**	**2**	2 [8, Thm 3.2]

Table 2.8: Overview of the best known bounds for the competitive ratios of SMARTERSTART for online DIAL-A-RIDE on the line (top), and online DIAL-A-RIDE on general continuous metric spaces (bottom). Results are split into DIAL-A-RIDE and TSP. Bold results are our contribution. Underlined results are original and imply the other results.

for schedule-based algorithms for closed online TSP on the line (see Thm 4.3). Thus, as SMARTSTART, algorithm SMARTERSTART is a best-possible schedule-based algorithm for closed online DIAL-A-RIDE and closed online TSP. For open online DIAL-A-RIDE on the line we provide an upper bound for the competitive ratio of SMARTERSTART of roughly 2.6662 (see Thm 6.5), which we complement with a matching lower bound (see Thm 6.19). This is a significant improvement in comparison to the 2.9377-competitive SMARTSTART algorithm. We published these results also in [12, Thm 3.7, Thm 1.2]. For open online TSP on the line we provide a slightly stronger upper bound for the competitive ratio of SMARTERSTART of roughly 2.6288 (see Thm 6.23), which we complement with a matching lower bound (see Thm 6.33). This proves that choosing a specific capacity for the server has no impact on SMARTSTART's competitive ratio in the open setting on the line, while disallowing transportation request has. For arbitrary continuous metric spaces, we obtain slightly weaker upper bounds: We show that SMARTERSTART is 2.6956-competitive for the open version of online DIAL-A-RIDE (see Thm 6.36) and 2.6625-competitive for the open version of online TSP (see Thm 6.37). The lower bounds obtained on the real line remain for the general setting, i.e., it remains unclear if restricting the underlying metric space to the real line has an impact on SMARTERSTART's competitive ratio. A detailed examination of SMARTERSTART is presented in Chapter 6 and a summary of results including the results of this thesis is given in Table 2.8.

	Replan	open upper bound	open lower bound	closed upper bound	closed lower bound
line	Dial-a-Ride $(c = 1)$	3	2.5	2.5	**2**
line	Dial-a-Ride $(1 < c < \infty)$	**4**	2.5	**3** **Thm 7.6**	**2**
line	Dial-a-Ride $(c = \infty)$	**3**	2.5	**2.5**	**2**
line	TSP	2.5	2.5 [8, Thm 4.1]	**2** **Thm 7.5**	**2** **Thm 7.4**
general	Dial-a-Ride $(c = 1)$	3 [32, Thm 2.27]	2.5	2.5 [32, Thm 2.15]	2
general	Dial-a-Ride $(1 < c < \infty)$	**4** **Thm 7.8**	2.5	3.5 [32, Thm 2.14]	2
general	Dial-a-Ride $(c = \infty)$	**3** **Thm 7.9**	2.5	**2.5** **Thm 7.7**	2
general	TSP	2.5 [8, Thm 4.1]	2.5	2.5	2 [8, Thm 3.2]

Table 2.9: Overview of the best known bounds for the competitive ratios of Replan for online Dial-a-Ride on the line (top), and online Dial-a-Ride on general continuous metric spaces (bottom). Results are split into Dial-a-Ride with capacities $c = 1$, $1 < c < \infty$ and $c = \infty$ and TSP. Bold results are our contribution. Underlined results are original and imply the other results.

Algorithm Replan

Algorithm Replan is the only online algorithm examined in this thesis that is not schedule-based. We present a lower bound of 2 for the competitive ratio of Replan for closed online TSP on the line (see Thm 7.4), which we complement with a matching upper bound for the closed version on the line (see Thm 7.5). The upper bound for the closed TSP version of Replan in the general setting remains 2.5. For closed online Dial-a-Ride, we provide an upper bound of 3 for capacity $c > 1$ on the line (see Thm 7.6) and an upper bound of 2.5 for capacity $c = \infty$ in the general setting (see Thm 7.7). For the open version of online Dial-a-Ride, we improve Krumke's upper bound of 4.5 for capacity $c > 1$ to a bound of 4 for capacity $1 < c < \infty$ (see Thm 7.8) and to a bound of 3 for capacity $c = \infty$ (Thm 7.9). However, even though we improve several upper and lower bounds for the competitive ratios of different versions of Replan, we only show a tight competitive ratio

	General Bounds	open		closed	
		upper bound	lower bound	upper bound	lower bound
line	Dial-a-Ride ($c < \infty$)	**2.6662** **Thm 6.5 [12]**	**2.0585** **Thm 3.2 [12]**	2	**1.7636** **Thm 3.1**
line	Dial-a-Ride ($c = \infty$)	2.4142	2.0346	2	1.6404
line	TSP	2.0346 [13, Thm 10]	2.0346 [13, Thm 4]	1.6404 [13, Thm 3]	1.6404 [8, Thm 3.3]
general	Dial-a-Ride ($c < \infty$)	**2.6956** **Thm 6.36**	**2.0585**	2 [5, Thm 6]	2
general	Dial-a-Ride ($c = \infty$)	2.4142 [13, Thm 12]	2.0346	2 [25, Thm 2.3]	2
general	TSP	2.4142	2.0346	2 [8, Thm 4.2]	2 [8, Thm 3.2]

Table 2.10: Overview of the best known bounds for the competitive ratios of online Dial-a-Ride on the line (top), and online Dial-a-Ride on general continuous metric spaces (bottom). Results are split into Dial-a-Ride with capacities $c < \infty$ and $c = \infty$ and TSP. Bold results are our contribution. Underlined results are original and imply the other results.

for closed online TSP on the line. For all other versions of the problem, gaps between upper and lower bounds remain. Consequently, it remains unclear if Replan has different competitive ratios for online Dial-a-Ride and online TSP and if restricting the underlying metric space to the real line has an impact on its competitive ratio. Furthermore, it remains unclear if choosing a specific capacity has an impact on the competitive ratio. A thorough examination of Replan in conducted in Chapter 7 and an overview of our results is given in Table 2.9.

General Bounds

The analysis of the SmarterStart algorithm leads to several now improved general upper bounds for the competitive ratio of the problem online Dial-a-Ride. In particular, for open online Dial-a-Ride we improve Krumke's upper bound of roughly 3.4142 provided by SmartStart (see [32, Thm 2.30]) to an upper bound of roughly 2.6662 on the line (see Thm 6.5, also published in [12]) and to roughly 2.6956 in the general setting (see Thm 6.36). Besides providing analyses of online algorithms, we also present improved general lower bounds for online Dial-a-Ride on the line. For closed online Dial-a-Ride on the line with capacity $c < \infty$, we provide an improved lower bound of roughly 1.7636 (see Thm 3.1) and for open online DIAL-A-RIDE on the line with capacity $c < \infty$ we provide a lower bound of roughly 2.0585 (see Thm 3.2, also published in [12]). The latter separates open online DIAL-A-RIDE on the line from open online TSP on the line in terms of competitiveness. The lower bounds for online DIAL-A-RIDE with infinite capacity remain. See Chapter 3 for a detailed discussion of the lower bound constructions.

2.3 Related Work

Many different versions and modification of online DIAL-A-RIDE have been studied in the past. As already mentioned, in this thesis, we focus on the non-preemptive variant of online DIAL-A-RIDE, where requests cannot be unloaded on the way in reaction to the arrival of new requests. For the closed version in the case where preemption is allowed, we have a tight competitive ratio of 2 in the general setting and bounds of $[1.64, 2]$ on the line. The lower bounds are implications of the lower bounds for closed online TSP on the line [8, Thm 3.3] and in the general setting [8, Thm 3.2] and the best algorithm for closed preemptive online DIAL-A-RIDE is the SMARTSTART algorithm for closed non-preemptive online DIAL-A-RIDE [5, Thm 6]. Thus, the preemptive version is neither separated from online TSP nor is it separated from non-preemptive online DIAL-A-RIDE on the line. The same is true for open preemptive DIAL-A-RIDE on the line as well as in the general setting: The best bounds for the open, preemptive variant are $[2.04, 2.41]$ (see [13, Thm 4] and [13, Thm 12]), where the lower bound is again an implication of the lower bound for open online TSP on the line. The problem can be further modified by weakening the adversary. A adversary is called *fair*, if it only releases requests such that the optimum server cannot leave the convex hull of the origin and the positions of the currently released requests. Blom et al. introduced this adversary model and provided a lower bound of $\frac{5+\sqrt{57}}{8} \approx 1.57$ [14] for the competitive ratio of closed online TSP that was complemented by a matching online algorithm published by Lipmann [35].

A randomized version of closed online TSP was examined by Chen et al. [17]. They show a tight competitive ratio of 1.5. In case of a fair adversary they present a lower bound of $\frac{1+\sqrt{17}}{4} \approx 1.28$, which they complement with an upper bound of $\frac{9+\sqrt{177}}{16} \approx 1.39$.

Besides these minor modifications of the problem, many variants with more involved modifications have been studied. A variant of the online DIAL-A-RIDE problem where the objective is to minimize the maximal flow time, instead of the makespan, has been studied by Krumke et al. [33, 34]. They established that in many metric spaces no online algorithm can be competitive with respect to this objective. Hauptmeier et al. [29] showed that a competitive algorithm is possible if we restrict ourselves to instances with “reasonable” load. They defined an instance as reasonable if requests that appear over a sufficiently large time period T can always be served in time at most T. Lipmann et al. [36] studied a natural variant of closed, online Dial-a-Ride where the destinations of requests are only revealed upon collection by the server. For general metric spaces and server capacity c, they showed a tight competitive ratio of 3 in the preemptive setting, and lower/upper bounds of $\max\{3.12, c\}$ and $2c + 2$, respectively, in the non-preemptive setting. Yi and Tian [45] considered the online Dial-a-Ride problem with deadlines, with the objective of serving the maximum number of requests. They provided bounds for the competitive ratio depending on the diameter of the metric space. In [46] they further studied this setting where the destinations of requests are only revealed upon collection by the server.

The offline version of Dial-a-Ride on the line has been studied in various settings. An overview has been provided in [41]. For the closed, non-preemptive case without release times, Gilmore and Gomory [27] and Atallah and Kosaraju [6] gave a polynomial time algorithm for a server with unit capacity $c = 1$, and Guan [28] showed that the problem is hard for $c = 2$. Bjelde et al. [13] extended this result to any finite $c \geq 2$ and both the open and closed case. They further showed that, with release times the, problem is already hard for finite $c \geq 1$. On the other hand, the complexity of the case $c = \infty$ has not yet been established. The closed, preemptive case without release times was shown to be polynomial time solvable for $c = 1$ by Atallah and Kosaraju [6], and for $c \geq 2$ by Guan [28]. For the closed, non-preemptive case with finite capacity, Krumke [32] provided a 3-approximation algorithm. Finally, Charikar and Raghavachari [16] gave approximation algorithms for the closed case without release times, both preemptive and non-preemptive, on general metric spaces.

3 Lower Bounds for Online Dial-a-Ride

In this chapter, we present two new lower bounds for online Dial-a-Ride. On one hand we will present a lower bound of 1.7636 for the closed version of online Dial-a-Ride on the line and on the other hand we provide a lower bound of 2.0585 for the open version of online Dial-a-Ride on the line. To be more precise we prove the following results.

Theorem 3.1. *Let $\rho_{\text{cl}} \approx 1.7636$ be the largest root of the polynomial $-14\rho^2 + 40\rho - 27$. There is no $(\rho_{\text{cl}} - \varepsilon)$-competitive algorithm for closed online* Dial-a-Ride *on the line with capacity $c < \infty$ for any $\varepsilon > 0$.*

Theorem 3.2. *Let $\rho_{\text{op}} \approx 2.0585$ be the second largest root of the polynomial $4\rho^3 - 26\rho^2 + 39\rho - 5$. There is no $(\rho_{\text{op}} - \varepsilon)$-competitive algorithm for open online* Dial-a-Ride *on the line with capacity $c < \infty$ for any $\varepsilon > 0$.*

The latter result separates the open versions of online Dial-a-Ride and online TSP on the line in terms of competitiveness since open online TSP on the line has a tight competitive ratio of 2.0346 [13, Thm 10]. For closed online Dial-a-Ride and online TSP on the line this separation was already established in [13]. Bjelde et al. showed a tight competitive ratio of roughly 1.6404 for closed online TSP on the line and improved the lower bound of online Dial-a-Ride to 1.75. Interestingly, this kind of separation is not possible in the general setting: Ausiello et al. were able to show in [8] that online TSP on an arbitrary continuous metric space has a tight competitive ratio of 2. This directly implies a lower bound of 2 for online Dial-a-Ride on general metric spaces, which Ascheuer et al. complemented with the 2-competitive Smartstart algorithm in [5]. See Table 2.1 in Chapter 2 for a full list of currently known bounds excluding the results of this thesis. We published Theorem 3.2 and its proof also in [12].

Preparations

The proof of Theorem 3.1 and the proof of Theorem 3.2 rely on constructing a request sequence that consists of two *stages*. The intuitive idea of these two stages is the following: The first stage forces the algorithm Alg to be in some critical situation and the second stage then exploits this situation and forces Alg to be at best ρ-competitive for a certain $\rho > 1$.

To simplify the exposition a bit, consider the situation in which the server is fully loaded with requests that have the same destination: Let ALG' be a online algorithm for online DIAL-A-RIDE. We call ALG' *eager* if, in the case that ALG' is fully loaded with requests that have the same destination, it immediately delivers all loaded requests without detour. It is clear that we can transform every algorithm ALG$''$ into an eager algorithm ALG$''_{\text{eager}}$ by letting it act exactly as ALG$''$, except if it is fully loaded. In the case it is fully loaded, we let ALG$''_{\text{eager}}$ deliver the requests right away, wait until ALG$''$ would have delivered them, and then let it continue like ALG$''$. Since ALG$''$ cannot collect or serve other requests while being fully loaded, we have ALG$''_{\text{eager}}(\sigma) \leq$ ALG$''(\sigma)$ for every request sequence σ.

Observation 3.3. *Every algorithm for online* DIAL-A-RIDE *can be turned into an eager algorithm with the same competitive ratio.*

Thus, we may assume in the following that the algorithm ALG is eager. Furthermore, in the following, we use the notation "move(a)" to describe the trajectory of a server for the tour that moves the server from its current position with unit speed to the position $a \in \mathbb{R}$. By pos(t) we denote ALG's position at time t.

3.1 Lower Bound for Closed Online DIAL-A-RIDE on the Line

In this section, we prove Theorem 3.1. Let $c < \infty$ and ALG be an algorithm for closed online DIAL-A-RIDE. Let $\rho_{\text{cl}} \approx 1.7636$ be the largest root of the polynomial $-14\rho^2+40\rho-27$. We describe a request sequence $\sigma_{\rho_{\text{cl}}}$ such that

$$\text{ALG}(\sigma_{\rho_{\text{cl}}}) \geq \rho_{\text{cl}}\text{OPT}(\sigma_{\rho_{\text{cl}}}).$$

We first give a more intuitive description of our construction omitting most technical details. As already mentioned, our construction consists of two stages, with the first stage ending when a critical situation for ALG is established. We start by describing this critical situation and how to exploit it, i.e., we first explain the second stage.

Suppose we have c times the request $s_R^j = (r_R, 0; r_R)$ and c times the request $s_L^j = (-r_L, 0; r_L)$ with $j \in \{1, \ldots, c\}$. We assume that ALG loads all c requests s_R^j at once before any of the c requests s_L^j at some time $t_R^* \geq (2\rho_{\text{cl}} - 2)r_L + (2\rho_{\text{cl}} - 3)r_R$. Then we could just release the request $s_R^* = (r_R, r_R; 2r_L + r_R)$ and we would have

$$\begin{aligned} \text{ALG}(\sigma_{\rho_{\text{cl}}}) &= t_R^* + 2r_L + 3r_R \\ &\geq (2\rho_{\text{cl}} - 2)r_L + (2\rho_{\text{cl}} - 3)r_R + 2r_L + 3r_R \\ &= 2\rho_{\text{cl}}r_L + 2\rho_{\text{cl}}r_R \end{aligned}$$

$$= \rho_{\text{cl}}\textsc{Opt}(\sigma_{\rho_{\text{cl}}}),$$

since OPT can serve the $2c+1$ requests in time $2r_L + 2r_L$ by serving the requests s_L^j first. In fact, we can force ALG into this situation, if the requests $s_R^j = (r_R, r_R; r_R)$ and $s_L^j = (-r_L, -r_L; r_L)$ satisfy the following properties.

Definition 3.4. Let $\rho > 1$. We call the last $2c$ requests $s_R^j = (r_R, r_R; r_R)$ and $s_L^j = (-r_L, -r_L; r_L)$ with $j \in \{1, \dots, c\}$ of a request sequence σ with $0 < r_L \leq r_R$ *c-ρ-critical* for ALG if the following conditions hold:

(i) Both tours

$$\text{move}(-r_L) \oplus \text{move}(r_R) \oplus \text{move}(0)$$

and

$$\text{move}(r_R) \oplus \text{move}(-r_L) \oplus \text{move}(0)$$

serve all requests presented until time r_R.

(ii) If ALG loads at least one request s_R^j before it loads any of the requests s_L^j, it does it no earlier than $t_R^* := (2\rho - 2)r_L + (2\rho - 3)r_R$.

(iii) If ALG loads at least one request s_L^j before it loads any of the requests s_R^j, it does it no earlier than $t_L^* := (2\rho - 2)r_R + (2\rho - 3)r_L$.

(iv) It holds that $\frac{r_R}{r_L} \leq \frac{2\rho^2 - 6\rho + 4}{-2\rho^2 + 4\rho - 1}$.

Analysis of the Second Stage

Definition 3.4 describes the critical situation ALG is forced into at the end of the first stage. There are two things we have to prove: We can present a request sequence that satisfies Definition 3.4 (first stage), and, once ALG is in the critical situation described by Definition 3.4, we can release additional requests, such that ALG is not better than ρ-competitive (second stage). We first show the latter.

Lemma 3.5. *Let $\rho \in (\frac{1}{2}(2 + \sqrt{2}), \frac{1}{4}(5 + \sqrt{5})] \approx (1.7071, 1.8090]$. If there is a request sequence with $2c$ c-ρ-critical requests for* ALG, *we can release additional requests such that* ALG *is not $(\rho - \varepsilon)$-competitive on the resulting instance for any $\varepsilon > 0$.*

Proof. Let σ_ρ be a request sequence with c-ρ-critical requests s_L^j and s_R^j according to Definition 3.4. First, we note that Definition 3.4 implies $0 < r_L \leq r_R$, i.e., Definition 3.4 (iv) can only hold if

$$1 \leq \frac{r_R}{r_L} \leq \frac{2\rho^2 - 6\rho + 4}{-2\rho^2 + 4\rho - 1},$$

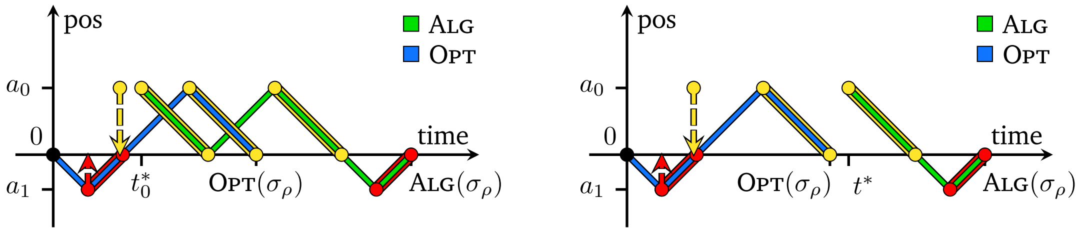

Figure 3.1: Left: ALG does not serve all requests s_0^j (yellow ○) at once and moves the distance from the origin to a_0 three times. Right: ALG waits too long before serving the requests s_0^j, i.e. t^* gets too large. We only show ALG's walk after time t_0^* and t^*. OPT is blue, the requests s_1^j are red ● and $\rho = \rho_{\text{cl}}$.

i.e., if $\rho \in (\frac{1}{2}(2+\sqrt{2}), \frac{1}{4}(5+\sqrt{5})]$ holds. Let $a_0 \in \{-r_L, r_R\}$ be the starting position of the requests of the subsequence $(s_0^j)_j \in \{(s_L^j)_j, (s_R^j)_j\}$ that are collected first by ALG and let $a_1 \in \{-r_L, r_R\}$ be the starting position of the requests $(s_1^j)_j \in \{(s_L^j)_j, (s_R^j)_j\}$ that are on the opposite side of the origin. By properties (ii) and (iii) of Definition 3.4, ALG cannot collect any request s_0^j before time $t_0^* := (2\rho-2)|a_1| + (2\rho-3)|a_0|$. Note that $\rho > 1.7071$ implies $t_0^* > 0$. Assume ALG only serves $c' < c$ of the requests s_0^j before loading the remaining $c - c'$. See the left part of Figure 3.1 for an illustration of this case. We have

$$\text{ALG}(\sigma_\rho) \geq t_0^* + 2|a_1| + 3|a_0| = 2\rho|a_1| + 2\rho|a_0| = \rho\text{OPT}(\sigma_\rho),$$

since all requests can be served with the tour move$(a_1) \oplus$ move$(a_0) \oplus$ move(0) according to (i).

Thus, we may assume that ALG loads all c requests s_0^j at once at some time t^*. If no additional requests are released, we have

$$\text{ALG}(\sigma_\rho) \geq t^* + |a_0| + 2|a_1|$$

and

$$\text{OPT}(\sigma_\rho) = 2|a_1| + 2|a_0|$$

again by (i). Thus, the claim is true if

$$t^* + |a_0| + 2|a_1| \geq 2\rho|a_1| + 2\rho|a_0|, \qquad \text{i.e.,} \qquad t^* \geq (2\rho-2)|a_1| + (2\rho-1)|a_0|$$

holds. See the right part of Figure 3.1 for an illustration of this case. If we have $t^* < (2\rho-2)|a_1| + (2\rho-1)|a_0|$ we release the request

$$s_0^+ := (\text{sgn}(a_0)\max\{|a_0|, t^* - 2|a_1|\}, \text{sgn}(a_0)\max\{|a_0|, t^* - 2|a_1|\}; t^*)$$

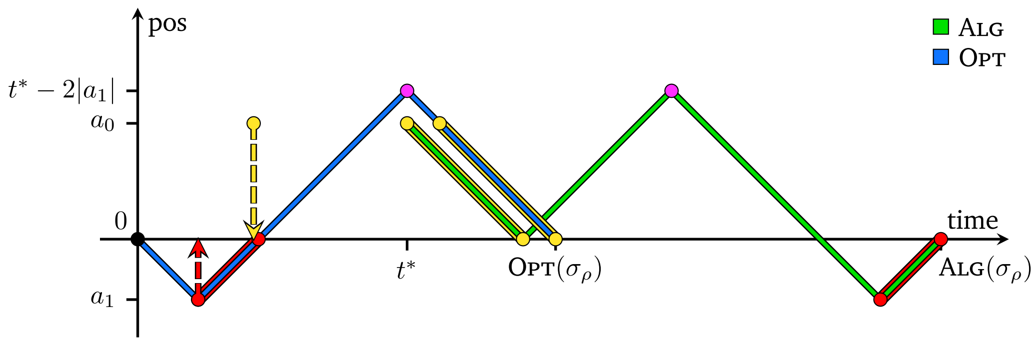

Figure 3.2: ALG (green) collects all s_0^j (yellow ○) at time $t^* = (2\rho - 2)|a_1| + (2\rho - 2)|a_0|$. This implies that the request s_0^* (violet ●) is released at position $t^* - 2|a_1|$ and after OPT reaches a_0. We only show ALG's walk after time t^*. OPT is blue, the requests s_1^j are red ● and $\rho = \rho_{\text{cl}}$.

and define σ_ρ^+ to be the request sequence σ_ρ plus the request s_0^+. Since ALG has loaded the requests s_0^j and is fully loaded, it has to deliver the loaded requests before being able to serve s_0^+. Thus, we have

$$\text{ALG}(\sigma_\rho^+) \geq t^* + |a_0| + 2\max\{|a_0|, t^* - 2|a_1|\} + 2|a_1|.$$

OPT serves the requests s_1^j first and thus reaches position $\operatorname{sgn}(a_0)\max\{|a_0|, t^* - 2|a_1|\}$ at time t^* or before. Therefore, OPT does not need to wait for the release of s_0^+ and we have

$$\text{OPT}(\sigma_\rho^+) = 2|a_1| + 2\max\{|a_0|, t^* - 2|a_1|\}.$$

If we have $\max\{|a_0|, t^* - 2|a_1|\} = |a_0|$, the claim is true if

$$t^* + 3|a_0| + 2|a_1| \geq 2\rho|a_1| + 2\rho|a_0|, \qquad \text{i.e.,} \qquad t^* \geq (2\rho - 2)|a_1| + (2\rho - 3)|a_0|$$

holds, which is always the case because of (ii) and (iii). Otherwise, if $\max\{|a_0|, t^* - 2|a_1|\} = t^* - 2|a_1|$, the claim is true if

$$3t^* + |a_0| - 2|a_1| \geq 2\rho t^* - 2\rho|a_1|, \qquad \text{i.e.,} \qquad t^* \leq \frac{2\rho - 2}{2\rho - 3}|a_1| + \frac{1}{2\rho - 3}|a_0| \tag{3.1}$$

holds. An illustration of this last case is given in Figure 3.2. The inequality (3.1) holds for all $t^* < (2\rho - 2)|a_1| + (2\rho - 1)|a_0|$ if

$$\frac{2\rho - 2}{2\rho - 3}|a_1| + \frac{1}{2\rho - 3}|a_0| \geq (2\rho - 2)|a_1| + (2\rho - 1)|a_0|.$$

This is equivalent to

$$\frac{|a_0|}{|a_1|} \leq \frac{2\rho^2 - 6\rho + 4}{-2\rho^2 + 4\rho - 1},$$

which holds because of (iv). □

Analysis of the First Stage

Thus, what remains is to construct a request sequence $\sigma'_{\rho_{\text{cl}}}$ that satisfies all properties of Definition 3.4.

We let ALG wait until time 1. Without loss of generality, we assume that ALG's position at time 1 is $\text{pos}(1) \leq 0$ (the other case is symmetric). Now, let the request $s_R^0 = (1, 1; 1)$ appear. ALG cannot serve s_R^0 before time 2. If ALG serves s_R^0 after time $2\rho_{\text{cl}} - 1$ it is not ρ_{cl}-competitive, since

$$\text{ALG}((s_R^0)) \geq 2\rho_{\text{cl}} - 1 + 1 = 2\rho_{\text{cl}} = \rho_{\text{cl}}\text{OPT}((s_R^0)).$$

Let $r_L \in [2, 2\rho_{\text{cl}} - 1)$ be the time ALG serves s_R^0 and let c requests $s_L^j = (-r_L, 0; r_L)$ with $j \in \{1, \dots, c\}$ appear. We define the line

$$\ell_{\text{cl}}(t) = (5 - 2\rho_{\text{cl}}) \cdot t - (2\rho_{\text{cl}} - 2) \cdot r_L.$$

We have $r_L \in [2, 2\rho_{\text{cl}} - 1)$ and $\ell_{\text{cl}}(r_L) = (7 - 4\rho_{\text{cl}})r_L$. Since $\rho_{\text{cl}} > 1.75$, we have $\ell_{\text{cl}}(r_L) < 0 < \text{pos}(r_L)$, i.e., ALG's position at time r_L is above the line ℓ_{cl} in the position-time diagram. Thus, ALG crosses the line ℓ_{cl} before loading any of the c requests s_L^j. Let r_R be the time ALG crosses the line ℓ_{cl} for the first time after time r_L and let c requests $s_R^j = (r_R, 0; r_R)$ with $j \in \{1, \dots, c\}$ appear. We define $\sigma'_{\rho_{\text{cl}}} := (s_R^0, s_L^1, \dots, s_L^c, s_R^1, \dots, s_R^c)$.

Lemma 3.6. *Let $j \in \{1, \dots, c\}$. ALG can neither collect a request s_L^j before time t_L^* nor can it collect request s_R^j before time t_R^*.*

Proof. See Figure 3.3 for an illustration of ALG collecting the requests s_L^j after time t_L^* (left) and an illustration of ALG collecting the requests s_R^j at time t_R^* (right). Assume ALG serves the requests s_R^j before s_L^j. Then it does not collect any request s_R^j before time

$$r_R + |\ell_{\text{cl}}(r_R) - r_R| = 2r_R - \ell_{\text{cl}}(r_R) = (2\rho_{\text{cl}} - 2)r_L + (2\rho_{\text{cl}} - 3)r_R = t_R^*$$

where the first equality follows since ALG is to the left of position r_R at time r_R. If we have

$$\ell_{\text{cl}}(r_R) \geq (2\rho_{\text{cl}} - 3)r_R - (4 - 2\rho_{\text{cl}})r_L, \tag{3.2}$$

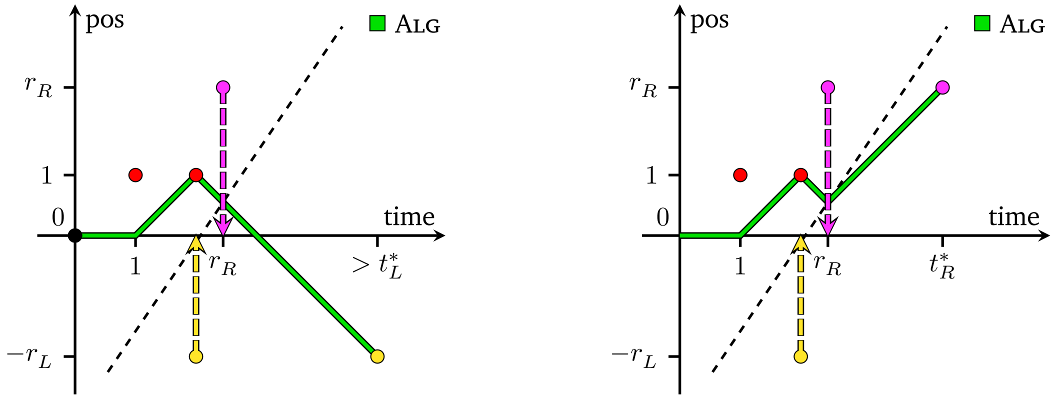

Figure 3.3: Left: ALG (green) collects s_L^j (yellow) before s_R^j (violet). The requests s_L^j are collected later than t_L^*. Right: ALG collects s_R^j before s_L^j. The requests s_R^j are not collected before t_R^*. Request s_R^0 is red and ℓ_{cl} is the dashed black line.

ALG cannot collect any request s_L^j before time

$$\begin{aligned} r_R + |\ell_{\text{cl}}(r_R) - (-r_L)| &= r_R + \ell_{\text{cl}}(r_R) + r_L \\ &\overset{(3.2)}{\geq} r_R + (2\rho_{\text{cl}} - 3)r_R - (4 - 2\rho_{\text{cl}})r_L + r_L \\ &= (2\rho_{\text{cl}} - 2)r_R + (2\rho_{\text{cl}} - 3)r_L = t_L^*. \end{aligned}$$

where the first equality follows since ALG is to the right of position $-r_L$ at time r_R. Thus, it is enough to show inequality (3.2). Inequality (3.2) holds for $r_R \geq \frac{2\rho_{\text{cl}}-3}{4-2\rho_{\text{cl}}} r_L$. ALG crosses ℓ_{cl} earliest if it moves towards position $-r_L$ directly after serving the request s_R^0. Thus, the earliest possible time ALG crosses ℓ_{cl} is the solution of

$$\ell_{\text{cl}}(r_R) = (5 - 2\rho_{\text{cl}})r_R - (2\rho_{\text{cl}} - 2)r_L = r_L + 1 - r_R,$$

which is $r_R^* := \frac{2\rho_{\text{cl}}-1}{5-2\rho_{\text{cl}}} r_L + \frac{1}{5-2\rho_{\text{cl}}}$. Because of the inequality

$$\begin{aligned} \left(\frac{2\rho_{\text{cl}} - 3}{4 - 2\rho_{\text{cl}}} - \frac{2\rho_{\text{cl}} - 1}{5 - 2\rho_{\text{cl}}}\right) r_L &= \frac{6\rho_{\text{cl}} - 11}{4\rho_{\text{cl}}^2 - 18\rho_{\text{cl}} + 20} r_L \\ &\leq \frac{4\rho_{\text{cl}} - 7}{2\rho_{\text{cl}}^2 - 10\rho_{\text{cl}} + 12}(2\rho_{\text{cl}} - 1) \\ &\leq \frac{6\rho_{\text{cl}}^2 - 17\rho_{\text{cl}} + 11}{4\rho_{\text{cl}}^2 - 18\rho_{\text{cl}} + 20} \end{aligned}$$

$$\overset{\rho_{\mathrm{cl}} < 1.85}{<} \frac{1}{5 - 2\rho_{\mathrm{cl}}},$$

we have $r_R^* > \frac{2\rho_{\mathrm{cl}}-3}{4-2\rho_{\mathrm{cl}}} r_L$, i.e., inequality (3.2) holds. □

Finally, we show that the requests s_R^j and s_L^j with $j \in \{1, \ldots, c\}$ are critical.

Lemma 3.7. *Let $j \in \{1, \ldots, c\}$. The requests s_R^j and s_L^j of the request sequence $\sigma'_{\rho_{\mathrm{cl}}}$ satisfy Definition 3.4.*

Proof. The release time of every request is equal to its starting position, thus every request can be served/loaded immediately once its starting position is visited and (i) of Definition 3.4 is satisfied. Lemma 3.6 shows that (ii) and (iii) of Definition 3.4 are satisfied. It remains to show that property (iv) of Definition 3.4 is satisfied. For this, we need to examine the release time r_R of the requests s_R^j. The time r_R is largest if Alg tries to avoid crossing the line ℓ_{cl} as long as possible, i.e., it continues to move right after serving the request s_R^0. Then, we have $\mathrm{pos}(t) = 1 - r_L + t$ for $t \geq r_L$ and r_R is the solution of

$$1 - r_L + r_R = (5 - 2\rho_{\mathrm{cl}}) r_R - (2\rho_{\mathrm{cl}} - 2) r_L.$$

Thus, in general, we have $r_R \leq \frac{2\rho_{\mathrm{cl}}-3}{4-2\rho_{\mathrm{cl}}} r_L + \frac{1}{4-2\rho_{\mathrm{cl}}}$, i.e.,

$$\frac{r_R}{r_L} \leq \frac{2\rho_{\mathrm{cl}} - 3}{4 - 2\rho_{\mathrm{cl}}} + \frac{1}{(4 - 2\rho_{\mathrm{cl}}) r_L} \overset{r_L \geq 2}{\leq} \frac{4\rho_{\mathrm{cl}} - 5}{8 - 4\rho_{\mathrm{cl}}}.$$

For property (iv), we need $\frac{r_R}{r_L} \leq \frac{2\rho_{\mathrm{cl}}^2 - 6\rho_{\mathrm{cl}} + 4}{-2\rho_{\mathrm{cl}}^2 + 4\rho_{\mathrm{cl}} - 1}$. This is satisfied if

$$\frac{4\rho_{\mathrm{cl}} - 5}{8 - 4\rho_{\mathrm{cl}}} \leq \frac{2\rho_{\mathrm{cl}}^2 - 6\rho_{\mathrm{cl}} + 4}{-2\rho_{\mathrm{cl}}^2 + 4\rho_{\mathrm{cl}} - 1}$$

holds, which is equivalent to

$$-14\rho_{\mathrm{cl}}^2 + 40\rho_{\mathrm{cl}} - 27 \geq 0.$$

This is true by definition of ρ_{cl}. □

Together with Lemma 3.5, this completes the proof of Theorem 3.1.

3.2 Lower Bound for Open Online DIAL-A-RIDE on the Line

In this section, we prove Theorem 3.2. Let $c < \infty$ and ALG be an eager online algorithm for open online DIAL-A-RIDE. Let $\rho_{\text{op}} \approx 2.0585$, be the second largest root of the polynomial $4\rho^3 - 26\rho^2 + 39\rho - 5$. We describe a request sequence $\sigma_{\rho_{\text{op}}}$ such that

$$\text{ALG}(\sigma_{\rho_{\text{op}}}) \geq \rho_{\text{op}}\text{OPT}(\sigma_{\rho_{\text{op}}}).$$

The proof follows the same design principle as the proof of Theorem 3.1: We construct a request sequence in a first stage that forces ALG into some critical situation that is then exploited in the second stage.

The request sequence that will be constructed in this section is based on the request sequence presented by Bjelde et al. in [13] to prove the lower bound of 2.0346 for online TSP on the line: The first stage starts with an initial request $(1, 1; 1)$ (assuming w.l.o.g. ALG's position at time 1 is at most 0). This stage consists of a loop, which ends as soon as two so-called critical requests are established. The second stage exploits the situation generated by the critical requests by releasing suitable additional requests to show the desired competitive ratio. A single iteration of the loop only yields a lower bound of roughly 2.0298, but as the number of iterations approaches infinity one can show the tight bound of roughly 2.0346 in the limit.

In Theorem 3.2, we show a lower bound of roughly 2.0585 using the same general structure but only a single iteration in the first stage. Our additional leeway stems from replacing the initial request $(1, 1; 1)$ with c initial requests of the form $(1, \delta; 1)$ where $\delta > 1$: At the time when an initial request is loaded, we show that w.l.o.g. all c requests are loaded and then proceed as we did when $(1, 1; 1)$ was served.

Before we explain the first stage in detail, we consider the second stage. We start by describing the critical situation we want to force ALG into and then explain how to exploit it. Suppose we have two requests $s_R = (r_R, r_R; r_R)$ and $s_L = (-r_L, -r_L, r_L)$ with $r_L \leq r_R$ to the right and to the left of the origin, respectively. We assume that ALG serves s_R first at some time $t^* \geq (2\rho_{\text{op}} - 2)r_L + (\rho_{\text{op}} - 2)r_R$. Now suppose we could force ALG to serve s_L directly after s_R, even if additional requests are released. Then we could just release the request $s_R^* = (r_R, r_R, 2r_L + r_R)$ and we would have

$$\text{ALG}(\sigma_{\rho_{\text{op}}}) = t^* + 2r_L + 2r_R \geq 2\rho_{\text{op}}r_L + \rho_{\text{op}}r_R = \rho_{\text{op}}\text{OPT}(\sigma_{\rho_{\text{op}}}),$$

since OPT can serve the three requests in time $2r_L + r_R$ by serving s_L first. In fact, we will show that we can force ALG into this situation (or a worse situation) if the requests $s_R = (r_R, r_R; r_R)$ and $s_L = (-r_L, -r_L, r_L)$ satisfy the following properties.

Definition 3.8. Let $\rho > 0$. We call the last two requests $s_R = (r_R, r_R; r_R)$ and $s_L = (-r_L, -r_L, r_L)$ of a request sequence with $0 < r_L \leq r_R$ *o-ρ-critical* for ALG if the following conditions hold:

(i) Both tours $\text{move}(-r_L) \oplus \text{move}(r_R)$ and $\text{move}(r_R) \oplus \text{move}(-r_L)$ serve all requests presented until time r_R.

(ii) ALG serves both s_R and s_L after time r_R and ALG's position at time r_R lies between r_R and $-r_L$.

(iii) If ALG serves s_R before s_L, it does so no earlier than $t_R^* := (2\rho - 2)r_L + (\rho - 2)r_R$.

(iv) If ALG serves s_L before s_R, it does so no earlier than $t_L^* := (2\rho - 2)r_R + (\rho - 2)r_L$.

(v) It holds that $\frac{r_R}{r_L} \leq \frac{4\rho^2 - 30\rho + 50}{-8\rho^2 + 50\rho - 66}$.

Definition 3.8 differs from [13, Definition 5] for online TSP on the line only in property (v), which is $\frac{r_R}{r_L} \leq 2$ in the original paper.

Analysis of the Second Stage

Definition 3.8 describes the critical situation ALG is forced into at the end of the first stage. As before in the closed case, there are two things we have to prove: We can present a request sequence that satisfies Definition 3.8 (first stage), and, once ALG is in the critical situation described by Definition 3.8, we can release additional requests, such that ALG is not better than ρ-competitive. In this subsection, we show the latter.

Lemma 3.9. *Let $\rho \in (2, \frac{1}{3}(10 - \sqrt{13})) \approx (2, 2.1315)$. If there is a request sequence with two o-ρ-critical requests for* ALG*, we can release additional requests such that* ALG *is not $(\rho - \varepsilon)$-competitive on the resulting instance for any $\varepsilon > 0$.*

Lemma 3.9 has been proved in [13, Lemma 6] for request sequences that satisfy the properties of [13, Definition 5], however, a careful inspection of the proof of [13, Lemma 6] shows that the statement of Lemma 3.9 also holds for request sequences that only satisfy (v) instead of $\frac{r_R}{r_L} \leq 2$. For the sake of completeness, we present the full proof for our slightly different version of Definition 3.8 and Lemma 3.9.

Let σ_ρ be a request sequence with o-ρ-critical requests s_L and s_R according to Definition 3.8. First, we note that Definition 3.8 implies $0 < r_L \leq r_R$, i.e., Definition 3.8 (v) can only hold if

$$1 \leq \frac{r_R}{r_L} \leq \frac{4\rho^2 - 30\rho + 50}{-8\rho^2 + 50\rho - 66},$$

i.e., if $\rho \in (\frac{1}{8}(25-\sqrt{97}), \frac{1}{3}(10-\sqrt{13})] \approx (1.8939, 2.1315]$ holds. Let $a_0 \in \{-r_L, r_R\}$ be the position of the request $s_0 \in \{s_L, s_R\}$ that is served first by ALG and let $a_1 \in \{-r_L, r_R\}$ be the position of the request $s_1 \in \{s_L, s_R\}$ that is not served first. By properties (iii) and (iv) of Definition 3.8, ALG cannot serve s_0 before time $t_0^* := (2\rho-2)|a_1| + (\rho-2)|a_0|$. Thus, we have

$$\textsc{Alg}(\sigma_\rho) \geq t_0^* + |a_0 - a_1| = (2\rho-1)|a_1| + (\rho-1)|a_0| =: t_1^*, \tag{3.3}$$

i.e., ALG cannot serve s_1 before time t_1^*. We have equality in inequality (3.3) if ALG serves s_0 the earliest possible time t_0^* and then moves directly to position a_1 serving s_1 at time t_1^*. However, in general ALG does not need to do this and instead might wait. If ALG still has to serve s_0 at time $t \geq \max\{|a_0|, |a_1|\}$, we have

$$\textsc{Alg}(\sigma_\rho) \geq t + |\text{pos}(t) - a_0| + |a_0 - a_1|$$

and if s_0 is served and only s_1 is left to be served, we have

$$\textsc{Alg}(\sigma_\rho) \geq t + |\text{pos}(t) - a_1|.$$

We want to measure the delay of ALG at a time $t \geq \max\{|a_0|, |a_1|\}$, i.e., the difference between the minimum time ALG needs to serve both requests s_0 and s_1 and the time t_1^*, which is the earliest possible time, both requests are served. For $t \geq \max\{|a_0|, |a_1|\}$ we define the function

$$\text{delay}(t) := \begin{cases} t + |\text{pos}(t) - a_0| + |a_0 - a_1| - t_1^* & \text{if } s_0 \text{ is not served at } t, \\ t + |\text{pos}(t) - a_1| - t_1^* & \text{if } s_0 \text{ is served at } t \text{, but } s_1 \text{ not}, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

We make the following observations about delay:

Observation 3.10. *Let $t \geq \max\{|a_0|, |a_1|\}$ be a time at which s_1 is not served yet, i.e. $\text{delay}(t)$ is defined. Then we have $\text{delay}(t) \geq 0$.*

Proof. First, we note that request s_1 is not served before time t_1^*. At every point of time $t \geq \max\{|a_0|, |a_1|\}$ at which s_0 is not served yet, ALG needs at least time $|\text{pos}(t) - a_0| + |a_0 - a_1|$ to serve s_0 and then s_1. This implies

$$t + |\text{pos}(t) - a_0| + |a_0 - a_1| \geq t_1^*,$$

i.e., $\text{delay}(t) \geq 0$. Similarly, at every point of time $t \geq \max\{|a_0|, |a_1|\}$ at which s_0 is served, but s_1 not, ALG needs at least time $|\text{pos}(t) - a_1|$ to serve s_1. This implies

$$t + |\text{pos}(t) - a_0| \geq t_1^*,$$

i.e., again $\text{delay}(t) \geq 0$. $\square$

Observation 3.11. *Let $t \geq \max\{|a_0|, |a_1|\}$ be a time at which s_1 is not served yet. The earliest time* ALG *can serve s_1 is $t_1^* + \operatorname{delay}(t)$.*

Proof. At every point of time $t \geq \max\{|a_0|, |a_1|\}$ at which s_0 is not served yet, ALG needs at least time

$$|\operatorname{pos}(t) - a_0| + |a_0 - a_1| = t_1^* + \operatorname{delay}(t)$$

to serve s_0 and then s_1. Similarly, at every point of time $t \geq \max\{|a_0|, |a_1|\}$ at which s_0 is served, but s_1 not, ALG needs at least time

$$|\operatorname{pos}(t) - a_1| = t_1^* + \operatorname{delay}(t)$$

to serve s_1. □

Lemma 3.12. *Let $\rho > 2$ and* ALG *be $(\rho - \varepsilon)$-competitive for some $\varepsilon > 0$. There is a $W \geq 0$ with*

$$\operatorname{delay}\left(2|a_1| + |a_0| + \frac{W}{\rho - 1}\right) = W.$$

Proof. Because of property (ii) of Definition 3.8, at time $\max\{|a_0|, |a_1|\}$ neither s_0 nor s_1 has been served by ALG yet. Since ALG serves s_1 after s_0, the request s_1 is not served before time

$$\max\{|a_0|, |a_1|\} + |a_0| + |a_1| \geq 2|a_1| + |a_0|,$$

i.e, $\operatorname{delay}(2|a_1| + |a_0|)$ is defined and because of Observation 3.10, we have $\operatorname{delay}(2|a_1| + |a_0|) \geq 0$. If $\operatorname{delay}(2|a_1| + |a_0|) = 0$, we have $W = 0$ and are done. Otherwise, by Observation 3.10, we have

$$\operatorname{delay}(2|a_1| + |a_0|) > 0. \tag{3.4}$$

ALG needs to serve s_1 at some point to be $(\rho - \varepsilon)$-competitive. Let W^* be chosen such that ALG serves s_1 at time $2|a_1| + |a_0| + \frac{W^*}{\rho - 1}$. Therefore,

$$\operatorname{delay}\left(2|a_1| + |a_0| + \frac{W^*}{\rho - 1} - \varepsilon'\right)$$

is defined for some sufficiently small $\varepsilon' \leq |a_1|$. Define the function

$$f(W) := \operatorname{delay}\left(2|a_1| + |a_0| + \frac{W}{\rho - 1}\right) - W.$$

Note that f is continuous, and we have $f(0) > 0$ by inequality (3.4). If

$$\operatorname{delay}\left(2|a_1| + |a_0| + \frac{W^*}{\rho - 1} - \varepsilon'\right) \leq \frac{W^*}{\rho - 1} - \varepsilon' \overset{\rho > 2}{<} W^* - (\rho - 1)\varepsilon',$$

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

$$\operatorname{delay}\left(2|a_1| + |a_0| + \frac{W^*}{\rho - 1} - \varepsilon'\right) > \frac{W^*}{\rho - 1} - \varepsilon'. \tag{3.5}$$

Then, by Observation 3.11, the earliest possible time Alg serves s_1 is

$$t_1^* + \operatorname{delay}\left(2|a_1| + |a_0| + \frac{W^*}{\rho - 1} - \varepsilon'\right) \overset{(3.5)}{>} t_1^* + \frac{W^*}{\rho - 1} - \varepsilon',$$

i.e., s_1 is not served at time $t_1^* + \frac{W^*}{\rho-1} - \varepsilon'$. However, we have

$$\begin{aligned} t_1^* + \frac{W^*}{\rho - 1} - \varepsilon' \quad &= \quad (2\rho - 1)|a_1| + (\rho - 1)|a_0| + \frac{W^*}{\rho - 1} - \varepsilon' \\ &\overset{\rho > 2, \varepsilon' \leq |a_1|}{>} \quad 2|a_1| + |a_0| + \frac{W^*}{\rho - 1}, \end{aligned}$$

which is a contradiction to the fact, that W^* was chosen such that Alg serves s_1 at time $2|a_1| + |a_0| + \frac{W^*}{\rho-1}$. □

Lemma 3.13. *Let $2 < \rho < \frac{5}{2}$ and Alg be $(\rho - \varepsilon)$-competitive for some $\varepsilon > 0$. Furthermore, let $W \geq 0$ with*

$$\operatorname{delay}\left(2|a_1| + |a_0| + \frac{W}{\rho - 1}\right) = W.$$

Then Alg serves s_0 no later than time $2|a_1| + |a_0| + \frac{W}{\rho-1}$.

Proof. Assume we have

$$2|a_1| + |a_0| + \frac{W}{\rho - 1} \geq t_0^* + W. \tag{3.6}$$

By definition of W and Observation 3.11, Alg can serve s_1 at time

$$t_1^* + \operatorname{delay}\left(2|a_1| + |a_0| + \frac{W}{\rho - 1}\right) = t_1^* + W. \tag{3.7}$$

Because of inequality (3.6), this can only be the case if Alg serves s_0 no later than time

$$t_1^* + W - |a_1 - a_0| = t_0^* + W \overset{(3.6)}{\leq} 2|a_1| + |a_0| + \frac{W}{\rho - 1}.$$

Thus, it remains to show inequality (3.6). Because of property (i) of Definition 3.8 all requests can be served by the tour move$(a_1)\oplus$move(a_0), i.e., we have $\textsc{Opt}(\sigma_\rho) \leq 2|a_1|+|a_0|$. By inequality (3.7), we have $\textsc{Alg}(\sigma_\rho) \geq t_1^* + W$. Thus, if we have

$$\textsc{Alg}(\sigma_\rho) \geq t_1^* + W > (\rho - \varepsilon)(2|a_1| + |a_0|) \geq (\rho - \varepsilon)\textsc{Opt}(\sigma_\rho),$$

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

$$t_1^* + W \leq (\rho - \varepsilon)(2|a_1| + |a_0|),$$

and thus

$$\begin{aligned} W &\leq (\rho - \varepsilon)(2|a_1| + |a_0|) - t_1^* \\ &= (\rho - \varepsilon)(2|a_1| + |a_0|) - (2\rho - 1)|a_1| - (\rho - 1)|a_0| \\ &= (1 - 2\varepsilon)|a_1| + (1 - \varepsilon)|a_0| \\ &< |a_1| + |a_0|. \end{aligned} \tag{3.8}$$

If we solve inequality (3.6) for W, it is equivalent to

$$\begin{aligned} \frac{2|a_1| + |a_0| - t_0^*}{1 - \frac{1}{\rho-1}} &= \frac{2|a_1| + |a_0| - ((2\rho - 2)|a_1| + (\rho - 2)|a_0|)}{1 - \frac{1}{\rho-1}} \\ &= \frac{(\rho - 1)((4 - 2\rho)|a_1| + (3 - \rho)|a_0|)}{\rho - 2} \\ &= \frac{(\rho - 1)(4 - 2\rho)}{\rho - 2}|a_1| + \frac{(\rho - 1)(3 - \rho)}{\rho - 2}|a_0| \\ &\overset{\text{Def 3.8 (v)}}{\geq} |a_0| + (2 - 2\rho)|a_1| + \frac{(-\rho^2 + 3\rho - 1)(-8\rho^2 + 50\rho - 66)}{(\rho - 2)(4\rho^2 - 30\rho + 50)}|a_1| \\ &\geq |a_0| + \frac{5\rho^3 - 36\rho^2 + 86\rho - 67}{2\rho^3 - 19\rho^2 + 55\rho - 50}|a_1| \\ &\overset{2<\rho<2.5}{>} |a_0| + |a_1| \\ &\overset{(3.8)}{>} W. \end{aligned}$$

□

Now we have all ingredients to prove Lemma 3.9.

Proof of Lemma 3.9. Let $W \geq 0$ with

$$\text{delay}\left(2|a_1| + |a_0| + \frac{W}{\rho - 1}\right) = W.$$

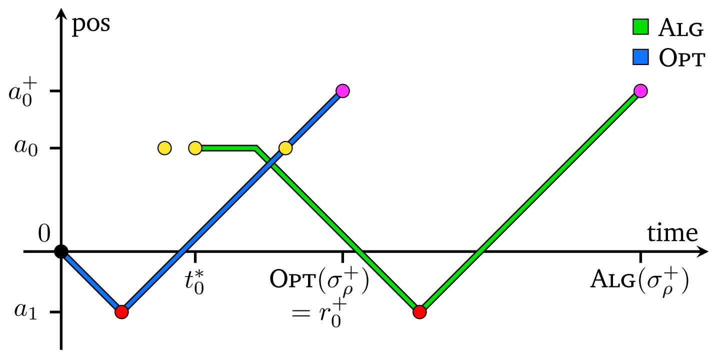

Figure 3.4: Case 1: ALG (green) serves s_1 (red ●) before s_0^+ (violet ●). In this figure we have $\text{delay}(t) = W = 2$ for $t \geq t_0^* + W$. OPT is blue, the request s_0 is yellow ●.

We present the request

$$\begin{aligned} s_0^+ &= (a_0^+, a_0^+; r_0^+) \\ &:= \left(a_0 + \text{sgn}(a_0)\frac{W}{\rho - 1}, a_0 + \text{sgn}(a_0)\frac{W}{\rho - 1}; 2|a_1| + |a_0| + \frac{W}{\rho - 1}\right) \end{aligned}$$

and define σ_ρ^+ to be the request sequence σ_ρ plus the request s_0^+. We distinguish two cases.

Case 1: At time r_0^+, ALG is at least as close to a_1 as to a_0^+ or it serves s_1 before s_0^+.
See Figure 3.4 for an illustration of this case. In this case, we do not present additional requests. By Lemma 3.13, ALG has served s_0 at time r_0^+ or before and by Observation 3.11 it does not serve s_1 earlier than time $t_1^* + W$. Thus, we have

$$\begin{aligned} \text{ALG}(\sigma_\rho^+) &\geq t_1^* + W + |a_1| + |a_0| + \frac{W}{\rho - 1} \\ &= \rho\left(2|a_1| + |a_0| + \frac{W}{\rho - 1}\right) \\ &= \rho\text{OPT}(\sigma_\rho^+). \end{aligned}$$

Case 2: At time r_0^+, ALG is closer to a_0^+ than to a_1 and it serves s_0^+ first.
We assume that the offline server first serves s_1, then s_0 and then s_0^+. If the offline server continues to move away from the origin after serving s_0^+ at time a_0^+, its position at time

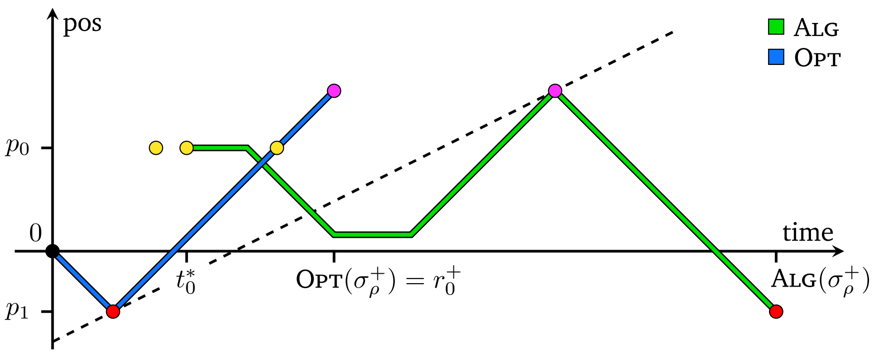

Figure 3.5: Case 2.1: The midpoint of OPT's position and a_1 (dashed line) reaches a_0^+ (violet) the same time as ALG (green). No new requests are released. In this figure we have $\text{delay}(t) = W = 2$ for $t \geq t_0^* + W$. OPT is blue, the request s_0 is yellow and the request s_1 is red.

$t \geq |a_1|$ is $\text{sgn}(a_0)t + 2a_1$. We denote by

$$M(t) := \frac{\text{sgn}(a_0)t + 3a_1}{2}$$

the midpoint between the current position of the offline server and the position a_1. Note that the time $M^{-1}(p)$, when the midpoint is at position p is given by

$$M^{-1}(p) := |2p - 3a_1|.$$

We again distinguish between two cases depending on the time, ALG serves the request s_0^+. We first take a look at the case that ALG serves s_0^+ too late.

Case 2.1: ALG does not serve s_0^+ until time $M^{-1}(a_0^+)$.

See Figure 3.5 for an illustration of this case. In this case, the midpoint of the offline server's position and position a_1 reaches a_0^+ before ALG. We do not present additional requests. Since we are in Case 2, neither s_0^+ nor s_1 is served at time $M^{-1}(a_0^+)$. Thus, we have

$$\begin{aligned}
\text{ALG}(\sigma_\rho^+) &\geq M^{-1}(a_0^+) + |a_0^+| + |a_1| \\
&= |2a_0^+ - 3a_1| + |a_0^+| + |a_1| \\
&= |2a_0 + 2\text{sgn}(a_0)\frac{W}{\rho - 1} - 3a_1| + |a_0| + \frac{W}{\rho - 1} + |a_1| \\
&= 3|a_0| + 4|a_1| + 3\frac{W}{\rho - 1}
\end{aligned}$$

$$\begin{aligned}
&= \rho|a_0| + (3-\rho)|a_0| + 4|a_1| + 3\frac{W}{\rho-1} \\
&\overset{\text{Def 3.8 (v)}}{\geq} \rho|a_0| + \frac{(3-\rho)(-8\rho^2+50\rho-66)}{4\rho^2-30\rho+50}|a_1| + 4|a_1| + 3\frac{W}{\rho-1} \\
&= \rho|a_0| + \frac{4\rho^3-29\rho^2+48\rho+1}{2\rho^2-15\rho+25}|a_1| + 3\frac{W}{\rho-1} \\
&\overset{2<\rho<2.5}{>} \rho|a_0| + 2\rho|a_1| + 3\frac{W}{\rho-1} \\
&\overset{\rho<3}{>} \rho\left(|a_0| + 2|a_1| + \frac{W}{\rho-1}\right) \\
&= \rho\textsc{Opt}(\sigma_\rho^+).
\end{aligned}$$

Case 2.2: ALG serves s_0^+ before time $M^{-1}(a_0^+)$.
See 3.6 for an illustration of this case. By definition of W, the function delay is defined for time a_0^+, hence ALG has not served s_1 before time a_0^+. Since ALG is to the right of the midpoint $M(a_0^+)$ at time a_0^+, there is a first time t_{mid} at which $M(t_{\text{mid}}) = \text{pos}(t_{\text{mid}})$. We present the request

$$s_0^{++} = (a_0^{++}, a_0^{++}; r_0^{++}) := (\text{sgn}(a_0)t_{\text{mid}} + 2a_1, \text{sgn}(a_0)t_{\text{mid}} + 2a_1; t_{\text{mid}}).$$

and define σ_ρ^{++} to be the request sequence σ_ρ^+ plus the request s_0^{++}. Note that ALG is at the midpoint between a_0^{++} and a_1 and thus, both tours $\text{move}(a_0^{++}) \oplus \text{move}(a_1)$ and $\text{move}(a_1) \oplus \text{move}(a_0^{++})$ incur identical costs for ALG. We have

$$\textsc{Alg}(\sigma_\rho) \geq t_{\text{mid}} + 3\left(\frac{|\text{sgn}(a_0)t_{\text{mid}} + 2a_1 - a_1|}{2}\right) = \frac{5t_{\text{mid}} - 3|a_1|}{2}.$$

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

$$\textsc{Alg}(\sigma_\rho) \geq \frac{5t_{\text{mid}} - 3|a_1|}{2} \geq \rho t_{\text{mid}} = \rho\textsc{Opt}(\sigma_\rho). \tag{3.9}$$

Inequality (3.9) is equivalent to

$$(5-2\rho)t_{\text{mid}} \geq 3|a_1|. \tag{3.10}$$

Since $\rho < 2.5$, the coefficient $(5-2\rho)$ of t_{mid} is positive. Thus, we may assume t_{mid} is minimal to show the inequality (3.10). By assumption, s_0^+ is already served at time t_{mid}.

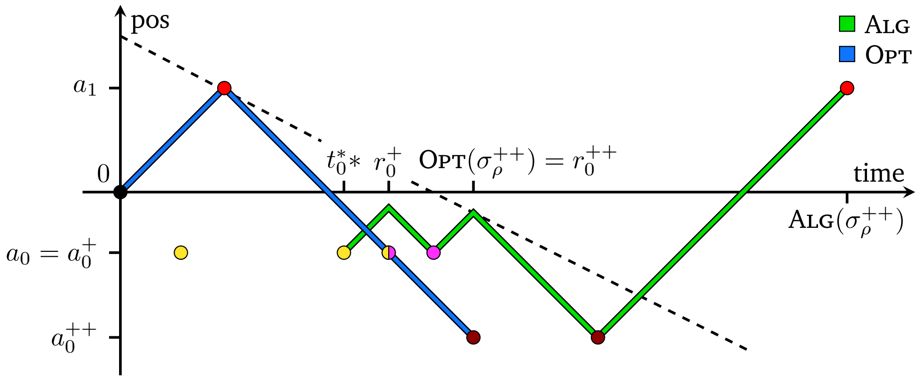

Figure 3.6: Case 2.2: ALG (green) serves s_0^+ (violet) before the midpoint of OPT's position and a_1 (dashed line) reaches a_0^+. Thus, s_0^+ (brown) is released. In this figure we have $\text{delay}(t) = W = 0$ for $t \geq t^* + W$. OPT is shown in blue, the request s_0 is shown in yellow, the request s_1 is shown in red.

Hence, t_{mid} is minimum if, starting at time r_0^+ at position $\text{pos}(r_0^+)$, ALG serves s_0^+ and then moves towards the origin. Then, t_{mid} is the solution of the equation

$$\text{sgn}(a_0)r_0^+ + |\text{pos}(r_0^+) - a_0^+| + a_0^+ - \text{sgn}(a_0)t_{\text{mid}} = \frac{\text{sgn}(a_0)t_{\text{mid}} + 3a_1}{2}. \tag{3.11}$$

Because of Lemma 3.13, the request s_0 is already served at time r_0^+. Furthermore, since the position of s_1 has not been visited yet at time r_0^+, we have $\text{sgn}(a_0)\text{pos}(r_0^+) > \text{sgn}(a_0)a_1$, i.e.,

$$|\text{pos}(r_0^+) - a_1| = \text{sgn}(a_0)(\text{pos}(r_0^+) - a_1) > 0$$

and thus, because of $-\text{sgn}(a_0)a_1 = |a_1|$, we get

$$\begin{aligned}\text{delay}(r_0^+) &= r_0^+ + |\text{pos}(r_0^+) - a_1| - t_1^* \\ &= r_0^+ + \text{sgn}(a_0)\text{pos}(r_0^+) - \text{sgn}(a_0)a_1 - t_1^* \\ &= r_0^+ + \text{sgn}(a_0)\text{pos}(r_0^+) + |a_1| - t_1^*. \end{aligned} \tag{3.12}$$

Solving equation (3.12) for $\text{sgn}(a_0)\text{pos}(r_0^+)$ gives

$$\begin{aligned}\text{sgn}(a_0)\text{pos}(r_0^+) \quad &= \quad \text{delay}\left(2|a_1| + |a_0| + \frac{W}{\rho - 1}\right) - \frac{W}{\rho - 1} \\ &\quad\quad + (\rho - 2)|a_0| + (2\rho - 4)|a_1| \\ &= \quad W - \frac{W}{\rho - 1} + (\rho - 2)|a_0| + (2\rho - 4)|a_1|\end{aligned}$$

$$
\begin{aligned}
&= \quad \frac{\rho-2}{\rho-1}W + (\rho-2)|a_0| + (2\rho-4)|a_1| \qquad (3.13)\\
&\overset{\rho<3}{<} \quad \frac{W}{\rho-1} + (\rho-2)|a_0| + (2\rho-4)|a_1|\\
&\overset{\text{Def 3.8 (v)}}{\leq} \quad \frac{W}{\rho-1} + \left((\rho-2) + (2\rho-4)\frac{4\rho^2-30\rho+50}{-8\rho^2+50\rho-66}\right)|a_0|\\
&\overset{1.9<\rho<4.3}{<} \quad \frac{W}{\rho-1} + |a_0|\\
&= \quad |a_0^+|\\
&\overset{\operatorname{sgn}(a_0^+)=\operatorname{sgn}(a_0)}{=} \quad \operatorname{sgn}(a_0)a_0^+.
\end{aligned}
$$

Thus, we have

$$
|\text{pos}(r_0^+) - a_0^+| = \operatorname{sgn}(a_0)(a_0^+ - \text{pos}(r_0^+)) > 0. \tag{3.14}
$$

Using inequality (3.14) and plugging inequality (3.13) into inequality (3.11) gives us

$$
\begin{aligned}
\operatorname{sgn}(a_0)t_{\text{mid}} &= \frac{1}{3}(2\operatorname{sgn}(a_0)r_0^+ + 2|\text{pos}(r_0^+) - 2a_0^+| + 2a_0^+ - 3a_1)\\
&\overset{(3.14)}{=} \frac{1}{3}(2\operatorname{sgn}(a_0)r_0^+ + 2\operatorname{sgn}(a_0)a_0^+ - 2\operatorname{sgn}(a_0)\text{pos}(r_0^+) + 2a_0^+ - 3a_1)\\
&= \frac{1}{3}\left(-7a_1 + 6a_0 + \frac{(6\operatorname{sgn}(a_0))W}{\rho-1} - 2\operatorname{sgn}(a_0)\text{pos}(r_0^+)\right)\\
&\overset{(3.13)}{=} \frac{1}{3}\left(-(15-4\rho)a_1 + (10-2\rho)a_0 + \frac{(10-2\rho)\operatorname{sgn}(a_0)W}{\rho-1}\right). \qquad (3.15)
\end{aligned}
$$

Note that we also used $\operatorname{sgn}(a_0) = \operatorname{sgn}(a_0^+) = -\operatorname{sgn}(a_1)$. Multiplying equality (3.15) with $\operatorname{sgn}(a_0)$ gives us

$$
t_{\text{mid}} = \frac{1}{3}\left((15-4\rho)|a_1| + (10-2\rho)|a_0| + \frac{(10-2\rho)W}{\rho-1}\right). \tag{3.16}
$$

By substituting equation (3.16) into inequality (3.10) and noting that it is hardest to satisfy, when $W = 0$, we get

$$
\frac{|a_1|}{|a_0|} \leq \frac{4\rho^2-30\rho+50}{-8\rho^2+50\rho-66},
$$

which is true due to Definition 3.8 (v). □

Analysis of the First Stage

In the previous subsection we have proved that there is no $(\rho - \varepsilon)$-competitive algorithm, if we are able to construct a request sequence with two o-ρ-critical requests. Thus, our goal is to construct a request sequence $\sigma'_{\rho_{\text{op}}}$ that satisfies all properties of Definition 3.8.

We let ALG wait until time 1. Without loss of generality, we assume that ALG position at time 1 is $\text{pos}(1) \leq 0$ (the other case is symmetric). We define

$$\delta := \frac{3\rho_{\text{op}}^2 - 11}{-3\rho_{\text{op}}^3 + 15\rho_{\text{op}} - 4}$$

and let c initial requests $s_0^j = (1, \delta; 1)$ with $j \in \{1, \dots, c\}$ appear. These are the only requests appearing in the entire construction with a starting position differing from the destination. We make a basic observation on how ALG has to serve these requests.

Lemma 3.14. ALG *cannot collect any of the requests* s_0^j *before time* 2*. If* ALG *collects the requests after time* $\rho_{\text{op}}\delta - (\delta - 1)$ *or serves* $c' < c$ *requests before loading the remaining* $c - c'$*, it is not* $(\rho_{\text{op}} - \varepsilon)$*-competitive.*

Proof. ALG cannot collect any s_0^j before time 2 since its position at time 1 is $\text{pos}(1) \leq 0$. Moreover, ALG is not $(\rho_{\text{op}} - \varepsilon)$-competitive if it collects one of the requests after time $\rho_{\text{op}}\delta - (\delta - 1)$, since it cannot finish before time $\rho_{\text{op}}\delta$, and we have

$$\text{ALG}((s_0^j)_{j\in\{1,\dots,c\}}) \geq \rho_{\text{op}}\delta = \rho_{\text{op}}\text{OPT}((s_0^j)_{j\in\{1,\dots,c\}}).$$

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

$$\delta = \frac{3\rho_{\text{op}}^2 - 11}{-3\rho_{\text{op}}^3 + 15\rho_{\text{op}} - 4} \overset{\rho_{\text{op}}>2.056}{>} \frac{2}{3 - \rho_{\text{op}}}, \tag{3.17}$$

we have

$$\text{ALG}((s_0^j)_{j\in\{1,\dots,c\}}) \geq \delta + 2(\delta - 1) \overset{(3.17)}{>} \rho_{\text{op}}\delta = \rho_{\text{op}}\text{OPT}((s_0^j)_{j\in\{1,\dots,c\}}). \qquad \square$$

We hence may assume that ALG loads all c requests s_0^j at the same time. Let $r_L \in [2, \rho_{\text{op}}\delta - (\delta - 1))$ be the time ALG loads the c requests s_0^j. We present a variant of a single iteration of the construction in [13]: We let the request $s_L = (-r_L, -r_L; r_L)$ appear and define the function

$$\ell_{\text{op}}(t) = (4 - \rho_{\text{op}}) \cdot t - (2\rho_{\text{op}} - 2) \cdot r_L,$$

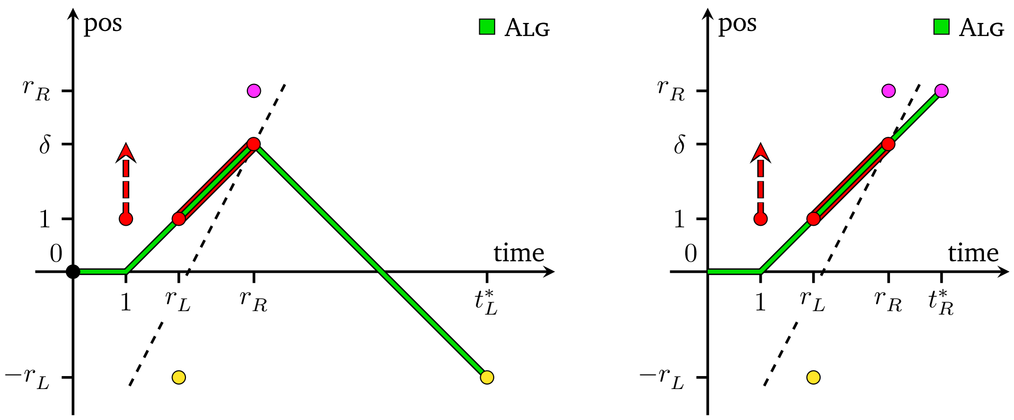

Figure 3.7: Left: ALG serves s_L (yellow ●) before the request s_R (violet ●) at time t_L^*. Right: ALG serves s_R before the request s_L at time t_R^*. The requests s_0^j are red ● and the line ℓ_{op} is the dashed black line.

which can be viewed as a line in the position-time diagram. Because of $\rho_{\text{op}} > 2$, we have $\ell_{\text{op}}(r_L) = (6 - 3\rho_{\text{op}})r_L < 0 < \text{pos}(r_L)$, i.e., ALG's position at time r_L is to the right of the line ℓ_{op}. Thus, ALG crosses the line ℓ_{op} before it serves s_L. Let r_R be the time ALG crosses ℓ_{op} for the first time and let the request $s_R = (r_R, r_R; r_R)$ appear. We define $\sigma'_{\rho_{\text{op}}} := (s_0^1, \dots, s_0^c, s_L, s_R)$

Lemma 3.15. ALG *can neither serve* s_L *before time* t_L^* *nor can it serve* s_R *before time* t_R^*.

Proof. For an illustration of this lemma's construction see Figure 3.7. Assume ALG crosses the line ℓ_{op} and serves s_R before s_L. Then it does not serve s_R before time

$$r_R + |\ell_{\text{op}}(r_R) - r_R| = (2\rho_{\text{op}} - 2)r_L + (\rho_{\text{op}} - 2)r_R = t_R^*.$$

Now assume ALG crosses ℓ_{op} at time

$$r_R \geq \frac{3\rho_{\text{op}} - 5}{7 - 3\rho_{\text{op}}} r_L \tag{3.18}$$

and serves s_L before s_R. Then it does not serve serve s_L before time

$$\begin{aligned} r_R + |\ell_{\text{op}}(r_R) - (-r_L)| &= (5 - \rho_{\text{op}})r_R - (2\rho_{\text{op}} - 3)r_L \\ &\overset{(3.18)}{\geq} (2\rho_{\text{op}} - 2)r_R + (7 - 3\rho_{\text{op}})\frac{3\rho_{\text{op}} - 5}{7 - 3\rho_{\text{op}}} r_L - (2\rho_{\text{op}} - 3)r_L \\ &= (2\rho_{\text{op}} - 2)r_R + (\rho_{\text{op}} - 2)r_L = t_L^*. \end{aligned}$$

Thus, it is enough to show inequality (3.18). Since ALG is eager, it delivers the c requests s_0^j without waiting or detour, i.e., we have $\text{pos}(r_L + (\delta - 1)) = \delta$. Furthermore, we have

$$\begin{aligned}
\ell_{\text{op}}(r_L + (\delta - 1)) &= (4 - \rho_{\text{op}})(r_L + (\delta - 1)) - (2\rho_{\text{op}} - 2)r_L \\
&= (6 - 3\rho_{\text{op}})r_L + (4 - \rho_{\text{op}})(\delta - 1) \\
&\leq (6 - 3\rho_{\text{op}})(\rho_{\text{op}}\delta - (\delta - 1)) + (4 - \rho_{\text{op}})(\delta - 1) \\
&= \frac{3\rho_{\text{op}}^4 - 18\rho_{\text{op}}^3 + 3\rho_{\text{op}}^2 + 50\rho_{\text{op}} - 14}{3\rho_{\text{op}}^3 - 15\rho_{\text{op}} + 4} \\
&\overset{\rho_{\text{op}} < 2.06}{<} \delta \\
&= \text{pos}(r_L + (\delta - 1)),
\end{aligned}$$

i.e., ALG's position at time $r_L + (\delta - 1)$ is to the right of ℓ_{op}. The earliest possible time ALG crosses ℓ_{op} is the solution of

$$\ell_{\text{op}}(r_R) = (4 - \rho_{\text{op}})r_R - (2\rho_{\text{op}} - 2)r_L = \text{pos}(r_L + (\delta - 1)) + r_L + (\delta - 1) - r_R,$$

which is $r_R = \frac{2\rho_{\text{op}}-1}{5-\rho_{\text{op}}} r_L + \frac{2\delta-1}{5-\rho_{\text{op}}}$. Finally, the inequality

$$\begin{aligned}
\left(\frac{3\rho_{\text{op}} - 5}{7 - 3\rho_{\text{op}}} - \frac{2\rho_{\text{op}} - 1}{5 - \rho_{\text{op}}}\right) r_L &= \frac{3\rho_{\text{op}}^2 + 3\rho_{\text{op}} - 18}{3\rho_{\text{op}}^2 - 22\rho_{\text{op}} + 35} r_L \\
&\overset{\text{Lem 3.14}}{\leq} \frac{3\rho_{\text{op}}^2 + 3\rho_{\text{op}} - 18}{3\rho_{\text{op}}^2 - 22\rho_{\text{op}} + 35} (\rho_{\text{op}}\delta - (\delta - 1))) \\
&= \frac{3\rho_{\text{op}}^3 + 6\rho_{\text{op}}^2 - 15\rho_{\text{op}} - 18}{3\rho_{\text{op}}^4 - 15\rho_{\text{op}}^3 - 15\rho_{\text{op}}^2 + 79\rho_{\text{op}} - 20} \\
&= \frac{2\delta - 1}{5 - \rho_{\text{op}}}
\end{aligned}$$

implies inequality (3.18). □

In fact, also the other properties of o-ρ_{op}-critical requests are satisfied.

Lemma 3.16. *The requests s_R and s_L of the request sequence $\sigma'_{\rho_{\text{op}}}$ satisfy Definition 3.8.*

Proof. We have to show that the requests s_R and s_L of the request sequence $\sigma_{\rho_{\text{op}}}$ satisfy the properties (i) to (v) of Definition 3.8. The release time of every request is equal to its starting position, thus every request can be served/loaded immediately once its starting position is visited and (i) of Definition 3.8 is satisfied. At time r_R ALG has not served s_R, because for that it would have needed to go right from time 0 on; it has not served s_L either, because during the period of time $[t_L, t_R]$ ALG and s_L were on different sides of ℓ_{op}. This establishes the first part of (ii) of Definition 3.8. Furthermore, at time r_R ALG is at position $\text{pos}(r_R) = (4 - \rho_{\text{op}})r_R - (2\rho_{\text{op}} - 2)r_L$ with

$$-r_L \leq (4 - \rho_{\text{op}})r_R - (2\rho_{\text{op}} - 2)r_L \leq r_R.$$

Therefore, the second part of (ii) of Definition 3.8 is satisfied as well.

Lemma 3.15 shows that (iii) and (iv) of Definition 3.8 are satisfied. It remains to show that property (v) is satisfied. For this we need to examine the release time r_R of s_R. The time r_R is largest if ALG tries to avoid crossing the line ℓ_{op} for as long as possible, i.e., it continues to move right after serving the requests s_0^j. Then, we have $\text{pos}(t) = 1 - r_L + t$ for $t \in [r_L, r_R]$ and r_R is the solution of

$$1 - r_L + r_R = (4 - \rho_{\text{op}})r_R - (2\rho_{\text{op}} - 2)r_L.$$

Thus, in general, we have $r_R \leq \frac{2\rho_{\text{op}}-3}{3-\rho_{\text{op}}} r_L + \frac{1}{3-\rho_{\text{op}}}$, i.e.,

$$\frac{r_R}{r_L} \leq \frac{2\rho_{\text{op}} - 3}{3 - \rho_{\text{op}}} + \frac{1}{(3 - \rho_{\text{op}})r_L} \overset{r_L \geq 2}{\leq} \frac{4\rho_{\text{op}} - 5}{6 - 2\rho_{\text{op}}}.$$

For property (v), we need $\frac{r_R}{r_L} \leq \frac{4\rho_{\text{op}}^2 - 30\rho_{\text{op}} + 50}{-8\rho_{\text{op}}^2 + 50\rho_{\text{op}} - 66}$. This is satisfied if

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

which is equivalent to

$$4\rho_{\text{op}}^3 - 26\rho_{\text{op}}^2 + 39\rho_{\text{op}} - 5 \geq 0,$$

which is true by definition of ρ_{op}. □

Together with Lemma 3.9, this completes the proof of Theorem 3.2.

4 Schedule-Based Algorithms

Imagine the release time r_n of the last request s_n would be known. Then we could simply wait at the origin until at time r_n all requests have been released and start an optimum offline walk serving all requests. Since the optimum also has to serve the last request appearing at time r_n, we have $\textsc{Opt}(\sigma) \geq r_n$. Furthermore, the optimum walk that our server starts at time r_n is also bounded by $\textsc{Opt}(\sigma)$. Thus, this approach would be 2-competitive, which would be an improvement to the currently known upper bounds – at least for the open version of online TSP and Dial-a-Ride (see Table 2.1). Unfortunately though, we neither know the number of requests nor the release time of the last request. However, we can still let the server wait at the origin for some time and then start an optimum walk serving all currently known requests. This motivates the class of *schedule-based algorithms* (see Algorithm 1) for online TSP and online Dial-a-Ride.

Algorithm 1 schedule-based algorithm

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

Schedule-based algorithms all follow a simple design rule: Wait a certain amount of time (dependent on the available data at the current time t), then serve all currently known unserved requests in an optimum offline walk while ignoring all new incoming requests and repeat. To be more precise, let R_t be the subsequence of requests that have not been served yet at time t and let

$$\begin{aligned} \mathrm{wait} : \mathbb{R}_0^+ &\longrightarrow \{\mathsf{true}, \mathsf{false}\}, \\ t &\longmapsto \mathrm{wait}(t) \end{aligned}$$

be a function mapping a time t to either true or false. A schedule-based algorithm for online TSP and online DIAL-A-RIDE waits until a time t^* with $\mathrm{wait}(t^*) = \mathsf{false}$ and executes an optimum walk serving R_t.

We call the executed optimum offline walks *schedules* S_j and index them in chronological order, i.e., we have $S_1, \ldots, S_N$ with $N \in \mathbb{N}$ being the number of schedules the algorithm executes. In the open setting of online TSP and online DIAL-A-RIDE a schedule finishes at the position of the last served request, while in the closed setting every schedule ends in the origin. The starting time of S_j is denoted by by t_j and its ending time by v_{j+1}. Note that we always have $v_j \leq t_j$. The starting position of S_j is denoted by p_j and its ending position by p_{j+1}. The subsequence of requests served in S_j is denoted by σ_j. For convenience, we set $t_0 = p_0 = 0$. We define $L(t, p, R)$ to be the length of a shortest walk that starts at position p at time t and serves all requests in the subsequence $R \subseteq \sigma$ after they appeared. Consequently, for every schedule-based algorithm ALG, we have

$$\textsc{Alg}(\sigma) = t_N + L(t_N, p_N, \sigma_N). \tag{4.1}$$

Note that, by definition, a walk must respect release times, i.e., it might contain waiting times caused by late release times of requests. However, no schedule executed by a schedule-based algorithm contains waiting times since only already released requests are served. For all $0 \leq t \leq t'$, $p, p' \in X$, and $R \subseteq \sigma$, we have

$$L(t, p, R) \geq L(t', p, R), \tag{4.2}$$
$$L(t, p, R) \leq d(p, p') + L(t, p', R), \tag{4.3}$$
$$L(t, 0, R) \leq L(t, 0, \sigma) \leq L(0, 0, \sigma) = \textsc{Opt}(\sigma). \tag{4.4}$$

Inequality (4.2) holds since an earlier starting time might cause additional waiting times because of late releases of some requests. Inequality (4.3) is a consequence of the triangle inequality, and inequality (4.4) holds since $R \subseteq \sigma$, $t \geq 0$ and since the optimum trajectory starts in the origin at time 0. Note that the schedules used by schedule-based algorithms are NP-hard to compute for $1 < c < \infty$ [13].

Throughout this chapter, we let the capacity $c \in \mathbb{N} \cup \{\infty\}$ of the server be arbitrary but fixed. We start our examination of schedule-based algorithms with the computation of several lower bounds for their competitive ratios.

4.1 Lower Bounds for Schedule-Based Algorithms

In this subsection, we compute lower bounds for the competitive ratio of schedule-based algorithms. For the open version of the problem, we compute separate lower bounds for online DIAL-A-RIDE and online TSP. For the closed version, we only provide a lower bound for the TSP version, however, since online TSP is a special case of online DIAL-A-RIDE, the computed lower bound is also valid for online DIAL-A-RIDE. We start with open online DIAL-A-RIDE. All presented request sequences are on the real line $\mathbb{R}$.

Theorem 4.1. *Let* ALG *be a ρ-competitive schedule-based algorithm for open online* DIAL-A-RIDE *on the line. Then we have $\rho \geq \frac{5}{2}$.*

Proof. Let $s_1 = (1, 1; 0)$. If ALG starts its first schedule S_1 at time $t_1 \geq \frac{3}{2}$, we release no additional requests. In this case, we have OPT$((s_1)) = 1$ and ALG$((s_1)) \geq \frac{5}{2}$. If ALG starts its first schedule S_1 at time $t_1 = 0$, we release the request $s_2 = (\frac{1}{4}, 1; \frac{1}{4})$. In this case, we have OPT$((s_1, s_2)) = 1$ and ALG$((s_1, s_2)) \geq 1 + 2 \cdot \frac{3}{4} = \frac{5}{2}$. Thus, we may assume $0 < t_1 < \frac{3}{2}$ in the following. Let $\varepsilon > 0$ with $\varepsilon < \min\{\frac{3}{2} - t_1, t_1\} \leq \frac{3}{4}$. We release the requests

$$s_2^{(1)} = \left(-t_1 + \frac{1}{4}\varepsilon, -t_1 + \frac{1}{4}\varepsilon; t_1 + \frac{1}{4}\varepsilon\right),$$
$$s_2^{(2)} = \left(\frac{1 - t_1}{2} + \frac{1}{4}\varepsilon, 1; t_1 + \frac{1}{4}\varepsilon\right).$$

ALG finishes schedule S_1 at time $v_2 = t_1 + 1$ at position $p_2 = 1$. The shortest schedule serving $s_2^{(1)}$ before $s_2^{(2)}$ has length

$$D\left(1 \to -t_1 + \frac{1}{4}\varepsilon \to 1\right) = 2 + 2t_1 - \frac{1}{2}\varepsilon.$$

On the other hand, the shortest schedule that serves $s_2^{(1)}$ after $s_2^{(2)}$ has length

$$D\left(1 \to \frac{1 - t_1}{2} + \frac{1}{4}\varepsilon \to 1 \to -t_1 + \frac{1}{4}\varepsilon\right) = 2 + 2t_1 - \frac{3}{4}\varepsilon.$$

Therefore, for all $t \geq v_2$, we have

$$L(t, p_2, (s_2^{(1)}, s_2^{(2)})) = 2 + 2t_1 - \frac{3}{4}\varepsilon \tag{4.5}$$

and schedule S_2 ends at position $p_3 = -t_1 + \frac{1}{4}\varepsilon$. We make a case distinction depending on the starting time t_2 of schedule S_2.

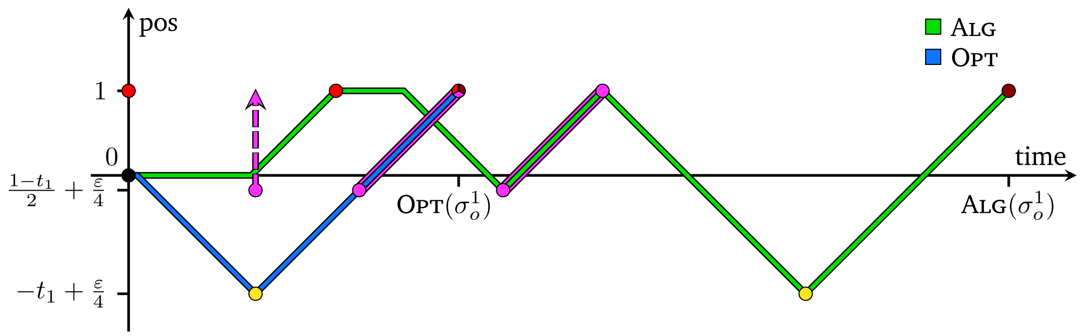

Figure 4.1: ALG's and OPT's walk serving σ_o^1 with $\varepsilon = 0.2$, $t_1 = 1.45$ and $t_2 = 3.25$. Request s_1 is red, request $s_2^{(1)}$ is yellow, request $s_2^{(2)}$ is violet and request s_3 is brown.

Case 1: $t_1 + 1 \leq t_2 < 2t_1 + 1$

We release $s_3 = (1, 1; 2t_1 + 1)$ and define $\sigma_o^1 := (s_1, s_2^{(1)}, s_2^{(2)}, s_3)$. See Figure 4.1 for ALG's walk (green) and OPT's walk (blue). OPT waits at the origin until time $\frac{1}{2}\varepsilon$ and then performs the walk

$$0 \to -t_1 + \frac{1}{4}\varepsilon \to 1.$$

Therefore, we have

$$\text{OPT}(\sigma_o^1) = \frac{1}{2}\varepsilon + D\left(0 \to -t_1 + \frac{1}{4}\varepsilon \to 1\right) = 2t_1 + 1. \tag{4.6}$$

For ALG, we obtain

$$\begin{aligned} v_3 &= t_2 + L(t_2, p_2, (s_2^{(1)}, s_2^{(2)})) \\ &\overset{t_2 \geq t_1 + 1}{\geq} t_1 + 1 + L(t_2, p_2, (s_2^{(1)}, s_2^{(2)})) \\ &\overset{(4.5)}{=} 3t_1 + 3 - \frac{3}{4}\varepsilon. \end{aligned} \tag{4.7}$$

For all $t \geq v_3$, we have

$$L(t, p_3, (s_3)) = D\left(-t_1 + \frac{1}{4}\varepsilon \to 1\right) = 1 + t_1 - \frac{1}{4}\varepsilon \tag{4.8}$$

since $v_3 > r_3$. Finally, we obtain

$$\text{ALG}(\sigma_o^1) \overset{(4.1)}{=} t_3 + L(t_3, p_3, (s_3))$$

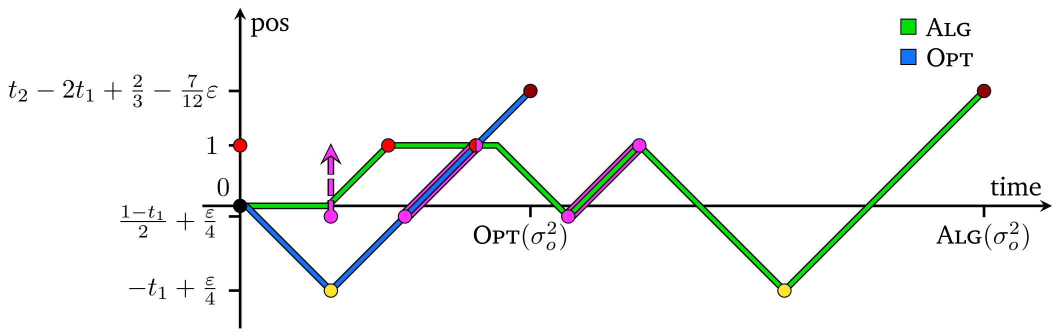

Figure 4.2: Alg's and Opt's walk serving σ_o^2 with $\varepsilon = 0.2$, $t_1 = 1.45$ and $t_2 = 4.25$. Request s_1 is red, request $s_2^{(1)}$ is yellow, request $s_2^{(2)}$ is violet and request s_3 is brown.

$$
\begin{aligned}
&\geq && v_3 + L(t_3, p_3, (s_3)) \\
&\overset{(4.7),(4.8)}{\geq} && 4t_1 + 4 - \varepsilon \\
&\overset{\varepsilon < \frac{3}{2} - t_1}{>} && 5t_1 + \frac{5}{2} \\
&\overset{(4.6)}{=} && \frac{5}{2}\textsc{Opt}(\sigma_o^1).
\end{aligned}
$$

Case 2: $2t_1 + 1 \leq t_2 < 2t_1 + 2 - \frac{1}{4}\varepsilon$
In this case, we have

$$t_1 > \frac{1}{2}t_2 - 1 + \frac{1}{8}\varepsilon. \tag{4.9}$$

We release the request

$$s_3 = \left(t_2 - 2t_1 + \frac{2}{3} - \frac{7}{12}\varepsilon, t_2 - 2t_1 + \frac{2}{3} - \frac{7}{12}\varepsilon; t_2 + \frac{2}{3} - \frac{7}{12}\varepsilon\right)$$

and define $\sigma_o^2 := (s_1, s_2^{(1)}, s_2^{(2)}, s_3)$. See Figure 4.2 for Alg's walk (green) and Opt's walk (blue). Note that we have $\frac{2}{3} - \frac{7}{12}\varepsilon > 0$, since $\varepsilon < \frac{3}{4}$, i.e., s_3 is released after S_2 is started. Opt waits at the origin until time $\frac{1}{2}\varepsilon$ and then performs the walk

$$0 \to -t_1 + \frac{1}{4}\varepsilon \to t_2 - 2t_1 + \frac{2}{3} - \frac{7}{12}\varepsilon.$$

Therefore, we have

$$\textsc{Opt}(\sigma_o^2) = \frac{1}{2}\varepsilon + D\left(0 \to -t_1 + \frac{1}{4}\varepsilon \to t_2 - 2t_1 + \frac{2}{3} - \frac{7}{12}\varepsilon\right)$$

$$= t_2 + \frac{2}{3} - \frac{7}{12}\varepsilon. \tag{4.10}$$

For Alg, we obtain

$$v_3 = t_2 + L(t, p_2, (s_2^{(1)}, s_2^{(2)})) \overset{(4.5)}{=} t_2 + 2 + 2t_1 - \frac{3}{4}\varepsilon. \tag{4.11}$$

For all $t \geq v_3$, we have

$$\begin{aligned} L(t, p_3, (s_3)) &= D\left(-t_1 + \frac{1}{4}\varepsilon \to t_2 - 2t_1 + \frac{2}{3} - \frac{7}{12}\varepsilon\right) \\ &= t_2 - t_1 - \frac{5}{6}\varepsilon + \frac{2}{3} \end{aligned} \tag{4.12}$$

since $v_3 > r_3$. Finally, we obtain

$$\begin{aligned} \text{Alg}(\sigma_o^2) &\overset{(4.1)}{=} t_3 + L(t_3, p_3, (s_3)) \\ &\geq v_3 + L(t_3, p_3, (s_3)) \\ &\overset{(4.11),(4.12)}{=} 2t_2 + t_1 + \frac{8}{3} - \frac{19}{12}\varepsilon \\ &\overset{(4.9)}{>} \frac{5}{2}t_2 + \frac{5}{3} - \frac{35}{24}\varepsilon \\ &= \frac{5}{2}\left(t_2 + \frac{2}{3} - \frac{7}{12}\varepsilon\right) \\ &\overset{(4.10)}{=} \frac{5}{2}\text{Opt}(\sigma_o^2). \end{aligned}$$

Case 3: $t_2 \geq 2t_1 + 2 - \frac{1}{4}\varepsilon$

We release no new requests and define $\sigma_o^3 := (s_1, s_2^{(1)}, s_2^{(2)})$. See Figure 4.3 for Alg's walk (green) and Opt's walk (blue). Opt waits at the origin until time $\frac{1}{2}\varepsilon$ and then performs the walk

$$0 \to -t_1 + \frac{1}{4}\varepsilon \to 1.$$

Therefore, we have

$$\text{Opt}(\sigma_o^3) = \frac{1}{2}\varepsilon + D\left(0 \to -t_1 + \frac{1}{4}\varepsilon \to 1\right) = 2t_1 + 1. \tag{4.13}$$

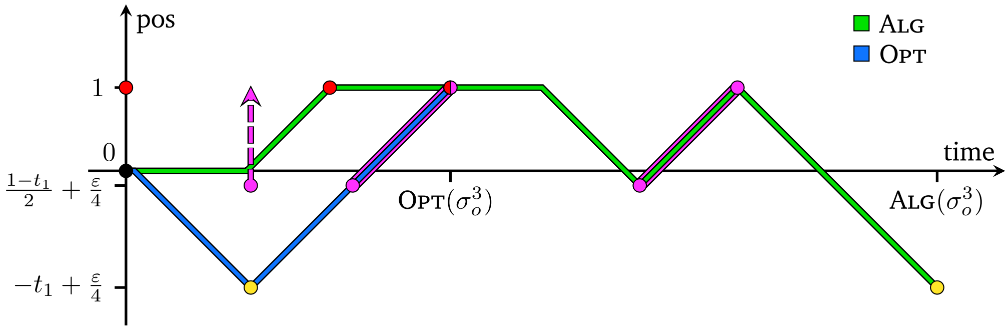

Figure 4.3: ALG's and OPT's walk serving σ_o^3 with $\varepsilon = 0.2$, $t_1 = 1.45$ and $t_2 = 5$. Request s_1 is red, request $s_2^{(1)}$ is yellow and request $s_2^{(2)}$ is violet.

For ALG, we obtain

$$\begin{aligned}
\text{ALG}(\sigma_o^3) &\overset{(4.1)}{=} t_2 + L(t_2, p_2, (s_2^{(1)}, s_2^{(2)})) \\
&\overset{t_2 \geq 2t_1 + 2 - \frac{1}{4}\varepsilon}{\geq} 2t_1 + 2 - \frac{1}{4}\varepsilon + L(t_2, p_2, (s_2^{(1)}, s_2^{(2)})) \\
&\overset{(4.5)}{=} 4t_1 + 4 - \varepsilon \\
&\overset{\varepsilon < \frac{3}{2} - t_1}{>} 5t_1 + \frac{5}{2} \\
&\overset{(4.13)}{=} \frac{5}{2}\text{OPT}(\sigma_o^3).
\end{aligned}$$

□

For open online TSP we obtain a slightly weaker bound since we are not allowed to use transportation requests.

Theorem 4.2. *Let* ALG *be a ρ-competitive schedule-based algorithm for open online* TSP *on the line. Then we have $\rho \geq \frac{7}{3}$.*

Proof. Let $s_1 = (1; 0)$. If ALG starts its first schedule S_1 at time $t_1 \geq \frac{4}{3}$, we release no additional requests. In this case, we have OPT$((s_1)) = 1$ and ALG$((s_1)) = \frac{7}{3}$. Thus, we may assume in the following $t_1 < \frac{4}{3}$. Let $\varepsilon > 0$ with $\varepsilon < \frac{4}{3} - t_1$. We release the requests

$$\begin{aligned}
s_2^{(1)} &= \left(-t_1 - \frac{1}{4}\varepsilon; t_1 + \frac{1}{4}\varepsilon\right), \\
s_2^{(2)} &= \left(2 + t_1 - \frac{1}{2}\varepsilon; t_1 + \frac{1}{4}\varepsilon\right).
\end{aligned}$$

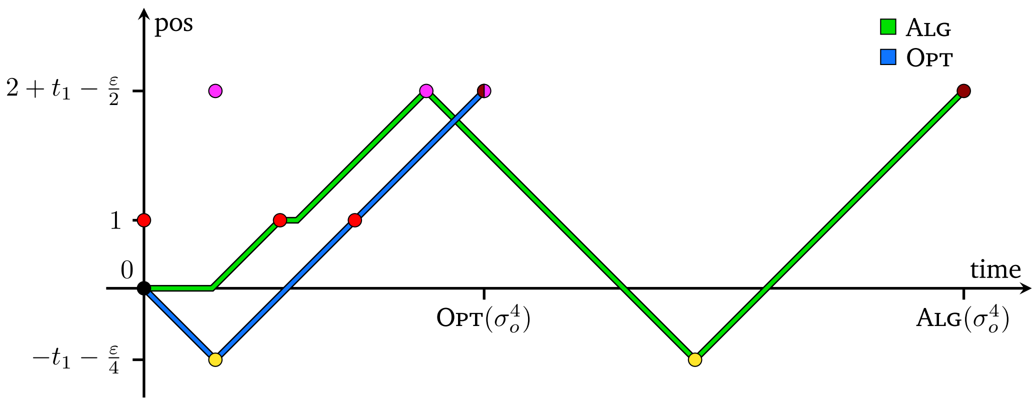

Figure 4.4: ALG's and OPT's walk serving σ_o^4 with $\varepsilon = 0.2$, $t_1 = 1$ and $t_2 = 2.25$. Request s_1 is red, request $s_2^{(1)}$ is yellow, request $s_2^{(2)}$ is violet and request s_3 is brown.

ALG finishes schedule S_1 at time $v_2 = t_1 + 1$ at position $p_2 = 1$. The shortest schedule serving $s_2^{(1)}$ before $s_2^{(2)}$ has length

$$D\left(1 \to -t_1 - \frac{1}{4}\varepsilon \to 2 + t_1 - \frac{1}{2}\varepsilon\right) = 3 + 3t_1.$$

On the other hand, the shortest schedule that serves $s_2^{(1)}$ after $s_2^{(2)}$ has length

$$D\left(1 \to 2 + t_1 - \frac{1}{2}\varepsilon \to -t_1 - \frac{1}{4}\varepsilon\right) = 3 + 3t_1 - \frac{3}{4}\varepsilon.$$

Therefore, for all $t \geq v_2$, we have

$$L(t, p_2, (s_2^{(1)}, s_2^{(2)})) = 3 + 3t_1 - \frac{3}{4}\varepsilon \tag{4.14}$$

since $v_2 > r_2$. Thus, schedule S_2 ends at position $p_3 = -t_1 - \frac{1}{4}\varepsilon$. We make a case distinction depending on the starting time t_2 of schedule S_2.

Case 1: $t_1 + 1 \leq t_2 < 3t_1 + 2$

We release $s_3 = (2 + t_1 - \frac{1}{2}\varepsilon; 3t_1 + 2)$ and define $\sigma_o^4 := (s_1, s_2^{(1)}, s_2^{(2)}, s_3)$. See Figure 4.4 for ALG's walk (green) and OPT's walk (blue). Assume OPT performs the walk

$$0 \to -t_1 - \frac{1}{4}\varepsilon \to 2 + t_1 - \frac{1}{2}\varepsilon.$$

Then, $\textsc{Opt}$ reaches position $a_3 = 2 + t_1 - \frac{1}{2}\varepsilon$ at time $3t_1 + 2 = r_3$, i.e., $\textsc{Opt}$ can serve s_3 upon arrival. Therefore, we have

$$\textsc{Opt}(\sigma_o^4) = D\left(0 \to -t_1 - \frac{1}{4}\varepsilon \to 2 + t_1 - \frac{1}{2}\varepsilon\right) = 3t_1 + 2. \tag{4.15}$$

For $\textsc{Alg}$, we obtain

$$\begin{aligned} v_3 &= t_2 + L(t_2, p_2, (s_2^{(1)}, s_2^{(2)})) \\ &\overset{t_2 \geq t_1 + 1}{\geq} t_1 + 1 + L(t_2, p_2, (s_2^{(1)}, s_2^{(2)})) \\ &\overset{(4.14)}{=} 4 + 4t_1 - \frac{3}{4}\varepsilon. \end{aligned} \tag{4.16}$$

For all $t \geq v_3$, we have

$$L(t, p_3, (s_3)) = D\left(-t_1 - \frac{1}{4}\varepsilon \to 2 + t_1 - \frac{1}{2}\varepsilon\right) = 2 + 2t_1 - \frac{1}{4}\varepsilon \tag{4.17}$$

since $v_3 > r_3$. Finally, we obtain

$$\begin{aligned} \textsc{Alg}(\sigma_o^4) &\overset{(4.1)}{=} t_3 + L(t_3, p_3, (s_3)) \\ &\geq v_3 + L(t_3, p_3, (s_3)) \\ &\overset{(4.16),(4.17)}{\geq} 6t_1 + 6 - \varepsilon \\ &\overset{\varepsilon < \frac{4}{3} - t_1}{>} 7t_1 + \frac{14}{3} \\ &\overset{(4.15)}{=} \frac{7}{3}\textsc{Opt}(\sigma_o^4). \end{aligned}$$

Case 2: $3t_1 + 2 \leq t_2 < 3t_1 + 3 - \frac{1}{4}\varepsilon$
in this case, we have

$$t_1 > \frac{1}{3}t_2 - 1 + \frac{1}{12}\varepsilon. \tag{4.18}$$

We release the request

$$s_3 = \left(\frac{5}{4}t_2 - 2t_1 + \frac{3}{4} - \frac{9}{8}\varepsilon; \frac{5}{4}t_2 + \frac{3}{4} - \frac{5}{8}\varepsilon\right).$$

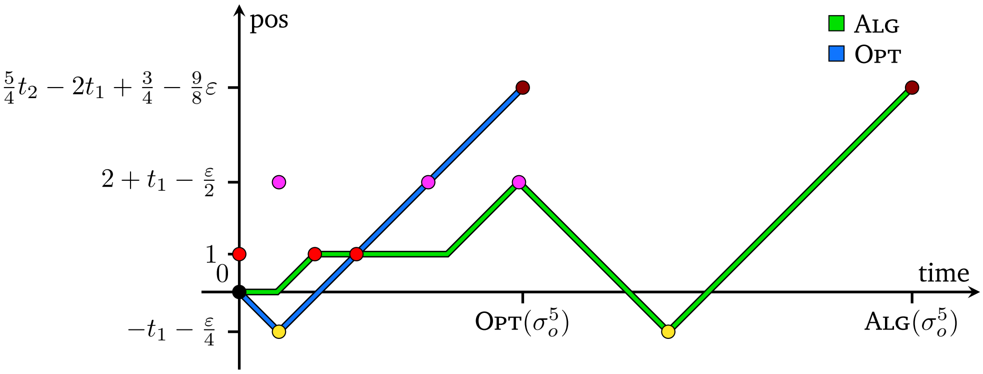

Figure 4.5: ALG's and OPT's walk serving σ_o^5 with $\varepsilon = 0.2$, $t_1 = 1$ and $t_2 = 5.5$. Request s_1 is red, request $s_2^{(1)}$ is yellow, request $s_2^{(2)}$ is violet and request s_3 is brown.

and define $\sigma_o^5 := (s_1, s_2^{(1)}, s_2^{(2)}, s_3)$. See Figure 4.5 for ALG's walk (green) and OPT's walk (blue). Note that we have

$$r_3 = \frac{5}{4}t_2 + \frac{3}{4} - \frac{5}{8}\varepsilon > t_2$$

because of $t_2 \geq 3t_1 + 2$ and $\varepsilon < \frac{4}{3} - t_1$, i.e., s_3 is released after S_2 is started. OPT peforms the walk

$$0 \to -t_1 - \frac{1}{4}\varepsilon \to \frac{5}{4}t_2 - 2t_1 + \frac{3}{4} - \frac{9}{8}\varepsilon.$$

Therefore, we have

$$\begin{aligned}\textsc{Opt}(\sigma_o^5) &= D\left(0 \to -t_1 - \frac{1}{4}\varepsilon \to \frac{5}{4}t_2 - 2t_1 + \frac{3}{4} - \frac{9}{8}\varepsilon\right) \\ &= \frac{5}{4}t_2 + \frac{3}{4} - \frac{5}{8}\varepsilon. \qquad (4.19)\end{aligned}$$

For ALG, we obtain

$$v_3 = t_2 + L(t_2, p_2, (s_2^{(1)}, s_2^{(2)})) \overset{(4.14)}{=} t_2 + 3 + 3t_1 - \frac{3}{4}\varepsilon. \qquad (4.20)$$

For all $t \geq v_3$, we have

$$\begin{aligned}L(t, p_3, (s_3)) &= D\left(-t_1 - \frac{1}{4}\varepsilon \to \frac{5}{4}t_2 - 2t_1 + \frac{3}{4} - \frac{9}{8}\varepsilon\right) \\ &= \frac{5}{4}t_2 - t_1 + \frac{3}{4} - \frac{7}{8}\varepsilon \qquad (4.21)\end{aligned}$$

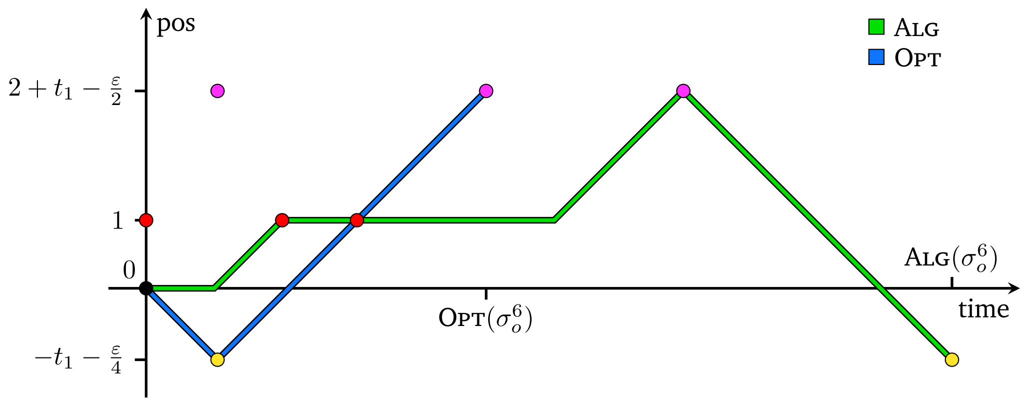

Figure 4.6: Alg's and Opt's walk serving σ_o^6 with $\varepsilon = 0.2$, $t_1 = 1$ and $t_2 = 6$. Request s_1 is red ●, request $s_2^{(1)}$ is yellow ● and request $s_2^{(2)}$ is violet ●.

since $v_3 > r_3$. Finally, we obtain

$$\begin{aligned}\text{Alg}(\sigma_o^5) &\overset{(4.1)}{=} t_3 + L(t_3, p_3, (s_3))\\ &\geq v_3 + L(t_3, p_3, (s_3))\\ &\overset{(4.20),(4.21)}{=} \frac{9}{4}t_2 + 2t_1 + \frac{15}{4} - \frac{13}{8}\varepsilon\\ &\overset{(4.18)}{>} \frac{35}{12}t_2 + \frac{7}{4} - \frac{35}{24}\varepsilon\\ &= \frac{7}{3}\left(\frac{5}{4}t_2 + \frac{3}{4} - \frac{5}{8}\varepsilon\right)\\ &\overset{(4.19)}{=} \frac{7}{3}\text{Opt}(\sigma_o^5).\end{aligned}$$

Case 3: $t_2 \geq 3t_1 + 3 - \frac{1}{4}\varepsilon$

We release no new requests and define $\sigma_o^6 := (s_1, s_2^{(1)}, s_2^{(2)})$. See Figure 4.6 for Alg's walk (green) and Opt's walk (blue). Opt performs the walk

$$0 \to -t_1 - \frac{1}{4}\varepsilon \to 2 + t_1 - \frac{1}{2}\varepsilon.$$

Therefore, we have

$$\text{Opt}(\sigma_o^6) = D\left(0 \to -t_1 - \frac{1}{4}\varepsilon \to 2 + t_1 - \frac{1}{2}\varepsilon\right) = 3t_1 + 2. \tag{4.22}$$

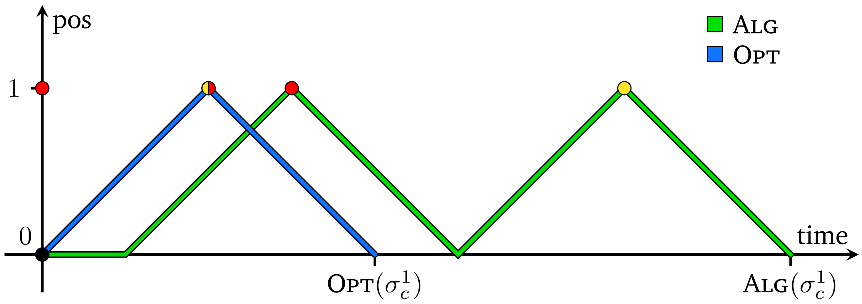

Figure 4.7: ALG's and OPT's walk serving σ_c^1 with $t_1 = 0.5$. Request s_1 is red ● and request s_2 is yellow ●.

For ALG, we obtain

$$\begin{aligned}
\textsc{Alg}(\sigma_o^6) &\overset{(4.1)}{=} t_2 + L(t_2, p_2, (s_2^{(1)}, s_2^{(2)})) \\
&\overset{t_2 \geq 3t_1 + 3 - \frac{1}{4}\varepsilon}{\geq} 3t_1 + 3 - \frac{1}{4}\varepsilon + L(t_2, p_2, (s_2^{(1)}, s_2^{(2)})) \\
&\overset{(4.14)}{=} 6t_1 + 6 - \varepsilon \\
&\overset{\varepsilon < \frac{4}{3} - t_1}{>} 7t_1 + \frac{14}{3} \\
&\overset{(4.22)}{=} \frac{7}{3}\textsc{Opt}(\sigma_o^6).
\end{aligned}$$

□

Finally, we present an lower bound for the competitive ratio of closed online TSP on the line. Since online DIAL-A-RIDE is a special case of online TSP, this lower bound is also valid for online DIAL-A-RIDE. Note that the following lower bound utilizes only requests with positions on the positive side of the origin. Therefore, this lower bound is also valid for online TSP on the *half-line*.

Theorem 4.3. *Let* ALG *be a ρ-competitive schedule-based algorithm for closed online* TSP *on the line. Then we have $\rho \geq 2$.*

Proof. Let $s_1 = (1; 0)$. For all $t \geq 0$, we have

$$L(t, 0, (s_1)) = D(0 \to 1 \to 0) = 2 \tag{4.23}$$

and thus, ALG finishes schedule S_1 at time

$$v_2 = t_1 + 2. \tag{4.24}$$

We make a case distinction based on the starting time t_1 of schedule S_1.

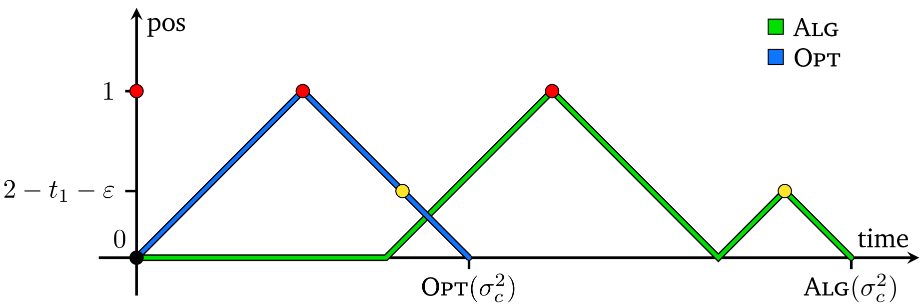

Figure 4.8: ALG's and OPT's walk serving σ_c^2 with $\varepsilon = 0.1$ and $t_1 = 1.5$. Request s_1 is red ● and request s_2 is yellow ○.

Case 1: $t_1 < 1$

We release the request $s_2 = (1; 1)$ and define $\sigma_c^1 := (s_1, s_2)$. See Figure 4.7 for ALG's walk (green) and OPT's walk (blue). OPT performs the walk $0 \to 1 \to 0$. Therefore, we have

$$\text{OPT}(\sigma_c^1) = 2. \tag{4.25}$$

For the second and last schedule S_2 of ALG, we have

$$L(t_2, 0, (s_2)) = D(0 \to 1 \to 0) = 2. \tag{4.26}$$

Thus, we obtain

$$\begin{aligned} \text{ALG}(\sigma_c^1) &\overset{(4.1)}{=} t_2 + L(t_2, 0, (s_2)) \\ &\geq v_2 + L(t_2, 0, (s_2)) \\ &\overset{(4.24),(4.26)}{=} 4 \\ &\overset{(4.25)}{=} 2\text{OPT}(\sigma_c^1). \end{aligned}$$

Case 2: $1 \leq t_1 < 2$

Let $\varepsilon > 0$ with $\varepsilon < 1 - \frac{1}{2}t_1$. We release the request $s_2 = (2 - t_1 - \varepsilon; t_1 + \varepsilon)$ and define $\sigma_c^2 := (s_1, s_2)$. See Figure 4.8 for ALG's walk (green) and OPT's walk (blue). Assume OPT performs the walk $0 \to 1 \to 0$. At time $r_2 = t_1 + \varepsilon$, OPT is at position $2 - t_1 - \varepsilon = a_2$, i.e., OPT can serve s_2 upon arrival. Therefore, we have

$$\text{OPT}(\sigma_c^2) = 2. \tag{4.27}$$

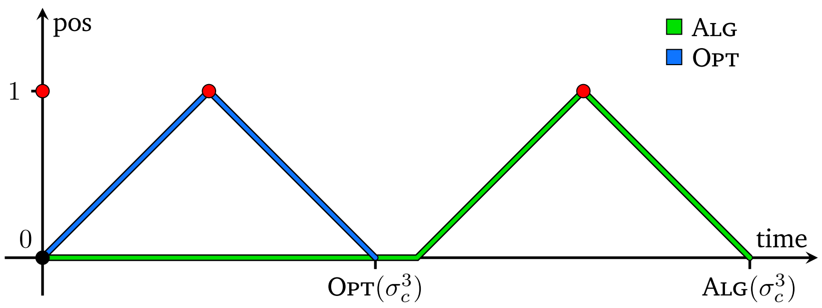

Figure 4.9: ALG's and OPT's walk serving σ_c^3 with $t_1 = 2.25$. Request s_1 is red ●.

For the second and last schedule S_2 of ALG, we have

$$L(t_2, 0, (s_2)) = D(0 \to 2 - t_1 - \varepsilon \to 0) = 4 - 2t_1 - 2\varepsilon. \tag{4.28}$$

Thus, we obtain

$$\begin{aligned}
\textsc{Alg}(\sigma_c^2) &\overset{(4.1)}{=} t_2 + L(t_2, 0, (s_2)) \\
&\geq v_2 + L(t_2, 0, (s_2)) \\
&\overset{(4.24),(4.28)}{=} 6 - t_1 - 2\varepsilon \\
&\overset{\varepsilon < 1 - \frac{1}{2}t_1}{>} 4 \\
&\overset{(4.27)}{=} 2\textsc{Opt}(\sigma_c^2).
\end{aligned}$$

Case 3: $t_1 \geq 2$

We release no new requests and define $\sigma_c^3 := (s_1)$. See Figure 4.9 for ALG's walk (green) and OPT's walk (blue). OPT performs the walk $0 \to 1 \to 0$. Therefore, we have

$$\textsc{Opt}(\sigma_c^3) = 2. \tag{4.29}$$

For ALG, we obtain

$$\begin{aligned}
\textsc{Alg}(\sigma_c^3) &\overset{(4.1)}{=} t_1 + L(t_1, 0, (s_1)) \\
&\overset{(4.23)}{=} t_1 + 2 \\
&\overset{t_1 \geq 2}{\geq} 4 \\
&\overset{(4.29)}{=} 2\textsc{Opt}(\sigma_c^3).
\end{aligned}$$

□

In the following, we will analyze three different schedule-based algorithms. To examine their competitiveness, it is integral to understand the structure of the executed optimum offline schedules. Because of that, we give a thorough analysis of schedules in the next section.

4.2 Schedules

We published the results of this section also in [10] and used the results in [12]. In this section we take a closer look at the lengths of the optimum schedules executed by schedule-based algorithms. While schedule-based algorithms may vary drastically in their waiting routines, some results for their schedules are universal and indispensable ingredients for their analysis. Interestingly, schedules are easy to analyze in the closed version of online Dial-a-Ride and online TSP, while their analysis is highly non-trivial in the open version. In the closed version, every schedule is a closed walk ending at the origin. Therefore, every schedule executed by a schedule-based algorithm for the closed version also starts at the origin, i.e., we have $p_j = 0$ for all schedules S_j. Inequality (4.4) then implies the following lemma.

Lemma 4.4. *For every schedule S_j of a schedule-based algorithm for closed online* Dial-a-Ride*, we have*

$$L(t_j, p_j, \sigma_j) \leq \textsc{Opt}(\sigma).$$

In our analysis of the open version we distinguish between bounds for the competitive ratio in the general setting, i.e., on arbitrary continuous metric spaces and bounds that only hold on the real line. We start with the more general bounds.

Open Version on General Metric Spaces

In this subsection, we give bounds for the length of a schedule in terms of the size of $\textsc{Opt}(\sigma)$ as well as the starting position of the schedule and the starting time of the previous schedule.

Lemma 4.5. *For every schedule S_j of a schedule-based algorithm for open online* Dial-a-Ride*, we have*

$$L(t_j, p_j, \sigma_j) \leq \min\{\textsc{Opt}(\sigma) + d(0, p_j), 2(\textsc{Opt}(\sigma) - t_{j-1})\}.$$

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

$$L(t_j, p_j, \sigma_j) \overset{(4.3)}{\leq} d(0, p_j) + L(t_j, 0, \sigma_j) \overset{(4.4)}{\leq} \textsc{Opt}(\sigma) + d(0, p_j). \tag{4.30}$$

Now, let $s_j^{\textsc{Opt}}$ be the first request of σ_j that is picked up by Opt and let $a_j^{\textsc{Opt}}$ be its starting position and $r_j^{\textsc{Opt}}$ be its release time. We have

$$L(t_j, p_j, \sigma_j) \overset{(4.3)}{\leq} d(p_j, a_j^{\textsc{Opt}}) + L(t_j, a_j^{\textsc{Opt}}, \sigma_j), \tag{4.31}$$

again by the triangle inequality. Since Opt serves all requests of σ_j starting at position $a_j^{\textsc{Opt}}$ no earlier than time $r_j^{\textsc{Opt}}$, we have

$$L(t_j, a_j^{\textsc{Opt}}, \sigma_j) \overset{(4.2)}{\leq} L(r_j^{\textsc{Opt}}, a_j^{\textsc{Opt}}, \sigma_j) \leq \textsc{Opt}(\sigma) - r_j^{\textsc{Opt}}, \tag{4.32}$$

which yields

$$\begin{aligned} L(t_j, p_j, \sigma_j) &\overset{(4.31)}{\leq} d(p_j, a_j^{\textsc{Opt}}) + L(t_j, a_j^{\textsc{Opt}}, \sigma_j) \\ &\overset{(4.32)}{\leq} \textsc{Opt}(\sigma) + d(p_j, a_j^{\textsc{Opt}}) - r_j^{\textsc{Opt}} \\ &\overset{t_{j-1} < r_j^{\textsc{Opt}}}{<} \textsc{Opt}(\sigma) + d(p_j, a_j^{\textsc{Opt}}) - t_{j-1}. \end{aligned} \tag{4.33}$$

Since p_j is the destination of a request, Opt needs to visit it. In the case that Opt visits p_j before collecting $s_j^{\textsc{Opt}}$, we have

$$\begin{aligned} \textsc{Opt}(\sigma) + d(0, p_j) &\geq \textsc{Opt}(\sigma) \\ &\geq d(p_j, a_j^{\textsc{Opt}}) + L(t_j, a_j^{\textsc{Opt}}, \sigma_j) \\ &\overset{(4.31)}{\geq} L(t_j, p_j, \sigma_j). \end{aligned}$$

On the other hand, if Opt collects $s_j^{\textsc{Opt}}$ before visiting the position p_j, we have

$$t_{j-1} + d(p_j, a_j^{\textsc{Opt}}) \overset{t_{j-1} < r_j^{\textsc{Opt}}}{<} r_j^{\textsc{Opt}} + d(p_j, a_j^{\textsc{Opt}}) \leq \textsc{Opt}(\sigma), \tag{4.34}$$

since Opt cannot collect $s_j^{\textsc{Opt}}$ before time $r_j^{\textsc{Opt}}$ and then still has to visit position p_j. Thus, we have

$$\begin{aligned} L(t_j, p_j, \sigma_j) &\overset{(4.33)}{<} \textsc{Opt}(\sigma) + d(p_j, a_j^{\textsc{Opt}}) - t_{j-1} \\ &\overset{(4.34)}{\leq} 2\textsc{Opt}(\sigma) - 2t_{j-1}. \end{aligned} \tag{4.35}$$

This implies

$$L(t_j, p_j, \sigma_j) \overset{(4.30),(4.35)}{\leq} \min\{\textsc{Opt}(\sigma) + d(0, p_j), 2(\textsc{Opt}(\sigma) - t_{j-1})\}. \qquad \square$$

Interestingly, we can improve the bound provided by Lemma 4.5 if we disallow transportation requests.

Lemma 4.6. *For every schedule S_j of a schedule-based algorithm for online* TSP*, we have*

$$L(t_j, p_j, \sigma_j) \leq \min\left\{\textsc{Opt}(\sigma) + d(0, p_j), \frac{3}{2}(\textsc{Opt}(\sigma) - t_{j-1})\right\}.$$

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

$$L(t_j, p_j, \sigma_j) \overset{(4.3)}{\leq} d(0, p_j) + L(t_j, 0, \sigma_j) \overset{(4.4)}{\leq} \textsc{Opt}(\sigma) + d(0, p_j). \tag{4.36}$$

Now, let $s_j^{\text{first}} = (a_j^{\text{first}}; r_j^{\text{first}})$ be the first request of σ_j that is served by Opt and let $s_j^{\text{last}} = (a_j^{\text{last}}; r_j^{\text{last}})$ be the last request of σ_j that is served by Opt. Furthermore, let $W_j^{\textsc{Opt}}$ be Opt's walk between serving s_j^{first} and s_j^{last} and let $c(W_j^{\textsc{Opt}})$ be its length. We have

$$c(W_j^{\textsc{Opt}}) \geq L(t_j, a_j^{\text{first}}, \sigma_j), \tag{4.37}$$

since the walk $W_j^{\textsc{Opt}}$ serves every request of σ_j starting from position a_j^{first}. However, we also have

$$c(W_j^{\textsc{Opt}}) \geq L(t_j, a_j^{\text{last}}, \sigma_j), \tag{4.38}$$

since walking $W_j^{\textsc{Opt}}$ backwards, i.e., starting from position a_j^{last}, also serves every request of σ_j, while walking the same distance. Note that this is only the case since we do not have transportation requests. Using the triangle inequality and the inequalities above, we obtain

$$L(t_j, p_j, \sigma_j) \overset{(4.3)}{\leq} d(p_j, a_j^{\text{first}}) + L(t_j, a_j^{\text{first}}, \sigma_j) \overset{(4.37)}{\leq} d(p_j, a_j^{\text{first}}) + c(W_j^{\textsc{Opt}}) \tag{4.39}$$

and

$$L(t_j, p_j, \sigma_j) \overset{(4.3)}{\leq} d(p_j, a_j^{\text{last}}) + L(t_j, a_j^{\text{last}}, \sigma_j) \overset{(4.38)}{\leq} d(p_j, a_j^{\text{last}}) + c(W_j^{\textsc{Opt}}). \tag{4.40}$$

Combining the inequalities (4.39) and (4.40), we get

$$L(t_j, p_j, \sigma_j) \overset{(4.39),(4.40)}{\leq} \min\{d(p_j, a_j^{\text{first}}), d(p_j, a_j^{\text{last}})\} + c(W_j^{\textsc{Opt}}). \tag{4.41}$$

Since p_j is the position of a request, Opt needs to visit it. In the case that Opt visits p_j before serving s_j^{first}, we have

$$\textsc{Opt}(\sigma) + d(0, p_j) \quad \geq \quad \textsc{Opt}(\sigma)$$

$$\begin{aligned} &\geq d(p_j, a_j^{\text{first}}) + c(W_j^{\text{OPT}}) \\ &\overset{(4.39)}{\geq} L(t_j, p_j, \sigma_j). \end{aligned}$$

In the case that OPT visits p_j after serving s_j^{last}, we have

$$\begin{aligned} \text{OPT}(\sigma) + d(0, p_j) &\geq \text{OPT}(\sigma) \\ &\geq c(W_j^{\text{OPT}}) + d(p_j, a_j^{\text{last}}) \\ &\overset{(4.40)}{\geq} L(t_j, p_j, \sigma_j), \end{aligned}$$

as claimed. Thus, we may assume that OPT visits p_j after serving s_j^{first} and before serving s_j^{last}. Since OPT cannot serve s_j^{first} before time r_j^{first}, this implies

$$\text{OPT}(\sigma) \geq r_j^{\text{first}} + d(a_j^{\text{first}}, p_j) + d(p_j, a_j^{\text{last}}),$$

i.e.,

$$\min\{d(p_j, a_j^{\text{first}}), d(p_j, a_j^{\text{last}})\} \leq \frac{1}{2}\Big(\text{OPT}(\sigma) - r_j^{\text{first}}\Big). \tag{4.42}$$

Again, since OPT serves all requests of σ_j starting at position a_j^{first} no earlier than time r_j^{first}, we have

$$\text{OPT}(\sigma) \geq r_j^{\text{first}} + c(W_j^{\text{OPT}}). \tag{4.43}$$

Combining the inequalities (4.42) and (4.43), we get

$$\begin{aligned} L(t_j, p_j, \sigma_j) &\overset{(4.41)}{\leq} \min\{d(p_j, a_j^{\text{first}}), d(p_j, a_j^{\text{last}})\} + c(W_j^{\text{OPT}}) \\ &\overset{(4.42),(4.43)}{\leq} \frac{3}{2}\Big(\text{OPT}(\sigma) - r_j^{\text{first}}\Big) \\ &\overset{r_j^{\text{first}} > t_{j-1}}{<} \frac{3}{2}(\text{OPT}(\sigma) - t_{j-1}). \end{aligned} \tag{4.44}$$

This implies

$$L(t_j, p_j, \sigma_j) \overset{(4.36),(4.44)}{\leq} \min\Big\{\text{OPT}(\sigma) + d(0, p_j), \frac{3}{2}(\text{OPT}(\sigma) - t_{j-1})\Big\}. \square$$

Open Version on the Line

Recall that we denote by

$$x^{\text{min}} := \min\{0, a_1, \dots, a_n, b_1, \dots, b_n\}$$

the leftmost and by

$$x^{\text{max}} := \max\{0, a_1, \dots, a_n, b_1, \dots, b_n\}$$

the rightmost position that needs to be visited by the server to serve σ. For a schedule S_j, we denote by

$$x_j^{\text{min}} := \min\{\min\{a, b\} \mid (a, b, r) \in \sigma_j\}$$

the leftmost and by

$$x_j^{\text{max}} := \max\{\max\{a, b\} \mid (a, b, r) \in \sigma_j\}$$

the rightmost starting position or destination of the requests σ_j.

Lemma 4.7. *Let S_j with $j \in \{1, \dots, N\}$ be a schedule of a schedule-based algorithm for open online* Dial-a-Ride *on the line. Moreover, let* $\textsc{Opt}(\sigma) = |x^{\text{min}}| + x^{\text{max}} + y$ *for some* $y \geq 0$. *Then, we have*

$$L(t_j, 0, \sigma_j) \leq |\min\{0, x_j^{\text{min}}\}| + \max\{0, x_j^{\text{max}}\} + y.$$

Proof. We need to analyze the amount of time the server needs to serve σ_j starting from position 0 at time t_j. First of all, note that the server does not wait at any point, since all requests of σ_j already have appeared at time t_j. Because of that, the server cannot go to the left of $\min\{0, x_j^{\text{min}}\}$ or to the right of $\max\{0, x_j^{\text{max}}\}$ while staying on an optimum route. Furthermore, we notice that the route Opt takes to serve σ is a valid route to serve σ_j, since $\sigma_j \subseteq \sigma$. However, we can skip every part of the route Opt takes that lies left of $\min\{0, x_j^{\text{min}}\}$ or right of $\max\{0, x_j^{\text{max}}\}$, since no requests of σ_j have a starting or ending position that lies in those intervals. Since all requests already have appeared at time t_j, this does not produce additional waiting time, i.e., we can just delete the parts of the route that lie left of $\min\{0, x_j^{\text{min}}\}$ and right of $\max\{0, x_j^{\text{max}}\}$ and still have a valid route for serving σ_j when starting at time t_j. This shortens the length of the route by at least

$$|x^{\text{min}}| - |\min\{0, x_j^{\text{min}}\}| + x^{\text{max}} - \max\{0, x_j^{\text{max}}\},$$

which gives us

$$\begin{aligned} L(t_j, 0, \sigma_j) &\leq \textsc{Opt}(\sigma) - (|x^{\text{min}}| - |\min\{0, x_j^{\text{min}}\}| + x^{\text{max}} - \max\{0, x_j^{\text{max}}\}) \\ &= |\min\{0, x_j^{\text{min}}\}| + \max\{0, x_j^{\text{max}}\} + y. \end{aligned}$$

□

Since we cannot orient an arbitrary continuous metric space from left to right, i.e., $x^{\min}$ and $x^{\max}$ cannot be defined, Lemma 4.7 cannot be formulated for the general setting. A natural modification of Lemma 4.7 for a continuous metric space X would be to let $\textsc{Opt}(\sigma) = x_X^{\max} + y$ with $y \geq 0$ and

$$x_X^{\max} := \max\{d(0, a_1), \ldots, d(0, a_n), d(0, b_1), \ldots, d(0, b_n)\}$$

being the starting position or destination of the requests σ that is furthest away from the origin. One might expect that for every schedule S_j, we would obtain

$$L(t_j, 0, \sigma_j) \leq x_{X,j}^{\max} + y.$$

with

$$x_{X,j}^{\max} := \max\{\max\{d(0, a), d(0, b_n)\} \mid (a, b; r) \in \sigma_j\}$$

being the starting position or destination of the requests σ_j that is furthest away from the origin. However, a small example shows that this is not true: Let X be the boundary of the unit square $[0, 1]^2$, i.e.,

$$X := \{(0, z) \mid z \in [0, 1]\} \cup \{(z, 1) \mid z \in [0, 1]\} \cup \{(1, z) \mid z \in [0, 1]\} \cup \{(z, 0) \mid z \in [0, 1]\}$$

with euclidean metric d. Furthermore, let $t < t_j < t' < \frac{3}{4}$ and $\sigma' = (s_1, s_2, s_3)$ with

$$\begin{aligned} s_1 &= ((1, 0), (1, 0), t), \\ s_2 &= \left(\left(0, \frac{3}{4}\right), \left(0, \frac{3}{4}\right), t\right), \\ s_3 &= ((1, 1), (1, 1), t'). \end{aligned}$$

We have $\sigma'_j = (s_1, s_2)$ since $t' > t_j$. Thus, we have $x_X^{\max} = d(0, a_3) = 2$ and $x_{X,j}^{\max} = d(0, a_2) = 1$. Since all release times are bounded by $\frac{3}{4}$, every request can be served at arrival without waiting times. $\textsc{Opt}$ performs the walk $(0, 0) \to (1, 0) \to (1, 1) \to (0, \frac{3}{4})$, which implies

$$\textsc{Opt}(\sigma') = \frac{13}{4} = x_X^{\max} + \frac{5}{4},$$

i.e., $y = \frac{5}{4}$. However, the shortest schedule to serve s_1 and s_2 has length

$$L(t_j, 0, (s_1, s_2)) = D\left((0, 0) \to \left(0, \frac{3}{4}\right) \to (1, 0)\right) = \frac{5}{2} > x_{X,j}^{\max} + y.$$

We continue our examination of schedules on the line.

Lemma 4.8. *Let S_j with $j \in \{1, \dots, N\}$ be a schedule of a schedule-based algorithm for open online* Dial-a-Ride *on the line. Moreover, let* $\textsc{Opt}(\sigma) = |x^{\min}| + x^{\max} + y$ *for some* $y \geq 0$*. Then we have*

$$L(t_j, \max\{0, x_j^{\min}\} + \min\{0, x_j^{\max}\}, \sigma_j) \leq x_j^{\max} - x_j^{\min} + y.$$

Proof. First note that the case $\max\{0, x_j^{\min}\} = \min\{0, x_j^{\max}\} = 0$ directly follows from Lemma 4.7. Assume we have $\max\{0, x_j^{\min}\} = x_j^{\min}$. Then all requests of σ_j have starting and ending positions on the right side of the origin, and we have $0 \leq x_j^{\min} \leq x_j^{\max}$, i.e.,

$$\min\{0, x_j^{\max}\} = 0.$$

Similarly, if we have $\min\{0, x_j^{\max}\} = x_j^{\max}$, we have

$$\max\{0, x_j^{\min}\} = 0.$$

Therefore, we have either

$$\max\{0, x_j^{\min}\} + \min\{0, x_j^{\max}\} = x_j^{\min}$$

or

$$\max\{0, x_j^{\min}\} + \min\{0, x_j^{\max}\} = x_j^{\max}.$$

Assume the former is the case. The other case is symmetric. We need to examine $L(t_j, x_j^{\min}, \sigma_j)$, i.e., the length of the optimum offline schedule serving the request sequence σ_j and starting from position $x_j^{\min}$ at time t_j. We note that the server does not wait at any point in time since all requests of σ_j already have appeared at time t_j. Because of that, the server cannot go to the left of $x_j^{\min}$ or to the right of $x_j^{\max}$ while staying on an optimum route. Furthermore, we notice that Opt cannot collect any requests of σ_j before passing $x_j^{\min}$ for the first time, since Opt starts at the origin. Therefore, removing the parts of the walk that Opt performs until it first crosses $x_j^{\min}$, gives us a valid route to serve σ_j, since $\sigma_j \subseteq \sigma$. Additionally, we can skip every part of the route Opt takes to collect requests that lie left of 0 or right of $x_j^{\max}$ since no requests of σ_j have a starting or ending position that lies in those intervals. Again, this does not produce additional waiting time. This shortens the length of the route by at least

$$|x^{\min}| + x_j^{\min} + x^{\max} - x_j^{\max},$$

which gives us

$$L(t_j, x_j^{\min}, \sigma_j) \leq \textsc{Opt}(\sigma) - (|x^{\min}| + x_j^{\min} + x^{\max} - x_j^{\max}) = x_j^{\max} - x_j^{\min} + y. \qquad \square$$

Next, we give an upper bound for the rightmost position that can be reached during a schedule.

Lemma 4.9. *Let S_j with $j \in \{1, \dots, N\}$ be a schedule of a schedule-based algorithm for open online* Dial-a-Ride *on the line. Moreover, let $|x^{\text{min}}| \leq x^{\text{max}}$ and* $\textsc{Opt}(\sigma) = |x^{\text{min}}| + x^{\text{max}} + y$ *for some $y \geq 0$. Then, for every position $p \in \mathbb{R}$ that is visited during the execution of S_j, we have*

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

Proof. First, we notice that the server does not wait at any point since all requests of σ_j already have appeared at time t_j. Because of that, the server cannot go to the left of $\min\{p_j, x_j^{\text{min}}\}$ or to the right of $\max\{p_j, x_j^{\text{max}}\}$ while staying on an optimum route. It suffices to show

$$\max\{p_j, x_j^{\text{max}}\} \leq |p_j| + |p_j - p_{j+1}| + y - |\min\{0, x_j^{\text{min}}\}|. \tag{4.45}$$

We first examine the case $\max\{p_j, x_j^{\text{max}}\} = p_j$: In this case, inequality (4.45) holds if $y \geq |\min\{0, x_j^{\text{min}}\}|$. The inequality $|x^{\text{min}}| \leq x^{\text{max}}$ implies $\textsc{Opt}(\sigma) \geq 2|x^{\text{min}}| + x^{\text{max}}$ and thus $y \geq |x^{\text{min}}|$. Furthermore, we obtain $|x^{\text{min}}| \geq |\min\{0, x_j^{\text{min}}\}|$ since we have $|x^{\text{min}}| \geq 0$ and $|x^{\text{min}}| \geq |x_j^{\text{min}}|$. The latter holds because we have $x^{\text{min}} \leq x_j^{\text{min}}$ if $x_j^{\text{min}} < 0$, i.e., if $\min\{0, x_j^{\text{min}}\} = x_j^{\text{min}}$. This implies $y \geq |\min\{0, x_j^{\text{min}}\}|$, i.e., inequality (4.45) holds.

Thus, we may assume $\max\{p_j, x_j^{\text{max}}\} = x_j^{\text{max}}$ in the following. Similarly to before, if we have $x_j^{\text{max}} \leq 0$, the inequality (4.45) again holds, since the right hand side is always non-negative. We may thus assume $x_j^{\text{max}} > 0$, i.e.,

$$\max\{0, x_j^{\text{max}}\} = x_j^{\text{max}} \tag{4.46}$$

in the following. According to the triangle inequality and Lemma 4.7, we have

$$\begin{aligned} L(t_j, p_j, \sigma_j) &\overset{(4.3)}{\leq} |p_j| + L(t_j, 0, \sigma_j) \\ &\overset{\text{Lem 4.7}}{\leq} |p_j| + |\min\{0, x_j^{\text{min}}\}| + \max\{0, x_j^{\text{max}}\} + y. \\ &\overset{(4.46)}{=} |p_j| + |\min\{0, x_j^{\text{min}}\}| + x_j^{\text{max}} + y. \end{aligned} \tag{4.47}$$

For the sake of contradiction, we assume

$$x_j^{\text{max}} > |p_j| + |p_j - p_{j-1}| + y - |\min\{0, x_j^{\text{min}}\}|. \tag{4.48}$$

Since the server has to visit both extreme positions, i.e., $\max\{p_j, x_j^{\text{max}}\} = x_j^{\text{max}}$ and $\min\{p_j, x_j^{\text{min}}\}$, we have two possible scenarios: the server either visits $\min\{p_j, x_j^{\text{min}}\}$ before

$x_j^{\max}$ or it visits $\min\{p_j, x_j^{\min}\}$ after $x_j^{\max}$. In both cases the schedule S_j ends in position p_{j+1}. In the first case, we have

$$\begin{aligned} L(t_j, p_j, \sigma_j) &\geq |p_j - \min\{p_j, x_j^{\min}\}| + |\min\{p_j, x_j^{\min}\} - x_j^{\max}| + |x_j^{\max} - p_{j+1}| \\ &= p_j - \min\{p_j, x_j^{\min}\} + x_j^{\max} - \min\{p_j, x_j^{\min}\} + x_j^{\max} - p_{j+1} \\ &= p_j - 2\min\{p_j, x_j^{\min}\} + 2x_j^{\max} - p_{j+1} \\ &\overset{(4.48)}{>} p_j - 2\min\{p_j, x_j^{\min}\} + x_j^{\max} + |p_j| + |p_j - p_{j+1}| \\ &\quad +y - |\min\{0, x_j^{\min}\}| - p_{j+1} \\ &\geq x_j^{\max} + |p_j| + y - |\min\{0, x_j^{\min}\}| - 2\min\{p_j, x_j^{\min}\}. \end{aligned} \tag{4.49}$$

In the second case, we obtain the same result

$$\begin{aligned} L(t_j, p_j, \sigma_j) &\geq |p_j - x_j^{\max}| + |x_j^{\max} - \min\{p_j, x_j^{\min}\}| + |\min\{p_j, x_j^{\min}\} - p_{j+1}| \\ &= x_j^{\max} - p_j + x_j^{\max} - \min\{p_j, x_j^{\min}\} + p_{j+1} - \min\{p_j, x_j^{\min}\} \\ &= p_{j+1} + 2x_j^{\max} - 2\min\{p_j, x_j^{\min}\} - p_j \\ &\overset{(4.48)}{>} p_{j+1} + x_j^{\max} + |p_j| + |p_j - p_{j+1}| + y \\ &\quad -|\min\{0, x_j^{\min}\}| - 2\min\{p_j, x_j^{\min}\} - p_j \\ &\geq x_j^{\max} + |p_j| + y - |\min\{0, x_j^{\min}\}| - 2\min\{p_j, x_j^{\min}\}. \end{aligned} \tag{4.50}$$

Now we again consider two cases.

Case 1: $\min\{p_j, x_j^{\min}\} \leq 0$

In this case, we claim that

$$-\min\{p_j, x_j^{\min}\} \geq |\min\{0, x_j^{\min}\}| \tag{4.51}$$

holds. This is clear for $\min\{0, x_j^{\min}\} = 0$ and for $\min\{p_j, x_j^{\min}\} = x_j^{\min}$. In the remaining case, we have $\min\{0, x_j^{\min}\} = x_j^{\min}$ and $\min\{p_j, x_j^{\min}\} = p_j$, i.e., $p_j \leq x_j^{\min} \leq 0$, which implies $-p_j \geq -x_j^{\min} = |x_j^{\min}|$ as desired. This gives us

$$\begin{aligned} L(t_j, p_j, \sigma_j) &\overset{(4.49),(4.50)}{>} x_j^{\max} + |p_j| + y - |\min\{0, x_j^{\min}\}| - 2\min\{p_j, x_j^{\min}\} \\ &\overset{(4.51)}{\geq} x_j^{\max} + |p_j| + y + |\min\{0, x_j^{\min}\}|, \end{aligned}$$

which is a contradiction to inequality (4.47).

Case 2: $\min\{p_j, x_j^{\text{min}}\} > 0$
The inequality $x_j^{\text{max}} \geq x_j^{\text{min}} > 0$ implies

$$\max\{0, x_j^{\text{min}}\} + \min\{0, x_j^{\text{max}}\} = x_j^{\text{min}}. \tag{4.52}$$

Therefore, we can apply Lemma 4.8 and the triangle inequality to obtain

$$\begin{aligned} L(t_j, p_j, \sigma_j) &\overset{(4.3)}{\leq} |p_j - x_j^{\text{min}}| + L(t_j, x_j^{\text{min}}, \sigma_j) \\ &\overset{(4.52),\text{ Lem 4.8}}{\leq} |p_j - x_j^{\text{min}}| + x_j^{\text{max}} - x_j^{\text{min}} + y \\ &= \max\{p_j, x_j^{\text{min}}\} - \min\{p_j, x_j^{\text{min}}\} + x_j^{\text{max}} - x_j^{\text{min}} + y. \end{aligned} \tag{4.53}$$

We have

$$\max\{p_j, x_j^{\text{min}}\} - \min\{p_j, x_j^{\text{min}}\} - x_j^{\text{min}} = p_j - 2\min\{p_j, x_j^{\text{min}}\}. \tag{4.54}$$

This gives us

$$\begin{aligned} L(t_j, p_j, \sigma_j) &\overset{(4.53)}{\leq} \max\{p_j, x_j^{\text{min}}\} - \min\{p_j, x_j^{\text{min}}\} + x_j^{\text{max}} - x_j^{\text{min}} + y \\ &\overset{(4.54)}{=} p_j - 2\min\{p_j, x_j^{\text{min}}\} + x_j^{\text{max}} + y. \end{aligned} \tag{4.55}$$

Finally, we have

$$\begin{aligned} L(t_j, p_j, \sigma_j) &\overset{(4.49),(4.50)}{>} x_j^{\text{max}} + |p_j| + y - |\min\{0, x_j^{\text{min}}\}| - 2\min\{p_j, x_j^{\text{min}}\} \\ &= x_j^{\text{max}} + p_j + y - 2\min\{p_j, x_j^{\text{min}}\}, \end{aligned}$$

which is a contradiction to inequality (4.55). We conclude that (4.48) does not hold, which in turn proves (4.45) in the case that $\max\{p_j, x_j^{\text{max}}\} = x_j^{\text{max}}$ holds. □

We finish this subsection by proving an upper bound for the length of a schedule in terms of the starting and ending position of the schedule.

Proposition 4.10. *Let S_j with $j \in \{1, \ldots, N\}$ be a schedule of a schedule-based algorithm for online* Dial-a-Ride *on the line. Moreover, let $|x^{\text{min}}| \leq x^{\text{max}}$ and* $\text{Opt}(\sigma) = |x^{\text{min}}| + x^{\text{max}} + y$ *for some $y \geq 0$. We have*

$$L(t_j, p_j, \sigma_j) \leq 2|p_j| + |p_j - p_{j+1}| + 2y.$$

Proof. By definition, position $x_j^{\max}$ is visited by the server in schedule S_j. Therefore, we have

$$x_j^{\max} \overset{\text{Lem 4.9}}{\leq} |p_j| + |p_j - p_{j+1}| + y - |\min\{0, x_j^{\min}\}|. \tag{4.56}$$

On the other hand, because of $|x^{\min}| \leq x^{\max}$, we have $\textsc{Opt}(\sigma) \geq 2|x^{\min}| + x^{\max}$, which implies $y \geq |x^{\min}|$. By definition of $x^{\min}$ and $x_j^{\min}$, we have $|x^{\min}| \geq |\min\{0, x_j^{\min}\}|$. This gives us $y \geq |\min\{0, x_j^{\min}\}|$ and

$$0 \leq |p_j| + |p_j - p_{j+1}| + y - |\min\{0, x_j^{\min}\}|. \tag{4.57}$$

To sum it up, we have

$$\max\{0, x_j^{\max}\} \overset{(4.56),(4.57)}{\leq} |p_j| + |p_j - p_{j+1}| + y - |\min\{0, x_j^{\min}\}|. \tag{4.58}$$

Using the triangle inequality and the inequality above, we obtain

$$\begin{aligned} L(t_j, p_j, \sigma_j) &\overset{(4.3)}{\leq} |p_j| + L(t_j, 0, \sigma_j) \\ &\overset{\text{Lem 4.7}}{\leq} |p_j| + |\min\{0, x_j^{\min}\}| + \max\{0, x_j^{\max}\} + y \\ &\overset{(4.58)}{\leq} 2|p_j| + |p_j - p_{j+1}| + 2y. \end{aligned}$$ □

Equipped with the results of this section, we are able to analyze schedule-based algorithms more easily. In the following, we will analyze three different schedule-based algorithms. We start with the simplest schedule-based algorithm Ignore.

4.3 Algorithm Ignore

The simplest waiting strategy is to never wait if there are unserved requests. The algorithm that utilizing this strategy is called Ignore (see Algorithm 2) and was published by Ascheuer et al. in [5]. However, a similar strategy already was examined in [43] for the machine scheduling problem.

Formally, Ignore is a schedule-based algorithm for online Dial-a-Ride and online TSP utilizing the waiting function

$$\text{wait}_{\text{IG}}(t) := \begin{cases} \text{false}, & \text{if } R_t \neq \emptyset, \\ \text{true}, & \text{otherwise.} \end{cases}$$

Algorithm 2 IGNORE

repeat
  if $R_t \neq \emptyset$ **then**
    Start optimal offline schedule serving R_t starting from the current position
  else
    wait

In [5, Theorem 4], Ascheuer et al. showed that the IGNORE algorithm is $\frac{5}{2}$-competitive for closed online DIAL-A-RIDE. Later, Krumke, one of the authors of [5], analyzed IGNORE for the open version of online DIAL-A-RIDE in his PhD thesis. He showed that the algorithm is 4-competitive [32, Theorem 2.29]. For a summary of results concerning the competitive ratio of IGNORE, excluding the results of this thesis, see Table 2.3.

We will complement the upper bound for IGNORE for closed online DIAL-A-RIDE with a lower bound of $\frac{5}{2}$ using only TSP requests on the line, i.e. the closed version of IGNORE has a competitive ratio of exactly $\frac{5}{2}$ for both, DIAL-A-RIDE and TSP on general metric spaces as well as on the real line. For open online DIAL-A-RIDE, we complement the upper bound of 4 with a lower bound of 4 on the real line. However, this lower bound construction utilizes transportation requests and is not valid for open online TSP. For open online TSP we instead provide an improved upper bound of $\frac{7}{2}$, which we will be complemented with a lower bound of 3 on the real line. We start with the closed version.

Theorem 4.11. *The competitive ratio of* IGNORE *for closed online* DIAL-A-RIDE *and online* TSP *is* $\frac{5}{2}$.

Proof. It was shown in [5, Theorem 4] that IGNORE is $\frac{5}{2}$-competitive for closed online DIAL-A-RIDE on arbitrary metric spaces and therefore in particular for online TSP and on the real line. It remains to show that for every sufficiently small $\varepsilon > 0$ there is a sequence of requests σ_1^{IG} containing no transportation requests such that

$$\textsc{Ignore}(\sigma_1^{\text{IG}}) \geq \left(\frac{5}{2} - \varepsilon\right)\textsc{Opt}(\sigma_1^{\text{IG}}).$$

Let $\varepsilon > 0$ with $\varepsilon < \frac{1}{2}$. We consider the sequence of requests σ_1^{IG} consisting of

$$\begin{aligned} s_1 &= \left(\frac{1}{2} - \varepsilon, \frac{1}{2} - \varepsilon; 0\right), \\ s_2 &= (1, 1; \varepsilon), \\ s_3 &= (1, 1; 1). \end{aligned}$$

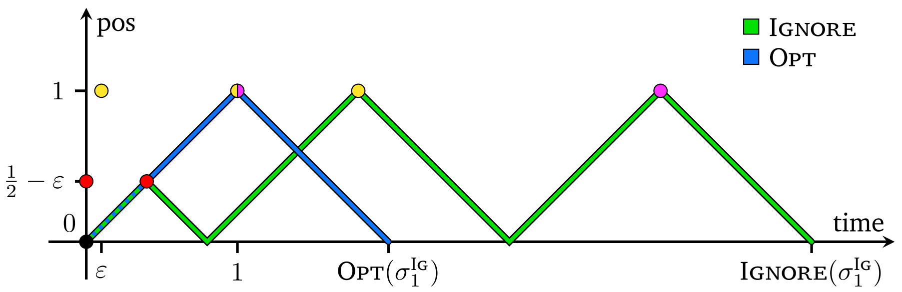

Figure 4.10: IGNORE's and OPT's walk serving σ_1^{IG} with $\varepsilon = 0.1$. Request s_1 is red, request s_2 is yellow and request s_3 is violet.

The walk IGNORE performs in a position-time diagram is illustrated in green in Figure 4.10. IGNORE first serves request s_1 in schedule S_1 and returns to the origin at time $t_2 = 1 - 2\varepsilon$. Note that s_3 is not yet released at time t_2. Thus, IGNORE serves only s_2 in schedule S_2 and returns to the origin at time $t_3 = 3 - \varepsilon$. The request s_3 is served in the final schedule S_3. To sum it up, we have

$$\text{IGNORE}(\sigma_1^{\text{IG}}) = 5 - 2\varepsilon.$$

OPT on the other hand serves all three requests on its way from the origin to position 1 and returns to the origin, resulting in

$$\text{OPT}(\sigma_1^{\text{IG}}) = 2.$$

The walk of OPT is illustrated in blue in Figure 4.10. To sum it up, we have

$$\text{IGNORE}(\sigma_1^{\text{IG}}) = \left(\frac{5}{2} - \varepsilon\right)\text{OPT}(\sigma_1^{\text{IG}}). \qquad \square$$

Next, we complement the upper bound of 4 [32, Theorem 2.29] for open online DIAL-A-RIDE with a matching lower bound. We published this lower bound also in [10].

Theorem 4.12. *The competitive ratio of* IGNORE *for open online* DIAL-A-RIDE *is* 4*.*

Proof. It was shown in [32, Theorem 2.29] that 4 is an upper bound for the competitive ratio of IGNORE for online DIAL-A-RIDE on arbitrary metric spaces and therefore in particular for the real line. It remains to show that for every sufficiently small $\varepsilon > 0$ there is a sequence of requests σ_2^{IG} such that

$$\text{IGNORE}(\sigma_2^{\text{IG}}) \geq (4 - \varepsilon)\text{OPT}(\sigma_2^{\text{IG}}).$$

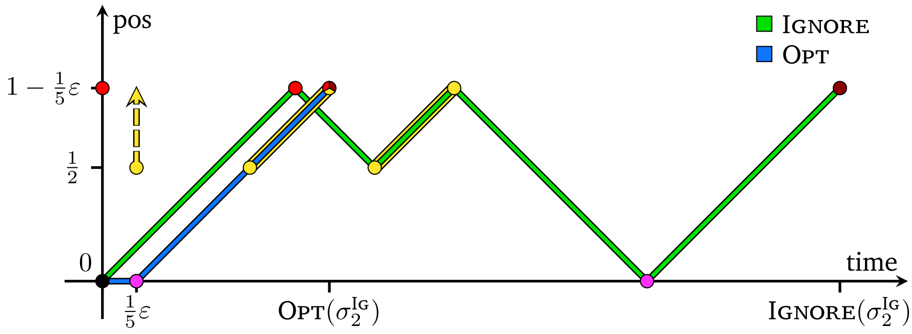

Figure 4.11: IGNORE's and OPT's walk serving σ_2^{IG} with $\varepsilon = 0.75$. Request s_1 is red ●, $s_2^{(1)}$ is yellow ●, $s_2^{(1)}$ is violet ● and s_3 is brown ●.

Let $\varepsilon > 0$ with $\varepsilon < \frac{5}{2}$. We consider the sequence of requests σ_2^{IG} consisting of

$$\begin{aligned} s_1 &= \left(1-\frac{1}{5}\varepsilon, 1-\frac{1}{5}\varepsilon; 0\right), \\ s_2^{(1)} &= \left(\frac{1}{2}, 1-\frac{1}{5}\varepsilon; \frac{1}{5}\varepsilon\right), \\ s_2^{(2)} &= \left(0, 0; \frac{1}{5}\varepsilon\right), \\ s_3 &= \left(1-\frac{1}{5}\varepsilon, 1-\frac{1}{5}\varepsilon; 1\right). \end{aligned}$$

The walk IGNORE performs in a position-time diagram is illustrated in green in Figure 4.11. IGNORE serves request s_1 in schedule S_1 and finishes at time $t_2 = 1 - \frac{1}{5}\varepsilon$ at position $p_2 = 1 - \frac{1}{5}\varepsilon$. Note that request s_3 is not yet released at time t_2. Thus, in schedule S_2 only the requests $s_2^{(1)}$ and $s_2^{(2)}$ are served. Note that serving $s_2^{(1)}$ before $s_2^{(2)}$ takes time $2 - \frac{3}{5}\varepsilon$, while serving $s_2^{(2)}$ first takes time $2 - \frac{2}{5}\varepsilon$. Therefore, IGNORE serves $s_2^{(1)}$ first and schedule S_2 ends at time $t_3 = 3 - \frac{4}{5}\varepsilon$ at position $p_3 = 0$. The final schedule S_3 has length $1 - \frac{1}{5}\varepsilon$ and serves s_3. To sum it up, we have

$$\text{IGNORE}(\sigma_2^{\text{IG}}) = 4 - \varepsilon.$$

OPT on the other hand waits until time $\frac{1}{5}\varepsilon$ at the origin for the request $s_2^{(2)}$ and then collects and delivers the remaining requests on its way to position $1 - \frac{1}{5}\varepsilon$, resulting in

$$\text{OPT}(\sigma_2^{\text{IG}}) = 1.$$

The walk of Opt is illustrated in blue in Figure 4.11. To sum it up, we have

$$\text{Ignore}(\sigma_2^{\text{Ig}}) = (4-\varepsilon)\text{Opt}(\sigma_2^{\text{Ig}}). \qquad \square$$

In case of open online TSP, we provide a better upper bound than for the competitive ratio of Ignore.

Theorem 4.13. Ignore *for open online* TSP *is* $\frac{7}{2}$*-competitive.*

Proof. Let $s_n = (a_n; r_n) \in \sigma$ be the request that is released last. If Ignore is not on a schedule at time r_n, we have

$$\text{Ignore}(\sigma) \overset{(4.1)}{=} t_N + L(t_N, p_N, \sigma_N) = r_n + L(r_n, p_N, \sigma_N). \tag{4.59}$$

Since Opt has to serve s_n, we have Opt $\geq r_n$. By Lemma 4.6, we get

$$\begin{aligned} \text{Ignore}(\sigma) &\overset{(4.59)}{=} r_n + L(r_n, p_N, \sigma N) \\ &\leq \text{Opt}(\sigma) + L(r_n, p_N, \sigma_N) \\ &\overset{\text{Lem 4.6}}{\leq} \frac{5}{2}\text{Opt}(\sigma) - \frac{3}{2}t_{N-1} \\ &\leq \frac{5}{2}\text{Opt}(\sigma). \end{aligned}$$

Now assume Ignore is busy executing a schedule at time r_n. Then, we have

$$\begin{aligned} \text{Ignore}(\sigma) &\overset{(4.1)}{=} t_N + L(t_N, p_N, \sigma_N) \\ &= t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + L(t_N, p_N, \sigma_N). \end{aligned} \tag{4.60}$$

Let $s_N^{\text{Opt}} = (a_N^{\text{Opt}}; r_N^{\text{Opt}})$ be the first request of σ_N that is served by Opt. We have

$$L(t_N, p_N, \sigma_N) \overset{(4.3)}{\leq} d(a_j^{\text{Opt}}, p_N) + L(t_N, a_N^{\text{Opt}}, \sigma_N) \tag{4.61}$$

by the triangle inequality. Since Opt cannot serve s_N^{Opt} before time r_j^{Opt}, we have

$$\text{Opt}(\sigma) \geq r_j^{\text{Opt}} + L(t_N, a_N^{\text{Opt}}, \sigma_N). \tag{4.62}$$

Combining the inequalities (4.61) and (4.62), we obtain

$$\text{Ignore}(\sigma) \overset{(4.60)}{=} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + L(t_N, p_N, \sigma_N)$$

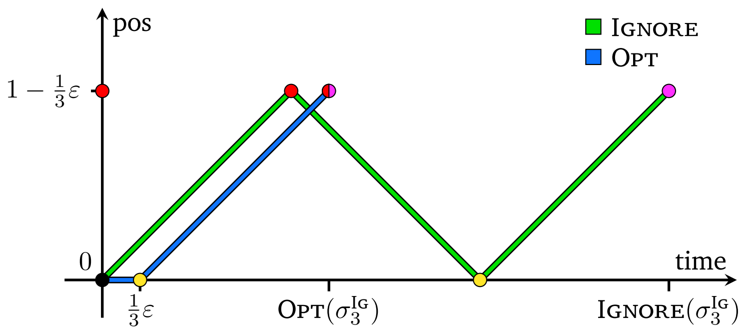

Figure 4.12: IGNORE's and OPT's walk serving σ_3^{IG} with $\varepsilon = 0.75$. Request s_1 is red ●, request s_2 is yellow ● and request s_3 is violet ●.

$$
\begin{aligned}
&\overset{(4.61)}{\leq} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + d(a_j^{\textsc{Opt}}, p_N) + L(t_N, a_N^{\textsc{Opt}}, \sigma_N) \\
&\overset{(4.62)}{\leq} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + d(a_j^{\textsc{Opt}}, p_N) + \textsc{Opt}(\sigma) - r_j^{\textsc{Opt}} \\
&\overset{t_{N-1} \leq r_j^{\textsc{Opt}}}{\leq} L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + d(a_j^{\textsc{Opt}}, p_N) + \textsc{Opt}(\sigma).
\end{aligned} \tag{4.63}
$$

Since OPT has to visit $a_j^{\textsc{Opt}}$ and p_N, we have $\textsc{Opt}(\sigma) \geq d(a_j^{\textsc{Opt}}, p_N)$ and Lemma 4.6 implies

$$
L(t_{N-1}, p_{N-1}, \sigma_{N-1}) \overset{\text{Lem 4.6}}{\leq} \frac{3}{2}(\textsc{Opt}(\sigma) - t_{N-2}) \leq \frac{3}{2}\textsc{Opt}(\sigma). \tag{4.64}
$$

Finally, combining the inequalities above gives

$$
\textsc{Ignore}(\sigma) \overset{(4.63)}{\leq} L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + d(a_j^{\textsc{Opt}}, p_N) + \textsc{Opt}(\sigma) \overset{(4.64)}{\leq} \frac{7}{2}\textsc{Opt}(\sigma). \qquad \square
$$

We complement this upper bound with a lower bound of 3.

Theorem 4.14. *For every sufficiently small $\varepsilon > 0$ there is a request sequence σ_3^{IG} only containing* TSP *requests such that*

$$
\textsc{Ignore}(\sigma_3^{\mathrm{IG}}) = (3 - \varepsilon)\textsc{Opt}(\sigma_3^{\mathrm{IG}}).
$$

Proof. Let $\varepsilon > 0$ with $\varepsilon < \frac{3}{2}$. We consider the sequence of requests σ_3^{IG} consisting of

$$
s_1 = \left(1 - \frac{1}{3}\varepsilon; 0\right),
$$

$$s_2 = \left(0; \frac{1}{3}\varepsilon\right),$$
$$s_3 = \left(1 - \frac{1}{3}\varepsilon; 1\right).$$

The walk IGNORE performs in a position-time diagram is illustrated in green in Figure 4.12. IGNORE serves request s_1 in schedule S_1 at position $p_2 = 1 - \frac{1}{3}\varepsilon$ at time $t_2 = 1 - \frac{1}{3}\varepsilon$. Note that request s_3 is not yet released at time t_2. In schedule S_2 only request s_2 is served. Schedule S_2 finishes at position $p_3 = 0$ at time $t_3 = 2 - \frac{2}{3}\varepsilon$. The final schedule S_3 has length $1 - \frac{1}{3}\varepsilon$ and serves s_3. To sum it up, we have

$$\text{IGNORE}(\sigma_3^{\text{IG}}) = 3 - \varepsilon.$$

OPT on the other hand waits until time $\frac{1}{3}\varepsilon$ at the origin for the request s_2 and then serves the remaining requests on its way to position $1 - \frac{1}{3}\varepsilon$, resulting in

$$\text{OPT}(\sigma_3^{\text{IG}}) = 1.$$

The walk of OPT is illustrated in blue in Figure 4.12. To sum it up, we have

$$\text{IGNORE}(\sigma_3^{\text{IG}}) = (3 - \varepsilon)\text{OPT}(\sigma_3^{\text{IG}}). \qquad \square$$

The results of this section are summarized in Table 2.6. We produced tight bounds for closed and open online DIAL-A-RIDE and closed online TSP. Only for open online TSP a gap remains. Note that we have the same bounds for IGNORE on the line as in the general setting. Furthermore, in the case of online DIAL-A-RIDE, we have the same bounds for all choices of the capacity c. Thus, we have shown that restricting the metric space to the real line has no impact on IGNORE's competitive ratio for closed and open online DIAL-A-RIDE and for closed online TSP. Moreover, choosing a specific capacity of the server has no impact on IGNORE's competitive ratio either.

However, there is a significant gap between IGNORE's competitive ratios and the lower bounds for the competitive ratios of schedule-based algorithms presented in Section 4.1. This indicates that IGNORE is a rather weak schedule-based algorithm. And indeed our analysis exposes a critical weakness of IGNORE: IGNORE is very easily lured away from the origin even though it would have been smarter in many cases to wait before executing a schedule. Ascheuer et al. also had this insight. To address this issue they proposed the SMARTSTART algorithm in [5], a schedule-based algorithm with a waiting routine dependent on the length of the upcoming schedule. This algorithm eliminates the critical weakness of IGNORE. We present a detailed examination of SMARTSTART in the next chapter.

5 Algorithm SMARTSTART

In comparison to IGNORE, the algorithm SMARTSTART [5] uses the given information about the subsequence of unserved requests R_t at time t. While IGNORE bases its waiting function only on whether the subsequence R_t is empty, SMARTSTART compares the current time with the time needed to serve R_t from its current position.

Algorithm 3 SMARTSTART

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

The algorithm SMARTSTART is given in Algorithm 3. Essentially, at time t, SMARTSTART waits before starting an optimal schedule to serve all available requests at time

$$\min\left\{t' \in \mathbb{R}_{\geq 0} : t' \geq t \wedge t' \geq \frac{L(t', p, R_{t'})}{\Theta - 1}\right\}, \tag{5.1}$$

where p is the current position of the server and $\Theta > 1$ is a parameter of the algorithm that scales the waiting time. Formally, SMARTSTART is a schedule-based algorithm with waiting function

$$\mathrm{wait}_{\mathrm{SM}}(t) := \begin{cases} \text{false}, & \text{if } R_t \neq \emptyset \text{ and } t \geq \frac{L(t,p,R_t)}{\Theta - 1}, \\ \text{true}, & \text{otherwise.} \end{cases}$$

The SMARTSTART algorithm is of particular importance since it achieves the best possible competitive ratio of 2 for the closed online DIAL-A-RIDE on arbitrary continuous metric spaces [5, Thm 6] [8, Thm 4.2], and the best known upper bound of roughly 3.4142 for the competitive ratio of the open variant [32, Thm 2.30]. SMARTSTART is also a best possible schedule-based algorithm for closed online DIAL-A-RIDE on the line according to Theorem 4.3. In this section, we provide the exact competitive ratio of SMARTSTART for open online DIAL-A-RIDE and online TSP on the line. Furthermore, we provide improved upper bounds for the competitive ratio of SMARTSTART for open online DIAL-A-RIDE and online TSP in the general setting.

Regarding open online DIAL-A-RIDE, we show that SMARTSTART attains a competitive ratio of $\rho_{\text{SM}}^{\text{D},\mathbb{R}} \approx 2.9377$ on the line for parameter value $\Theta_{\text{SM}}^{\text{D},\mathbb{R}} \approx 2.0526$ (Thm 5.26) and is $\rho_{\text{SM}}^{\text{D},X}$-competitive with $\rho_{\text{SM}}^{\text{D},X} = 3$ in the general setting for parameter value $\Theta_{\text{SM}}^{\text{D},X} = 2$ (Thm 5.43). For open online TSP, we show that SMARTSTART achieves a competitive ratio of $\rho_{\text{SM}}^{\text{T},\mathbb{R}} \approx 2.7604$ on the line for parameter value $\Theta_{\text{SM}}^{\text{T},\mathbb{R}} \approx 1.8607$ (Thm 5.41) and is $\rho_{\text{SM}}^{\text{T},X}$-competitive with $\rho_{\text{SM}}^{\text{T},X} \approx 2.8229$ in the general setting for parameter value $\Theta_{\text{SM}}^{\text{T},X} \approx 1.8229$ (Thm 5.44).

We published the results of the first two sections also in [10]. To show the upper bounds for the competitive ratio of open online DIAL-A-RIDE on the line, we derive two separate upper bounds depending on Θ: an upper bound for the case that SMARTSTART postpones starting its final schedule and an upper bound for the case that SMARTSTART does not postpone its final schedule (see Section 5.1). We complement the upper bounds with matching lower bounds in Section 5.2. For online TSP on the line we show a slightly better upper bound for the competitive ratio for the case that the final schedule is postponed, which improves the general upper bound in comparison to the DIAL-A-RIDE version (see Section 5.3). In the same section, we match this slightly improved upper bound with a matching lower bound. For arbitrary continuous metric spaces, we show slightly weaker upper bounds for the competitive ratio of SMARTSTART for open online DIAL-A-RIDE and online TSP in the case that the final schedule is not postponed. This yields slightly weaker general upper bounds than on the real line (see Section 5.4).

5.1 Upper Bound for Open Online DIAL-A-RIDE on the Line

In this section, we give an upper bound for the completion time of SMARTSTART in comparison the optimum offline time to OPT(σ). To do this, we consider two cases, depending on whether or not SMARTSTART postpones the execution of the final schedule S_N. If SMARTSTART postpones the execution of S_N (i.e., it waits even though there are unserved requests), the starting time of schedule S_N is given by

$$t_N = \frac{1}{\Theta - 1} L(t_N, p_N, \sigma_N). \tag{5.2}$$

If SMARTSTART does not postpone the final schedule, we have

$$t_N = t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) \tag{5.3}$$

if the final schedule S_N is executed directly after the second to final schedule and

$$t_N = r_n \tag{5.4}$$

if there are no unserved requests at the point of time the execution of S_{N-1} is finished and the last requests are released at time $r_n > \frac{1}{\Theta-1} L(t_N, p_N, \sigma_N)$.

First, we examine the case that the final schedule is postponed. We start by giving a lower bound for the starting time of a schedule.

Lemma 5.1. *Algorithm* SMARTSTART *for open online* DIAL-A-RIDE *does not start schedule S_j earlier than time $\frac{1}{\Theta} d(0, p_{j+1})$, i.e., we have $t_j \geq \frac{1}{\Theta} d(0, p_{j+1})$.*

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

$$L(t_j, p_j, \sigma_j) \geq d(p_j, p_{j+1}).$$

Note however that SMARTSTART needs at least time $d(p_j, 0)$ to reach p_j. Therefore, we have

$$\begin{aligned} t_j &\overset{(5.1)}{\geq} \min\left\{ t \in \mathbb{R}_{\geq 0} : t \geq d(0, p_j) \wedge t \geq \frac{d(p_j, p_{j+1})}{\Theta - 1} \right\} \\ &= \max\left\{ d(0, p_j), \frac{d(p_j, p_{j+1})}{\Theta - 1} \right\}. \end{aligned} \tag{5.5}$$

It remains to show

$$\max\left\{ d(0, p_j), \frac{d(p_j, p_{j+1})}{\Theta - 1} \right\} \geq \frac{d(0, p_{j+1})}{\Theta}.$$

For $d(0, p_j) \geq \frac{d(0, p_{j+1})}{\Theta}$ we trivially have

$$\max\left\{ d(0, p_j), \frac{d(p_j, p_{j+1})}{\Theta - 1} \right\} \geq d(0, p_j) \geq \frac{d(0, p_{j+1})}{\Theta}. \tag{5.6}$$

For $d(0, p_j) < \frac{d(0, p_{j+1})}{\Theta}$, the triangle inequality implies

$$\max\left\{ d(0, p_j), \frac{d(p_j, p_{j+1})}{\Theta - 1} \right\} \quad \geq \quad \frac{d(p_j, p_{j+1})}{\Theta - 1}$$

$$\begin{aligned}
&\geq && \frac{d(0,p_{j+1})}{\Theta-1} - \frac{d(0,p_j)}{\Theta-1} \\
&\overset{d(0,p_j) < \frac{d(0,p_{j+1})}{\Theta}}{>} && \frac{d(0,p_{j+1})}{\Theta-1} - \frac{d(0,p_{j+1})}{\Theta(\Theta-1)} \\
&= && \frac{d(0,p_{j+1})}{\Theta}. \qquad (5.7)
\end{aligned}$$

To sum it up, we have

$$t_j \overset{(5.5)}{\geq} \max\left\{d(0,p_j), \frac{d(p_j,p_{j+1})}{\Theta-1}\right\} \overset{(5.6),(5.7)}{\geq} \frac{d(0,p_{j+1})}{\Theta}. \qquad \square$$

Using Lemmas 5.1 and 4.5, we can compute an upper bound for the length of SMARTSTART's schedules that is only dependent on the scaling parameter Θ.

Lemma 5.2. *For every schedule S_j of* SMARTSTART *for open online* DIAL-A-RIDE*, we have*

$$L(t_j,p_j,\sigma_j) \leq \left(1 + \frac{\Theta}{\Theta+2}\right)\textsc{Opt}(\sigma).$$

Proof. By Lemma 4.5 and Lemma 5.1 we have

$$\begin{aligned}
L(t_j,p_j,\sigma_j) &\overset{\text{Lem 4.5}}{\leq} \min\{\textsc{Opt}(\sigma) + d(p_j,0), 2(\textsc{Opt}(\sigma) - t_{j-1})\} \\
&\overset{\text{Lem 5.1}}{\leq} \min\left\{\textsc{Opt}(\sigma) + d(p_j,0), 2\left(\textsc{Opt}(\sigma) - \frac{1}{\Theta}d(p_j,0)\right)\right\} \\
&\leq \left(1 + \frac{\Theta}{\Theta+2}\right)\textsc{Opt}(\sigma)
\end{aligned}$$

since the minimum above is largest if the two terms are equal, which is the case for $d(p_j,0) = \frac{\Theta}{\Theta+2}\textsc{Opt}(\sigma)$. $\square$

We are now ready to present an upper bound for the competitive ratio of SMARTSTART for open online DIAL-A-RIDE in the case that the final schedule is postponed.

Proposition 5.3. *In the case that* SMARTSTART *for open online* DIAL-A-RIDE *postpones executing S_N, we have*

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

Proof. Assume SMARTSTART postpones the final schedule. Then we have

$$\textsc{SmartStart}(\sigma) \stackrel{(4.1)}{=} t_N + L(t_N, p_N, \sigma_N) \stackrel{(5.2)}{=} \frac{\Theta}{\Theta - 1} L(t_N, p_N, \sigma_N). \tag{5.8}$$

Lemma 5.2 thus yields the claimed bound:

$$\begin{aligned} \textsc{SmartStart}(\sigma) &\stackrel{(5.8)}{=} \frac{\Theta}{\Theta - 1} L(t_N, p_N, \sigma_N) \\ &\stackrel{\text{Lem 5.2}}{\leq} \frac{\Theta}{\Theta - 1}\left(1 + \frac{\Theta}{\Theta + 2}\right)\textsc{Opt}(\sigma) \\ &= \frac{2\Theta^2 + 2\Theta}{\Theta^2 + \Theta - 2}\textsc{Opt}(\sigma). \end{aligned}$$ □

Note that the upper bound presented in Proposition 5.3 is valid in the general setting and thus also on the real line. It remains to examine the case where the algorithm SMARTSTART does not postpone the final schedule.

Proposition 5.4. *If* SMARTSTART *for open online* DIAL-A-RIDE *on the line does not postpone executing S_N, we have*

$$\frac{\textsc{SmartStart}(\sigma)}{\textsc{Opt}(\sigma)} \leq \frac{3\Theta^2 + 5\Theta + 4}{3\Theta + 3} =: f_2^{\text{up}}(\Theta).$$

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

Let the latter be the case, then the final schedule is started at the release time r_n of the last request. Since OPT also has to serve the last request, we have $\textsc{Opt}(\sigma) \geq r_n$ and since the execution of the final schedule is not postponed, we have $r_n > \frac{1}{\Theta - 1} L(t_N, p_N, \sigma_N)$, i.e.,

$$L(t_N, p_N, \sigma_N) < (\Theta - 1)\textsc{Opt}(\sigma). \tag{5.9}$$

In total we have

$$\begin{aligned} \textsc{SmartStart}(\sigma) &\stackrel{(4.1)}{=} t_N + L(t_N, p_N, \sigma_N) \\ &\stackrel{(5.4)}{=} r_n + L(t_N, p_N, \sigma_N) \\ &\stackrel{(5.9)}{<} \Theta\textsc{Opt}(\sigma) \\ &< \frac{3\Theta^2 + 5\Theta + 4}{3\Theta + 3}\textsc{Opt}(\sigma). \end{aligned}$$

Now let the final schedule be started immediately after the second to final schedule. Without loss of generality, we assume $|x^{\min}| \leq x^{\max}$ throughout the rest of the proof. The other case follows by symmetry. Let s_N^{OPT} be the first request of σ_N that is served by OPT and let a_N^{OPT} be its starting position and r_N^{OPT} be its release time. We have

$$\begin{aligned}
\text{SMARTSTART}(\sigma) &\stackrel{(4.1)}{=} t_N + L(t_N, p_N, \sigma_N) \\
&\stackrel{(5.3)}{=} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + L(t_N, p_N, \sigma_N) \\
&\stackrel{(4.2)}{\leq} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + L(r_N^{\text{OPT}}, p_N, \sigma_N). \qquad (5.10)
\end{aligned}$$

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

$$\text{OPT}(\sigma) \geq r_N^{\text{OPT}} + L(r_N^{\text{OPT}}, a_N^{\text{OPT}}, \sigma_N). \qquad (5.11)$$

Furthermore, we have

$$r_N^{\text{OPT}} > t_{N-1} \qquad (5.12)$$

since otherwise $s_N^{\text{OPT}} \in \sigma_{N-1}$ would hold and

$$t_{N-1} \stackrel{(5.1)}{\geq} \frac{1}{\Theta - 1} L(t_{N-1}, p_{N-1}, \sigma_{N-1}). \qquad (5.13)$$

by definition of SMARTSTART. This gives us

$$\begin{aligned}
\text{SMARTSTART}(\sigma) &\stackrel{(5.10)}{\leq} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + L(r_N^{\text{OPT}}, p_N, \sigma_N) \\
&\stackrel{(4.3)}{\leq} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + |a_N^{\text{OPT}} - p_N| + L(r_N^{\text{OPT}}, a_N^{\text{OPT}}, \sigma_N) \\
&\stackrel{(5.11)}{\leq} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + |a_N^{\text{OPT}} - p_N| + \text{OPT}(\sigma) - r_N^{\text{OPT}} \\
&\stackrel{(5.12)}{<} L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + |a_N^{\text{OPT}} - p_N| + \text{OPT}(\sigma). \qquad (5.14)
\end{aligned}$$

We denote by s_{N-1}^{SM} the last request that is delivered during schedule S_{N-1} by SMARTSTART. Note that the destination of s_{N-1}^{SM} is p_N. We consider two cases.

Case 1: OPT collects s_N^{OPT} before delivering the request s_{N-1}^{SM}

Obviously OPT cannot collect the request s_N^{OPT} before its release time r_N^{OPT}. Furthermore, since OPT still has to go to position p_N for delivering request s_{N-1}^{SM} after collecting s_N^{OPT}, we have

$$\text{OPT}(\sigma) \geq r_N^{\text{OPT}} + |a_N^{\text{OPT}} - p_N| \stackrel{(5.12)}{>} t_{N-1} + |a_N^{\text{OPT}} - p_N|. \qquad (5.15)$$

The inequality above gives us

$$\begin{aligned}
\textsc{SmartStart}(\sigma) &\overset{(5.14)}{<} L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + |a_N^{\textsc{Opt}} - p_N| + \textsc{Opt}(\sigma) \\
&\overset{(5.15)}{<} L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + 2\textsc{Opt}(\sigma) - t_{N-1} \\
&\overset{(5.13)}{\leq} \frac{\Theta - 2}{\Theta - 1} L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + 2\textsc{Opt}(\sigma) \\
&\overset{\text{Lem 5.2}}{\leq} \frac{\Theta - 2}{\Theta - 1}\left(1 + \frac{\Theta}{\Theta + 2}\right)\textsc{Opt}(\sigma) + 2\textsc{Opt}(\sigma) \\
&= \frac{4\Theta^2 - 8}{\Theta^2 + \Theta - 2}\textsc{Opt}(\sigma) \\
&\overset{\Theta \geq 1}{<} \frac{3\Theta^2 + 5\Theta + 4}{3\Theta + 3}\textsc{Opt}(\sigma).
\end{aligned}$$

Case 2: OPT delivers $s_{N-1}^{\textsc{Sm}}$ before collecting the request $s_N^{\textsc{Opt}}$

In this case we have

$$\begin{aligned}
\textsc{SmartStart}(\sigma) &\overset{(5.14)}{<} L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + |a_N^{\textsc{Opt}} - p_N| + \textsc{Opt}(\sigma) \\
&\overset{(5.13)}{\leq} (\Theta - 1)t_{N-1} + |a_N^{\textsc{Opt}} - p_N| + \textsc{Opt}(\sigma) \\
&\overset{t_{N-1} < \textsc{Opt}(\sigma)}{<} \Theta\textsc{Opt}(\sigma) + |a_N^{\textsc{Opt}} - p_N|.
\end{aligned}$$

This means the claim is shown if we have

$$|p_N - a_N^{\textsc{Opt}}| < \frac{2\Theta + 4}{3\Theta + 3}\textsc{Opt}(\sigma).$$

Therefore, we may assume in the following that

$$|p_N - a_N^{\textsc{Opt}}| \geq \frac{2\Theta + 4}{3\Theta + 3}\textsc{Opt}(\sigma). \tag{5.16}$$

Let $\textsc{Opt}(\sigma) = |x^{\min}| + x^{\max} + y$ for some $y \geq 0$. By definition of $x^{\min}$ and $x^{\max}$ we have

$$|p_N - a_N^{\textsc{Opt}}| + y \leq \textsc{Opt}(\sigma). \tag{5.17}$$

Since by assumption OPT delivers $s_{N-1}^{\textsc{Sm}}$ to position p_N before collecting $s_N^{\textsc{Opt}}$ at position $a_N^{\textsc{Opt}}$, we have

$$|p_N - a_N^{\textsc{Opt}}| + |p_N| \leq \textsc{Opt}(\sigma), \tag{5.18}$$

and since s^{SM}_{N-1} appears after time t_{N-2}, we also have

$$|p_N - a^{\text{OPT}}_N| + t_{N-2} < \text{OPT}(\sigma). \tag{5.19}$$

To sum it up, we may assume that

$$\max\{y, |p_N|, t_{N-2}\} \overset{(5.16),(5.17),(5.18),(5.19)}{\leq} \frac{\Theta - 1}{3\Theta + 3}\text{OPT}(\sigma) \tag{5.20}$$

holds. We compute

$$\begin{aligned}
\text{SMARTSTART}(\sigma) &\overset{(5.14)}{<} L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + |p_N - a^{\text{OPT}}_N| + \text{OPT}(\sigma) \\
&\overset{\text{Prop 4.10}}{\leq} 2|p_{N-1}| + |p_{N-1} - p_N| + 2y + |p_N - a^{\text{OPT}}_N| + \text{OPT}(\sigma) \\
&\leq 3|p_{N-1}| + |p_N| + 2y + |p_N - a^{\text{OPT}}_N| + \text{OPT}(\sigma) \\
&\overset{(5.18)}{\leq} 3|p_{N-1}| + 2y + 2\text{OPT}(\sigma) \\
&\overset{\text{Lem 5.1}}{\leq} 3\Theta t_{N-2} + 2y + 2\text{OPT}(\sigma) \\
&\overset{(5.20)}{\leq} 3\Theta\frac{\Theta - 1}{3\Theta + 3} + 2\frac{\Theta - 1}{3\Theta + 3} + 2\text{OPT}(\sigma) \\
&= \frac{3\Theta^2 + 5\Theta + 4}{3\Theta + 3}\text{OPT}(\sigma).
\end{aligned}$$

□

We combine the results of Proposition 5.3 and Proposition 5.4 to obtain a general upper bound for the competitive ratio of SMARTSTART for online DIAL-A-RIDE on the line.

Theorem 5.5. *The function* $\max\{f^{\text{up}}_1, f^{\text{up}}_2\}$ *gives an upper bound for the competitive ratio of* SMARTSTART *for open online* DIAL-A-RIDE *on the line for all* $\Theta > 1$*. Let* $\Theta^{\text{D},\mathbb{R}}_{\text{SM}} \approx 2.0526$ *be the unique solution of the equation* $f^{\text{up}}_1(\Theta) = f^{\text{up}}_2(\Theta)$*, i.e., of*

$$\frac{2\Theta^2 + 2\Theta}{\Theta^2 + \Theta - 2} = \frac{3\Theta^2 + 5\Theta + 4}{3\Theta + 3},$$

in the interval $(1, \infty)$*. Then,* $\Theta^{\text{D},\mathbb{R}}_{\text{SM}}$ *is the unique minimum of the function* $\max\{f^{\text{up}}_1, f^{\text{up}}_2\}$ *and* SMARTSTART *with scaling parameter* $\Theta^{\text{D},\mathbb{R}}_{\text{SM}}$ *is* $\rho^{\text{D},\mathbb{R}}_{\text{SM}}$*-competitive with*

$$\rho^{\text{D},\mathbb{R}}_{\text{SM}} = f^{\text{up}}_1(\Theta^{\text{D},\mathbb{R}}_{\text{SM}}) = f^{\text{up}}_2(\Theta^{\text{D},\mathbb{R}}_{\text{SM}}) \approx 2.9377.$$

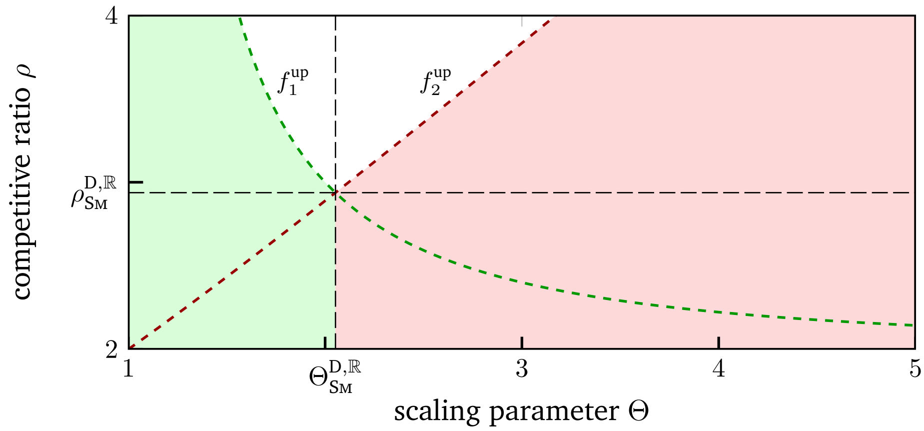

Figure 5.1: Functions f_1^{up} (green) / f_2^{up} (red): upper bounds for competitive ratio for postponing / non-postponing case. Green / red area: possible values for the competitive ratio, bounded by f_1^{up} / f_2^{up}.

Proof. For the case where SMARTSTART postpones the final schedule, we have established the upper bound

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

in Proposition 5.3 and for the case where SMARTSTART does not postpone final schedule, we have established the upper bound

$$\frac{\text{SMARTSTART}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{3\Theta^2 + 5\Theta + 4}{3\Theta + 3} = f_2^{\text{up}}(\Theta)$$

in Proposition 5.4. Thus, the maximum of both bounds is a general upper bound for the competitive ratio of SMARTSTART for open online DIAL-A-RIDE on the line that is independent of SMARTSTART's behavior before the final schedule.

Function f_1^{up} is strictly decreasing for $\Theta > 1$ and function f_2^{up} is strictly increasing for $\Theta > 1$. Therefore the minimum of $\max\{f_1^{\text{up}}, f_2^{\text{up}}\}$ in the interval $(1, \infty)$ lies in the intersection point of f_1^{up} and f_2^{up}, i.e., in $\Theta_{\text{SM}}^{\text{D},\mathbb{R}} \approx 2.0526$. The resulting upper bound for the competitive ratio is

$$\rho_{\text{SM}}^{\text{D},\mathbb{R}} = f_1^{\text{up}}(\Theta_{\text{SM}}^{\text{D},\mathbb{R}}) = f_2^{\text{up}}(\Theta_{\text{SM}}^{\text{D},\mathbb{R}}) \approx 2.9377. \qquad \square$$

See Figure 5.1 for a visualization of the upper bound for the competitive ratio of SMARTSTART for open online DIAL-A-RIDE on the line presented in Theorem 5.5

5.2 Lower Bound for Open Online DIAL-A-RIDE on the Line

To complete our analysis of SMARTSTART for open online DIAL-A-RIDE on the line, we give lower bound constructions for different values of Θ. In particular, we show that for $\Theta \in (2,3)$ there are request sequences where SMARTSTART postpones its final schedule and has competitive ratio at least $f_1^{\text{up}}(\Theta)$. Similarly, we show that for $\Theta \in [2, 2.303]$ there are instances where SMARTSTART does not postpone its final schedule and has competitive ratio at least $f_2^{\text{up}}(\Theta)$. Together, this implies that the general upper bound of $\max\{f_1^{\text{up}}(\Theta), f_2^{\text{up}}(\Theta)\}$ is tight for $\Theta \in (2, 2.303]$, and thus for $\Theta = \Theta_{\text{SM}}^{\text{D},\mathbb{R}}$ as defined in Theorem 5.5. To conclude the analysis of SMARTSTART for open online DIAL-A-RIDE on the line, we present four more lower bounds that establish that outside of the interval $(2, 2.303]$ there is no $\Theta \neq \Theta_{\text{SM}}^{\text{D},\mathbb{R}}$ that yields a better competitive ratio than $\rho_{\text{SM}}^{\text{D},\mathbb{R}}$. All our lower bounds rely on the following lemma that provides a way to *lure* SMARTSTART away from the origin with almost no time overhead. More specifically, the lemma provides a way to make SMARTSTART move to any position $p > 0$ within time $p + \mu$ where $\mu > 0$ is arbitrarily small.

Lemma 5.6. *Let $p > 0$ be any position on the real line and $\mu > 0$ be any positive number. Furthermore, let $\delta > 0$ be such that $\frac{p}{\delta\Theta} = m \in \mathbb{N}$ and $\delta < (\Theta - 1)\mu$. Algorithm* SMARTSTART *for open online* DIAL-A-RIDE *or online* TSP *finishes serving the request sequence $\sigma_{p,\mu}^{\text{lure}} = (s_1, \dots, s_{m+1})$ with*

$$s_i = \left(i\delta, i\delta; \frac{\delta}{\Theta - 1} + (i-1)\delta\right) \text{ for } i \in \{1, \dots, m\}$$
$$s_{m+1} = (p, p; m\delta + \mu) = \left(p, p; \frac{p}{\Theta} + \mu\right)$$

and reaches the position p at time $p + \mu$, provided that no additional requests appear until time $\frac{p}{\Theta} + \mu$. The final schedule serving s_{m+1} is started at time $\frac{p}{\Theta} + \mu$.

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

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

This is clear for $i = 1$: By definition, SMARTSTART starts from $p_1 = 0$. The schedule S_1 to serve s_1 is started at time

$$t_1 = \min\left\{t \in \mathbb{R}_{\geq 0} : t \geq \frac{\delta}{\Theta - 1} \wedge \frac{L(t, 0, (s_1))}{\Theta - 1} \leq t\right\} = \frac{\delta}{\Theta - 1},$$

and reaches position $p_2 = \delta$ at time $v_2 = \frac{1}{\Theta-1}\delta + \delta = \frac{\Theta}{\Theta-1}\delta$. Note that the release time of every request s_i is larger than t_1, ensuring that S_1 indeed only serves s_1.

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

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

On the other hand, we have

$$\frac{L\Big(\frac{\delta}{\Theta-1} + k\delta, k\delta, (s_{k+1})\Big)}{\Theta - 1} = \frac{\delta}{\Theta - 1} < \frac{\delta}{\Theta - 1} + k\delta = v_{k+1}.$$

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

$$v_{m+1} = t_m + L(t_m, p_m, (s_m)) = \frac{\delta}{\Theta - 1} + (m-1)\delta + \delta = \frac{\delta}{\Theta - 1} + m\delta < \mu + m\delta$$

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

$$\frac{L\big(\mu + m\delta, \frac{p}{\Theta}, (s_{m+1})\big)}{\Theta - 1} = \frac{\frac{\Theta-1}{\Theta}p}{\Theta - 1} = \frac{p}{\Theta} = m\delta < \mu + m\delta.$$

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

$$\begin{aligned} \textsc{Smartstart}((s_i)_{i \in \{1,\ldots,m+1\}}) &= t_{m+1} + L(t_{m+1}, p_{m+1}, (s_{m+1})) \\ &= \mu + \frac{p}{\Theta} + \frac{\Theta - 1}{\Theta}p \\ &= \mu + p. \end{aligned}$$

The request sequence $\sigma_{p,\mu}^{\text{lure}}$ contains no transportation request. Thus, our construction remains valid for every capacity $c \in \mathbb{N} \cup \{\infty\}$ and also for online TSP. Furthermore, there is no interference with requests that are released after time $t_{m+1} = \mu + \frac{p}{\Theta}$. $\square$

Equipped with this strategy to lure SMARTSTART away from the origin, we now move on to establish lower bounds matching Propositions 5.3 and 5.4. For convenience, whenever we apply Lemma 5.6, we start the enumeration of schedules with the first schedule after the subsequence $\sigma_{p,\mu}^{\text{lure}}$ is served. To make the analysis of the following constructions a bit more clear, we denote by w_j the earliest time, a potential waiting period before schedule S_j is over, i.e., $w_j = \frac{1}{\Theta-1} L(v_j, p_j, \sigma_{\leq v_j})$. Consequently, if no new requests appear between the ending time v_j of schedule S_{j-1} and the starting time of schedule S_j (which is the case for all request sequences that will be analyzed in this section), we have $t_j = \max\{v_j, w_j\}$.

In the following, we will analyze four different request sequences σ_1^{SM} to σ_4^{SM}. We will see that the ratio of SMARTSTART's and OPT's completion time of σ_1^{SM} tightly matches the upper bound of Proposition 5.3 for $\Theta \in (2,3)$ and that the ratio of SMARTSTART's and OPT's completion time of σ_2^{SM} tightly matches the upper bound of Proposition 5.4 for $\Theta \in [2, 2.303]$. The request sequences σ_3^{SM} and σ_4^{SM} will provide additional lower bounds for the competitive ratio of SMARTSTART for open online DIAL-A-RIDE on the line for larger values of Θ. We start with request sequence σ_1^{SM}.

Definition 5.7. Let $\varepsilon' > 0$ with $\varepsilon' < \frac{1}{\Theta}$. We define

$$\sigma_1^{\text{SM}} := (\sigma_{1,\varepsilon'/2}^{\text{lure}}, s_1^{(1)}, s_1^{(2)}),$$

where $\sigma_{1,\varepsilon'/2}^{\text{lure}}$ is a subsequence of requests resulting from the application of Lemma 5.6 with $p = 1$ and $\mu = \frac{\varepsilon'}{2}$ and

$$s_1^{(1)} = \left(-\frac{1}{\Theta} + \varepsilon', 0; \frac{1}{\Theta} + \varepsilon'\right),$$
$$s_1^{(2)} = \left(\frac{1}{\Theta}, 1; \frac{1}{\Theta} + \varepsilon'\right).$$

Note that both requests appear after time $\frac{1}{\Theta} + \frac{\varepsilon'}{2}$ and therefore do not interfere with the application of Lemma 5.6 and that $\varepsilon' < \frac{1}{\Theta}$ implies $a_1^{(1)} < 0$, i.e., the starting position of request $s_1^{(1)}$ is on the left side of the origin. We begin our analysis of σ_1^{SM} with the computation of $\text{OPT}(\sigma_1^{\text{SM}})$.

Lemma 5.8. *We have*

$$\text{OPT}(\sigma_1^{\text{SM}}) = \frac{\Theta + 2}{\Theta}.$$

Proof. OPT waits at the origin until time $2\varepsilon'$ and then performs the walk

$$0 \to -\frac{1}{\Theta} + \varepsilon' \to 1.$$

OPT's walk is presented in blue in Figure 5.2 for $\Theta = 1.9$ and in blue in Figure 5.3 for $\Theta = \Theta_{\text{SM}}^{\text{D},\mathbb{R}}$. We show that all requests are served this way: OPT collects $s_1^{(1)}$ at time $\frac{1}{\Theta} + \varepsilon'$

and delivers it to the origin at time $\frac{2}{\Theta}$. Let $q > 0$ be the position of a request of $\sigma^{\text{lure}}_{1,\varepsilon'/2}$ arising from the application of Lemma 5.6. Then this request is released earlier than time $q + \frac{\varepsilon'}{2}$. On the other hand, OPT reaches position q not earlier than time $\frac{2}{\Theta} + q$. Since we have $\varepsilon' < \frac{1}{\Theta}$, we have $\frac{2}{\Theta} + q > q + \frac{\varepsilon'}{2}$ and OPT can go straight from the origin to position 1, collecting and delivering all requests that occur by the application of Lemma 5.6 as well as $s_1^{(2)}$ on the way. Therefore, we have

$$\textsc{Opt}(\sigma_1^{\text{SM}}) = 2\varepsilon' + D\left(0 \to -\frac{1}{\Theta} + \varepsilon' \to 1\right) = \frac{\Theta + 2}{\Theta}.$$

OPT can do this even if the capacity is $c = 1$, since no transportation requests need to be carried over $[0, \frac{1}{\Theta}] \cup \{1\}$, where the requests of the application of Lemma 5.6 appear, and because the carrying paths of $s_1^{(1)}$ and $s_1^{(2)}$ are disjoint. □

Next, we compute SMARTSTART's completion time. We will see that SMARTSTART's completion time for serving σ_1^{SM} depends strongly on the choice of Θ.

Lemma 5.9. *Let $\Theta \in (1, 3)$ and $\varepsilon' < \frac{2}{9}$. Then, we have*

$$\textsc{SmartStart}(\sigma_1^{\text{SM}}) = \min\left\{\frac{3\Theta}{\Theta - 1}, \frac{2\Theta + 2}{\Theta - 1}\right\} - \frac{2\Theta}{\Theta - 1}\varepsilon'.$$

Proof. SMARTSTART's walk is presented in green in Figure 5.2 for $\Theta = 1.9$ and in green in Figure 5.3 for $\Theta = \Theta^{\text{D},\mathbb{R}}_{\text{SM}}$. SMARTSTART reaches position $p_1 = 1$ at time $v_1 = 1 + \frac{\varepsilon'}{2}$. The shortest schedule serving $s_1^{(2)}$ before serving $s_1^{(1)}$ has length

$$D\left(1 \to \frac{1}{\Theta} \to 1 \to -\frac{1}{\Theta} + \varepsilon' \to 0\right) = 3 - 2\varepsilon'.$$

On the other hand the shortest schedule that serves $s_1^{(1)}$ before serving $s_1^{(2)}$ has length

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

Thus, for all $t \geq v_1$, we have

$$L(t, p_1, (s_1^{(1)}, s_1^{(2)})) = \min\left\{3, 2 + \frac{2}{\Theta}\right\} - 2\varepsilon'.$$

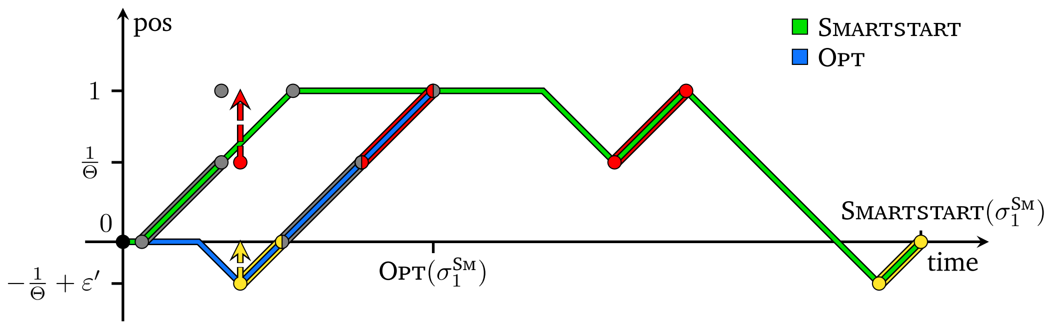

Figure 5.2: SmartStart's and Opt's walk serving σ_1^{SM} with $\varepsilon' = 0.25$ and $\Theta = 1.9$. Request $s_1^{(1)}$ is red ● and request $s_1^{(2)}$ is yellow ○. The requests of Lemma 5.6 are gray ●.

By assumption, we have $\Theta < 3$ and $\varepsilon' < \frac{2}{9}$, which implies that for the time $v_1 = 1 + \frac{\varepsilon'}{2}$, when SmartStart reaches position $p_1 = 1$, the inequality

$$w_1 = \min\left\{\frac{3}{\Theta-1}, \frac{2\Theta+2}{\Theta(\Theta-1)}\right\} - \frac{2}{\Theta-1}\varepsilon' \overset{\Theta<3}{>} \frac{4}{3} - \varepsilon' \overset{\varepsilon'<\frac{2}{9}}{>} 1 + \frac{\varepsilon'}{2} = v_1 \tag{5.21}$$

holds. Note that inequality (5.21) also holds for slightly larger Θ if we let $\varepsilon \to 0$. Because of inequality (5.21), SmartStart has a waiting period and starts the schedule S_1 at time

$$t_1 = \max\{v_1, w_1\} \overset{(5.21)}{=} w_1 = \min\left\{\frac{3}{\Theta-1}, \frac{2\Theta+2}{\Theta(\Theta-1)}\right\} - \frac{2}{\Theta-1}\varepsilon'.$$

In total, we have

$$\text{SmartStart}(\sigma_1^{\text{SM}}) = t_1 + L(t, p_1, (s_1^{(1)}, s_1^{(2)})) = \min\left\{\frac{3\Theta}{\Theta-1}, \frac{2\Theta+2}{\Theta-1}\right\} - \frac{2\Theta}{\Theta-1}\varepsilon'. \quad \square$$

Equipped with Lemmas 5.8 and 5.9, we can compute lower bounds for the competitive ratio of SmartStart for online Dial-a-Ride for $\Theta \in (1, 3)$. We start with the subinterval $(1, 2]$.

Lemma 5.10. *Let $1 < \Theta \le 2$. For every sufficiently small $\varepsilon > 0$, we have*

$$\frac{\text{SmartStart}(\sigma_1^{\text{SM}})}{\text{Opt}(\sigma_1^{\text{SM}})} = \frac{3\Theta^2}{\Theta^2+\Theta-2} - \varepsilon =: f_1^{\text{low}}(\Theta) - \varepsilon.$$

In particular, we have

$$\frac{\text{SmartStart}(\sigma_1^{\text{SM}})}{\text{Opt}(\sigma_1^{\text{SM}})} > \rho_{\text{SM}}^{\text{D},\mathbb{R}} \approx 2.9377$$

for $\Theta \in (1, 2]$ and sufficiently small $\varepsilon > 0$.

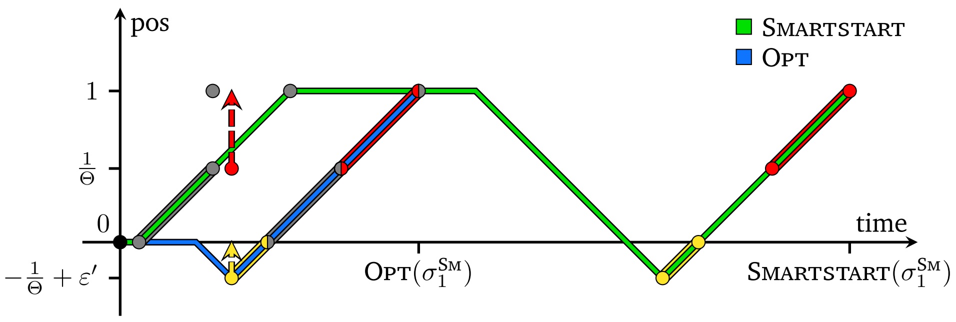

Figure 5.3: SmartStart's and Opt's walk serving σ_1^{SM} with $\varepsilon' = 0.25$ and $\Theta = \Theta_{\text{SM}}^{\text{D},\mathbb{R}}$. Request $s_1^{(1)}$ is red ● and request $s_1^{(2)}$ is yellow ●. The requests of Lemma 5.6 are gray ●.

Proof. Let $\varepsilon > 0$ with $\varepsilon < \min\{\frac{2}{9}(\frac{2\Theta^2}{\Theta^2+\Theta-2}), \frac{1}{20}\}$ and $\varepsilon' = \frac{\Theta^2+\Theta-2}{2\Theta^2}\varepsilon < \frac{2}{9}$. By Lemma 5.9, we have

$$\begin{aligned}\text{SmartStart}(\sigma_1^{\text{SM}}) &\overset{\text{Lem 5.9}}{=} \min\left\{\frac{3\Theta}{\Theta-1}, \frac{2\Theta+2}{\Theta-1}\right\} - \frac{2\Theta}{\Theta-1}\varepsilon' \\ &\overset{\Theta\leq 2}{=} \frac{3\Theta}{\Theta-1} - \frac{2\Theta}{\Theta-1}\varepsilon'\end{aligned}$$

Lemma 5.8 implies

$$\text{Opt}(\sigma_1^{\text{SM}}) = \frac{\Theta+2}{\Theta}.$$

Since we have $\varepsilon' = \frac{\Theta^2+\Theta-2}{2\Theta^2}\varepsilon$, we obtain

$$\frac{\text{SmartStart}(\sigma_1^{\text{SM}})}{\text{Opt}(\sigma_1^{\text{SM}})} = \frac{3\Theta^2}{\Theta^2+\Theta-2} - \frac{2\Theta^2}{\Theta^2+\Theta-2}\varepsilon' = \frac{3\Theta^2}{\Theta^2+\Theta-2} - \varepsilon = f_1^{\text{low}}(\Theta) - \varepsilon,$$

as claimed. The function f_1^{low} is monotonically decreasing on $(1,2]$. Therefore, we have

$$\frac{\text{SmartStart}(\sigma_1^{\text{SM}})}{\text{Opt}(\sigma_1^{\text{SM}})} - \varepsilon \geq f_1^{\text{low}}(2) - \varepsilon = 3 - \varepsilon > \rho_{\text{SM}}^{\text{D},\mathbb{R}} \approx 2.9377$$

for all $\Theta \in (1,2]$ and $\varepsilon < \frac{1}{20}$. □

The following proposition shows that in the case $\Theta \in (2,3)$ the ratio of SmartStart's and Opt's completion time for the request sequence σ_1^{SM} tightly matches the upper bound provided by Proposition 5.3.

Proposition 5.11. *Let $2 < \Theta < 3$. For every sufficiently small $\varepsilon > 0$ we have*

$$\frac{\textsc{Smartstart}(\sigma_1^{\textsc{Sm}})}{\textsc{Opt}(\sigma_1^{\textsc{Sm}})} = \frac{2\Theta^2 + 2\Theta}{\Theta^2 + \Theta - 2} - \varepsilon = f_1^{\text{up}}(\Theta) - \varepsilon$$

and SMARTSTART *postpones the final schedule, i.e., the upper bound established in Proposition 5.3 is tight for $\Theta \in (2, 3)$.*

Proof. Let $\varepsilon > 0$ with $\varepsilon < \frac{2}{9}(\frac{2\Theta^2}{\Theta^2+\Theta-2})$ and $\varepsilon' = \frac{\Theta^2+\Theta-2}{2\Theta^2}\varepsilon < \frac{2}{9}$. By Lemma 5.9, we have

$$\begin{aligned}\textsc{Smartstart}(\sigma_1^{\textsc{Sm}}) &\overset{\text{Lem 5.9}}{=} \min\left\{\frac{3\Theta}{\Theta-1}, \frac{2\Theta+2}{\Theta-1}\right\} - \frac{2\Theta}{\Theta-1}\varepsilon' \\ &\overset{\Theta \geq 2}{=} \frac{2\Theta+2}{\Theta-1} - \frac{2\Theta}{\Theta-1}\varepsilon'\end{aligned}$$

Lemma 5.8 implies

$$\textsc{Opt}(\sigma_1^{\textsc{Sm}}) = \frac{\Theta+2}{\Theta}.$$

Since we have $\varepsilon' = \frac{\Theta^2+\Theta-2}{2\Theta^2}\varepsilon$, we obtain

$$\frac{\textsc{Smartstart}(\sigma_1^{\textsc{Sm}})}{\textsc{Opt}(\sigma_1^{\textsc{Sm}})} = \frac{2\Theta^2 + 2\Theta}{\Theta^2 + \Theta - 2} - \frac{2\Theta^2}{\Theta^2 + \Theta - 2}\varepsilon' = \frac{2\Theta^2 + 2\Theta}{\Theta^2 + \Theta - 2} - \varepsilon = f_1^{\text{up}}(\Theta) - \varepsilon. \quad \square$$

Figure 5.4 is a visualization of the upper bound for the competitive ratio of online DIAL-A-RIDE on the line presented in Theorem 5.5 together with the lower bounds of Proposition 5.11 and Lemma 5.10. Next we examine the request sequence $\sigma_2^{\textsc{Sm}}$.

Definition 5.12. Let $\varepsilon' > 0$ with $\varepsilon' < \frac{\Theta+1}{\Theta}$. We define

$$\sigma_2^{\textsc{Sm}} := (\sigma_{1,\varepsilon'/2}^{\text{lure}}, s_1^{(1)}, s_1^{(2)}, s_2),$$

where $\sigma_{1,\varepsilon'/2}^{\text{lure}}$ is a subsequence of requests resulting from the application of Lemma 5.6 with $p = 1$ and $\mu = \frac{\varepsilon'}{2}$ and

$$\begin{aligned} s_1^{(1)} &= \left(\frac{2\Theta+1}{\Theta} - \varepsilon', \frac{2\Theta+1}{\Theta} - \varepsilon'; \frac{1}{\Theta} + \varepsilon'\right), \\ s_1^{(2)} &= \left(-\frac{1}{\Theta}, -\frac{1}{\Theta}; \frac{1}{\Theta} + \varepsilon'\right), \\ s_2 &= \left(\max\left\{\frac{\Theta+5}{\Theta^2-\Theta}, \frac{2\Theta+1}{\Theta}\right\} - \varepsilon', \max\left\{\frac{\Theta+5}{\Theta^2-\Theta}, \frac{2\Theta+1}{\Theta}\right\} - \varepsilon'; \frac{3\Theta+3}{\Theta^2-\Theta}\right) \end{aligned}$$

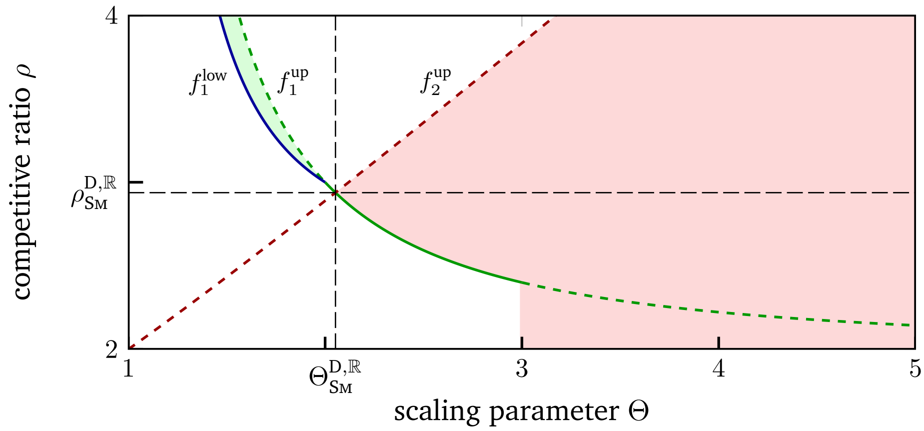

Figure 5.4: Functions f_1^{up} (green) / f_2^{up} (red): upper bounds for competitive ratio for postponing / non-postponing case, drawn solid if tight. Function: f_1^{low} (blue): lower bounds for competitive ratio. Green / red area: possible values for the competitive ratio, bounded by f_1^{up} / f_2^{up} and f_1^{low}.

Note that the requests $s_1^{(1)}$, $s_1^{(2)}$ and s_2 appear after time $\frac{1}{\Theta} + \frac{\varepsilon'}{2}$ and therefore do not interfere with the application of Lemma 5.6 and that $\varepsilon' < \frac{\Theta+1}{\Theta}$ implies $a_2 \geq a_1^{(1)} > 1$, i.e., both requests $s_1^{(1)}$ and s_2 appear on the right side of position 1. Note that σ_2^{SM} contains no transportation requests. Thus, every lower bound implied by request sequence σ_2^{SM} is also valid for open online TSP. We begin our analysis of σ_2^{SM} with the computation of $\textsc{Opt}(\sigma_2^{\mathrm{SM}})$.

Lemma 5.13. *We have*

$$\textsc{Opt}(\sigma_2^{\mathrm{SM}}) = \max\left\{\frac{3\Theta+3}{\Theta^2-\Theta}, \frac{2\Theta+3}{\Theta}\right\}.$$

Proof. Opt waits at the origin until time ε' and then performs the walk

$$0 \to -\frac{1}{\Theta} \to \max\left\{\frac{\Theta+5}{\Theta^2-\Theta}, \frac{2\Theta+1}{\Theta}\right\} - \varepsilon'.$$

Opt's walk is presented in blue in Figure 5.5 for $\Theta = \Theta_{\mathrm{SM}}^{\mathrm{D},\mathbb{R}}$ and in blue in Figure 5.6 for $\Theta = 2.5$. We show that all requests are served this way: Opt serves the request $s_1^{(2)}$ at time $\frac{1}{\Theta} + \varepsilon'$ and returns to the origin at time $\frac{2}{\Theta} + \varepsilon'$. Let $q > 0$ be the position of a request that has occurred by the application of Lemma 5.6. Then this request is released earlier than time $q + \frac{\varepsilon'}{2}$. Since Opt reaches position q not earlier than time $\frac{2}{\Theta} + \varepsilon' + q > q + \frac{\varepsilon'}{2}$, OPT can go straight from the origin to the right and can serve all requests of Lemma 5.6 without waiting. OPT reaches position $\max\{\frac{\Theta+5}{\Theta^2-\Theta}, \frac{2\Theta+1}{\Theta}\} - \varepsilon'$ at time

$$\begin{aligned}\varepsilon' + D\left(0 \to -\frac{1}{\Theta} \to \max\left\{\frac{\Theta+5}{\Theta^2-\Theta}, \frac{2\Theta+1}{\Theta}\right\} - \varepsilon'\right) &= \max\left\{\frac{3\Theta+3}{\Theta^2-\Theta}, \frac{2\Theta+3}{\Theta}\right\} \\ &\geq \frac{3\Theta+3}{\Theta^2-\Theta} \\ &= r_2,\end{aligned}$$

i.e., OPT serves the requests $s_1^{(1)}$ and s_2 at arrival and we have

$$\text{OPT}(\sigma_2^{\text{SM}}) = \max\left\{\frac{3\Theta+3}{\Theta^2-\Theta}, \frac{2\Theta+3}{\Theta}\right\}. \qquad \square$$

Next, we compute SMARTSTART's completion time.

Lemma 5.14. *Let* $\Theta \in [\frac{7}{4}, 4]$ *and* $\varepsilon' < \frac{3}{14}$. *Then, we have*

$$\text{SMARTSTART}(\sigma_2^{\text{SM}}) = \max\left\{\frac{3\Theta^2+5\Theta+4}{\Theta^2-\Theta}, \frac{5\Theta^2+3\Theta-2}{\Theta^2-\Theta}\right\} - \frac{3\Theta-1}{\Theta-1}\varepsilon'.$$

Proof. SMARTSTART's walk is presented in green in Figure 5.5 for $\Theta = \Theta_{\text{SM}}^{\text{D},\mathbb{R}}$ and in green in Figure 5.6 for $\Theta = 2.5$. SMARTSTART reaches position $p_1 = 1$ at time $v_1 = 1 + \frac{\varepsilon'}{2}$. At this time the requests $s_1^{(1)}$ and $s_1^{(2)}$ are released, but s_2 is not. The shortest schedule serving $s_1^{(2)}$ before serving $s_1^{(1)}$ has length

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

On the other hand, the shortest schedule serving $s_1^{(2)}$ after serving $s_1^{(1)}$ has length

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

Thus, SMARTSTART serves $s_1^{(2)}$ after serving $s_1^{(1)}$, and, for all $t \geq v_1$, we obtain

$$L(t, p_1, (s_1^{(1)}, s_1^{(2)})) = \frac{3\Theta+3}{\Theta} - 2\varepsilon'.$$

By assumption, we have $\Theta \leq 4$ and $\varepsilon' < \frac{3}{14}$, which implies that for the time $v_1 = 1 + \frac{\varepsilon'}{2}$, when SMARTSTART reaches position $p_1 = 1$, the inequality

$$w_1 = \frac{3\Theta+3}{\Theta(\Theta-1)} - \frac{2\varepsilon'}{\Theta-1} \overset{\Theta \leq 4}{\geq} \frac{5}{4} - \frac{2}{3}\varepsilon' \overset{\varepsilon' < \frac{3}{14}}{>} 1 + \frac{\varepsilon'}{2} = v_1 \tag{5.22}$$

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

$$t_1 = \max\{v_1, w_1\} \overset{(5.22)}{=} w_1 = \frac{3\Theta+3}{\Theta(\Theta-1)} - \frac{2\varepsilon'}{\Theta-1},$$

which is before s_2 is released. SMARTSTART finishes schedule S_1 at time

$$v_2 = t_1 + L(t_1, p_1, (s_1^{(1)}, s_1^{(2)})) = \frac{3\Theta+3}{\Theta-1} - \frac{2\Theta}{\Theta-1}\varepsilon'$$

at position $p_2 = -\frac{1}{\Theta}$. It remains to serve s_2. For all $t \geq v_2$, we obtain

$$\begin{aligned} L(t, p_2, (s_2)) &= D\left(-\frac{1}{\Theta} \to \max\left\{\frac{\Theta+5}{\Theta^2-\Theta}, \frac{2\Theta+1}{\Theta}\right\} - \varepsilon'\right) \\ &= \max\left\{\frac{2\Theta+4}{\Theta^2-\Theta}, \frac{2\Theta+2}{\Theta}\right\} - \varepsilon'. \end{aligned}$$

Assume we have $\max\left\{\frac{2\Theta+4}{\Theta^2-\Theta}, \frac{2\Theta+2}{\Theta}\right\} = \frac{2\Theta+4}{\Theta^2-\Theta}$. By assumption, we have $\Theta \geq \frac{7}{4}$ and $\varepsilon' < \frac{3}{14} < \frac{3\Theta^3-5\Theta-4}{2\Theta^3-3\Theta^2+\Theta}$. For the finishing time $v_2 = \frac{3\Theta+3}{\Theta-1} - \frac{2\Theta}{\Theta-1}\varepsilon'$ of schedule S_1, we have the inequality

$$\begin{aligned} w_2 &= \frac{2\Theta+4}{\Theta(\Theta-1)^2} - \frac{\varepsilon'}{\Theta-1} \\ &= \frac{2\Theta+4}{\Theta(\Theta-1)^2} + \frac{2\Theta-1}{\Theta-1}\varepsilon' - \frac{2\Theta}{\Theta-1}\varepsilon' \\ &\overset{\varepsilon' < \frac{3\Theta^3-5\Theta-4}{2\Theta^3-3\Theta^2+\Theta}}{<} \frac{3\Theta+3}{\Theta-1} - \frac{2\Theta}{\Theta-1}\varepsilon' \\ &= v_2. \end{aligned} \tag{5.23}$$

Note that inequality (5.23) still holds for slightly smaller Θ if we let $\varepsilon \to 0$. Now assume we have $\max\{\frac{2\Theta+4}{\Theta^2-\Theta}, \frac{2\Theta+2}{\Theta}\} = \frac{2\Theta+2}{\Theta}$. By assumption, we have $\varepsilon' < \frac{3}{14} < \frac{\Theta+1}{\Theta}$, which implies that, for the finishing time $v_2 = \frac{3\Theta+3}{\Theta-1} - \frac{2\Theta}{\Theta-1}\varepsilon'$ of schedule S_1, the inequality

$$w_2 = \frac{2\Theta+2}{\Theta(\Theta-1)} - \frac{\varepsilon'}{\Theta-1}$$

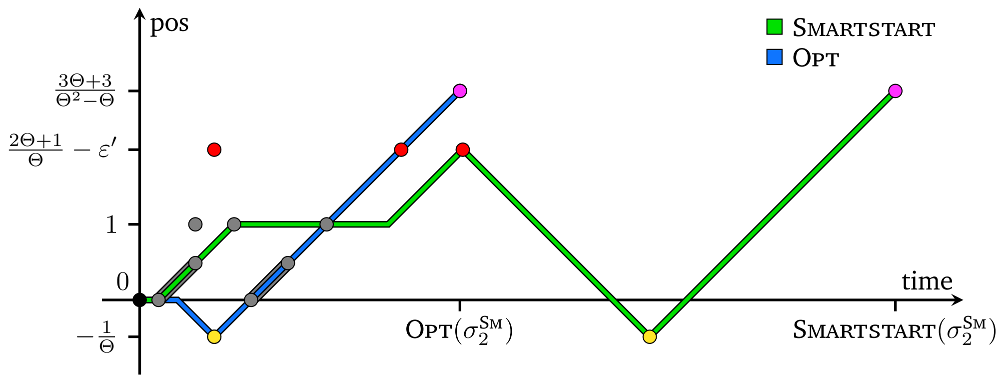

Figure 5.5: SMARTSTART's and OPT's walk serving σ_2^{SM} with $\varepsilon' = 0.5$ and $\Theta = \Theta_{\text{SM}}^{\text{D},\mathbb{R}}$. Request $s_1^{(1)}$ is red ●, request $s_1^{(2)}$ is yellow ● and request s_2 is violet ●. The requests of Lemma 5.6 are gray ●.

$$\begin{aligned}
&= \frac{1}{\Theta}\left(\frac{3\Theta+3}{\Theta-1} - \frac{\Theta+1+\varepsilon'\Theta}{\Theta-1}\right) \\
&\overset{\varepsilon < \frac{\Theta+1}{\Theta}}{<} \frac{1}{\Theta}\left(\frac{3\Theta+3}{\Theta-1} - \frac{2\Theta}{\Theta-1}\varepsilon'\right) \\
&\overset{\Theta > 1}{<} \frac{3\Theta+3}{\Theta-1} - \frac{2\Theta}{\Theta-1}\varepsilon' \\
&= v_2
\end{aligned} \tag{5.24}$$

holds. Because of the inequalities (5.23), (5.24) starting time of the schedule S_2 is the ending time of the schedule S_1, i.e.,

$$t_2 = \max\{v_2, w_2\} = v_2 = \frac{3\Theta+3}{\Theta-1} - \frac{2\Theta}{\Theta-1}\varepsilon'.$$

To sum it up, we have

$$\begin{aligned}
\text{SMARTSTART}(\sigma_2^{\text{SM}}) &= t_2 + L(t_2, p_2, (s_2)) \\
&= \max\left\{\frac{3\Theta^2+5\Theta+4}{\Theta^2-\Theta}, \frac{5\Theta^2+3\Theta-2}{\Theta^2-\Theta}\right\} - \frac{3\Theta-1}{\Theta-1}\varepsilon'.
\end{aligned}$$ □

Equipped with the Lemmas 5.13 and 5.14, we can compute lower bounds for the competitive ratio of SMARTSTART for online DIAL-A-RIDE for $\Theta \in [\frac{7}{4}, 4]$. We start with the subinterval $[\frac{7}{4}, \frac{1}{2}(1+\sqrt{13})]$.

Proposition 5.15. *Let $\frac{7}{4} \leq \Theta \leq \frac{1}{2}(1+\sqrt{13})$. For every sufficiently small $\varepsilon > 0$, we have*

$$\frac{\textsc{Smartstart}(\sigma_2^{\text{SM}})}{\textsc{Opt}(\sigma_2^{\text{SM}})} = \frac{3\Theta^2+5\Theta+4}{3\Theta+3} - \varepsilon = f_2^{\text{up}}(\Theta) - \varepsilon$$

and SMARTSTART *does not postpone the final schedule, i.e., the upper bound established in Proposition 5.3 is tight for $\Theta \in [\frac{7}{4}, \frac{1}{2}(1+\sqrt{13})] \approx [2, 2.303]$.*

Proof. Let $\varepsilon > 0$ with $\varepsilon < \frac{3}{14}\left(\frac{3\Theta^2-\Theta}{3\Theta+3}\right)$ and $\varepsilon' = \frac{3\Theta+3}{3\Theta^2-\Theta}\varepsilon < \frac{3}{14}$. By Lemma 5.14, we have

$$\begin{aligned}\textsc{Smartstart}(\sigma_2^{\text{SM}}) &\overset{\text{Lem 5.14}}{=} \max\left\{\frac{3\Theta^2+5\Theta+4}{\Theta^2-\Theta}, \frac{5\Theta^2+3\Theta-2}{\Theta^2-\Theta}\right\} - \frac{3\Theta-1}{\Theta-1}\varepsilon' \\ &\overset{\Theta \leq \frac{1}{2}(1+\sqrt{13})}{=} \frac{3\Theta^2+5\Theta+4}{\Theta^2-\Theta} - \frac{3\Theta-1}{\Theta-1}\varepsilon'.\end{aligned}$$

Lemma 5.13 implies

$$\textsc{Opt}(\sigma_2^{\text{SM}}) = \max\left\{\frac{3\Theta+3}{\Theta^2-\Theta}, \frac{2\Theta+3}{\Theta}\right\} \overset{\Theta\leq\frac{1}{2}(1+\sqrt{13})}{=} \frac{3\Theta+3}{\Theta^2-\Theta}.$$

Since we have $\varepsilon' = \frac{3\Theta+3}{3\Theta^2-\Theta}\varepsilon$, we finally obtain

$$\frac{\textsc{Smartstart}(\sigma_2^{\text{SM}})}{\textsc{Opt}(\sigma_2^{\text{SM}})} = \frac{3\Theta^2+5\Theta+4}{3\Theta+3} - \frac{3\Theta^2-\Theta}{3\Theta+3}\varepsilon' = \frac{3\Theta^2+5\Theta+4}{3\Theta+3} - \varepsilon = f_2^{\text{up}}(\Theta) - \varepsilon. \quad \square$$

Next, we examine the subinterval $(\frac{1}{2}(1+\sqrt{13}), 4]$.

Lemma 5.16. *Let $\frac{1}{2}(1+\sqrt{13}) < \Theta \leq 4$. For every sufficiently small $\varepsilon > 0$, we have*

$$\frac{\textsc{Smartstart}(\sigma_2^{\text{SM}})}{\textsc{Opt}(\sigma_2^{\text{SM}})} = \frac{5\Theta^2+3\Theta-2}{2\Theta^2+\Theta-3} - \varepsilon =: f_2^{\text{low}}(\Theta) - \varepsilon.$$

In particular, we have

$$\frac{\textsc{Smartstart}(\sigma_2^{\text{SM}})}{\textsc{Opt}(\sigma_2^{\text{SM}})} > \rho_{\text{SM}}^{\text{D},\mathbb{R}} \approx 2.93768$$

for $\Theta \in (\frac{1}{2}(1+\sqrt{13}, \sqrt{7}] \approx (2.303, 2.646]$.

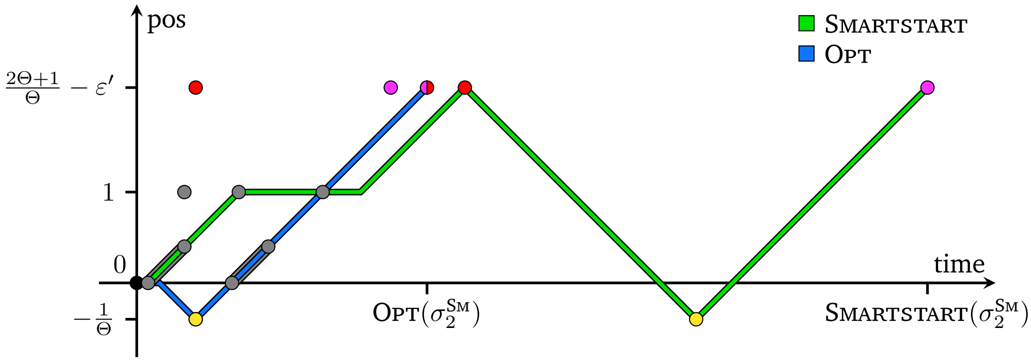

Figure 5.6: SMARTSTART's and OPT's walk serving σ_2^{SM} with $\varepsilon' = 0.25$ and $\Theta = 2.5$. Request $s_1^{(1)}$ is red ●, request $s_1^{(2)}$ is yellow ○ and request s_2 is violet ●. The requests of Lemma 5.6 are gray ●.

Proof. Let $\varepsilon > 0$ with $\varepsilon < \min\{\frac{3}{14}(\frac{3\Theta^2-\Theta}{3\Theta+3}), \frac{1}{20}\}$ and $\varepsilon' = \frac{3\Theta+3}{3\Theta^2-\Theta}\varepsilon < \frac{3}{14}$. By Lemma 5.14, we have

$$\begin{aligned}\textsc{SmartStart}(\sigma_2^{\text{SM}}) &\overset{\text{Lem 5.14}}{=} \max\left\{\frac{3\Theta^2+5\Theta+4}{\Theta^2-\Theta}, \frac{5\Theta^2+3\Theta-2}{\Theta^2-\Theta}\right\} - \frac{3\Theta-1}{\Theta-1}\varepsilon' \\ &\overset{\Theta>\frac{1}{2}(1+\sqrt{13})}{=} \frac{5\Theta^2+3\Theta-2}{\Theta^2-\Theta} - \frac{3\Theta-1}{\Theta-1}\varepsilon'.\end{aligned}$$

Lemma 5.13 implies

$$\textsc{Opt}(\sigma_2^{\text{SM}}) = \max\left\{\frac{3\Theta+3}{\Theta^2-\Theta}, \frac{2\Theta+3}{\Theta}\right\} \overset{\Theta>\frac{1}{2}(1+\sqrt{13})}{=} \frac{2\Theta+3}{\Theta}.$$

An illustration of OPT's walk is presented in blue in Figure 5.6. Since we have $\varepsilon' = \frac{3\Theta+3}{3\Theta^2-\Theta}\varepsilon$, we finally obtain

$$\begin{aligned}\frac{\textsc{SmartStart}(\sigma_2^{\text{SM}})}{\textsc{Opt}(\sigma_2^{\text{SM}})} &= \frac{5\Theta^2+3\Theta-2}{2\Theta^2+\Theta-3} - \frac{3\Theta^2-\Theta}{2\Theta^2+\Theta-3}\varepsilon' \\ &= \frac{5\Theta^2+3\Theta-2}{2\Theta^2+\Theta-3} - \varepsilon \\ &= f_2^{\text{low}}(\Theta) - \varepsilon,\end{aligned}$$

as claimed. The function f_2^{low} is monotonically decreasing on $(\frac{1}{2}(1+\sqrt{13}), \sqrt{7}]$. Therefore, we have

$$\frac{\textsc{SmartStart}(\sigma_2^{\text{SM}})}{\textsc{Opt}(\sigma_2^{\text{SM}})} \geq f_2^{\text{low}}(\sqrt{7}) - \varepsilon = 3 - \varepsilon' > \rho_{\text{SM}}^{\text{D},\mathbb{R}} \approx 2.93768$$

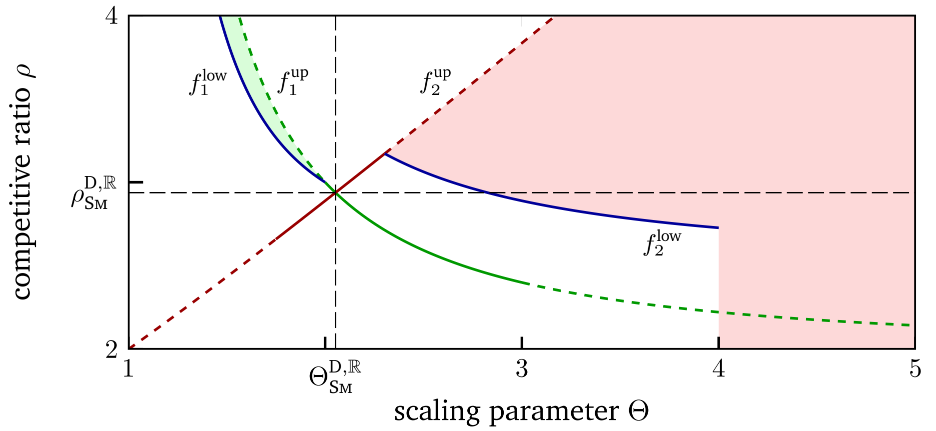

Figure 5.7: Functions f_1^{up} (green) / f_2^{up} (red): upper bounds for the competitive ratio for the postponing / non-postponing case, drawn solid if tight. Functions: f_1^{low}, f_2^{low} (blue): lower bounds for the competitive ratio. Green / red area: possible values for the competitive ratio, bounded by f_1^{up} / f_2^{up} and f_1^{low}, f_2^{low}.

for all $(\frac{1}{2}(1+\sqrt{13}), \sqrt{7}]$ and $\varepsilon < \frac{1}{20}$. $\square$

Figure 5.7 is a visualization of the upper bound for the competitive ratio of online DIAL-A-RIDE on the line presented in Theorem 5.5 together with the lower bounds of Propositions 5.11 and 5.15 as well as Lemmas 5.10 and 5.16.

Recall that the optimal parameter $\Theta_{\text{SM}}^{\text{D},\mathbb{R}}$ established in Theorem 5.5 is the only positive, real solution of the equation

$$\frac{2\Theta^2+2\Theta}{\Theta^2+\Theta-2} = \frac{3\Theta^2+5\Theta+4}{3\Theta+3},$$

which is $\Theta_{\text{SM}}^{\text{D},\mathbb{R}} \approx 2.0526$. Therefore, by Proposition 5.11 and Proposition 5.15 the parameter $\Theta_{\text{SM}}^{\text{D},\mathbb{R}}$ lies in the interval where the upper bounds of Propositions 5.3 and 5.4 are both tight. Moreover, by Propositions 5.11 and 5.15 and by Lemmas 5.10 and 5.16 there is no scaling parameter $\Theta \in (1,\sqrt{7}] \setminus \{\Theta_{\text{SM}}^{\text{D},\mathbb{R}}\}$ that yields an equal or better competitive ratio than $\Theta_{\text{SM}}^{\text{D},\mathbb{R}}$ does. Thus, it remains to make sure that there is no $\Theta > \sqrt{7}$ that yields an equal or better competitive ratio than $\rho_{\text{SM}}^{\text{D},\mathbb{R}} \approx 2.93768$. To show this, we analyze two more request sequences σ_3^{SM} and σ_4^{SM}. We start with σ_3^{SM}.

Definition 5.17. Let $\varepsilon' > 0$ with $\varepsilon' < \max\{\frac{2\Theta-2}{\Theta}, \frac{1}{\Theta}\}$. We define

$$\sigma_3^{\text{SM}} := (\sigma_{1,\varepsilon'/2}^{\text{lure}}, s_1^{(1)}, s_1^{(2)}, s_2),$$

where $\sigma^{\text{lure}}_{1,\varepsilon'/2}$ is a subsequence of requests resulting from the application of Lemma 5.6 with $p = 1$ and $\mu = \frac{\varepsilon'}{2}$ and

$$\begin{aligned} s_1^{(1)} &= \left(\frac{1}{\Theta} + \frac{\varepsilon'}{2}, 1; \frac{1}{\Theta} + \varepsilon'\right), \\ s_1^{(2)} &= \left(-\frac{1}{\Theta} + \varepsilon', -\frac{1}{\Theta} + \varepsilon'; \frac{1}{\Theta} + \varepsilon'\right), \\ s_2 &= \left(1, 1; \frac{3\Theta - 1}{\Theta(\Theta - 1)}\right). \end{aligned}$$

Note that the requests $s_1^{(1)}$, $s_1^{(2)}$ and s_2 requests appear after time $\frac{1}{\Theta} + \frac{\varepsilon'}{2}$ and therefore do not interfere with the application of Lemma 5.6. Furthermore, $\varepsilon' < \frac{2\Theta-2}{\Theta}$ implies $a_1^{(1)} < 1$ and $\varepsilon' < \frac{1}{\Theta}$ implies $a_1^{(2)} < 0$, i.e., the position of request $s_1^{(1)}$ is on the left side of position 1 and the position of request $s_1^{(2)}$ is on the left side of the origin. We begin our analysis of σ_3^{SM} with the computation of $\text{OPT}(\sigma_3^{\text{SM}})$.

Lemma 5.18. *Let $\Theta \geq 1 + \sqrt{2}$. Then, we have*

$$\text{OPT}(\sigma_3^{\text{SM}}) = \frac{\Theta + 2}{\Theta}.$$

Proof. OPT waits at the origin until time $2\varepsilon'$ and then performs the walk

$$0 \to -\frac{1}{\Theta} + \varepsilon' \to 1.$$

OPT's walk is presented in blue in Figure 5.8 for $\Theta = 2.75$. We show that all requests are served this way: OPT collects $s_1^{(1)}$ at time $\frac{1}{\Theta} + \varepsilon'$ and returns to the origin at time $\frac{2}{\Theta}$. Let $q > 0$ be the position of a request that has occurred by the application of Lemma 5.6. Then this request is released earlier than time $q + \frac{\varepsilon'}{2}$. OPT reaches position q not earlier than time $\frac{2}{\Theta}$. We have $\varepsilon' < \frac{1}{\Theta}$, which implies $\frac{2}{\Theta} + q > q + \frac{\varepsilon'}{2}$ and OPT can go straight from position $-\frac{1}{\Theta} + \varepsilon$ to position 1 collecting and delivering all requests that occur by the application of Lemma 5.6 as well as $s_1^{(2)}$. Note that OPT can also collect s_2 at arrival at position 1 at time $1 + \frac{2}{\Theta}$ since we have

$$1 + \frac{2}{\Theta} \overset{\Theta \geq 1+\sqrt{2}}{\geq} \frac{3\Theta - 1}{\Theta(\Theta - 1)}.$$

Therefore, we have

$$\text{OPT}(\sigma_3^{\text{SM}}) = 2\varepsilon' + D\left(0 \to -\frac{1}{\Theta} + \varepsilon' \to 1\right) = \frac{\Theta + 2}{\Theta}.$$

Note that OPT can do this even if capacity $c = 1$ holds since no additional requests need to be carried over $[0, \frac{1}{\Theta}] \cup \{1\}$, where the requests of the application of Lemma 5.6 appear. $\square$

Next, we examine SMARTSTART's completion time.

Lemma 5.19. *Let $\Theta \in [1 + \sqrt{2}, 3)$ and $\varepsilon' < \frac{2}{9}$. Then, we have*

$$\textsc{SmartStart}(\sigma_3^{\text{SM}}) = \frac{4\Theta^2 - \Theta - 1}{\Theta^2 - \Theta} - \frac{3\Theta - 1}{\Theta - 1}\varepsilon'.$$

Proof. SMARTSTART's walk is presented in green in Figure 5.8 for $\Theta = 2.75$. SMARTSTART reaches position $p_1 = 1$ at time $v_1 = 1 + \frac{\varepsilon'}{2}$. The shortest schedule serving $s_1^{(2)}$ before delivering $s_1^{(1)}$ has length

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

On the other hand, the shortest schedule that serves $s_1^{(2)}$ after delivering $s_1^{(1)}$ has length

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

By assumption, we have $\Theta < 3$, which implies $3 - \frac{1}{\Theta} - 2\varepsilon' < 2 + \frac{2}{\Theta} - 2\varepsilon'$. Therefore SMARTSTART serves $s_1^{(2)}$ after delivering $s_1^{(1)}$ and for all $t \geq v_1$ we have

$$L(t, p_1, (s_1^{(1)}, s_1^{(2)})) = 3 - \frac{1}{\Theta} - 2\varepsilon'.$$

Again, by assumption, we have $\Theta < 3$ and $\varepsilon' < \frac{2}{9}$, which implies that for the time $v_1 = 1 + \frac{\varepsilon'}{2}$, when SMARTSTART reaches position $p_1 = 1$ the inequality

$$w_1 = \frac{3\Theta - 1}{\Theta(\Theta - 1)} - \frac{2\varepsilon'}{\Theta - 1} \overset{\Theta<3}{>} \frac{4}{3} - \varepsilon' \overset{\varepsilon'<\frac{2}{9}}{>} 1 + \frac{\varepsilon'}{2} = v_1 \tag{5.25}$$

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

$$t_1 = \max\{v_1, w_1\} \overset{(5.25)}{=} w_1 = \frac{3\Theta - 1}{\Theta(\Theta - 1)} - \frac{2\varepsilon'}{\Theta - 1}$$

before request s_3 is released. SMARTSTART finishes schedule S_1 at time

$$v_2 = t_1 + L(t_1, p_1, (s_1^{(1)}, s_1^{(2)})) = \frac{3\Theta - 1}{\Theta - 1} - \frac{2\Theta}{\Theta - 1}\varepsilon'.$$

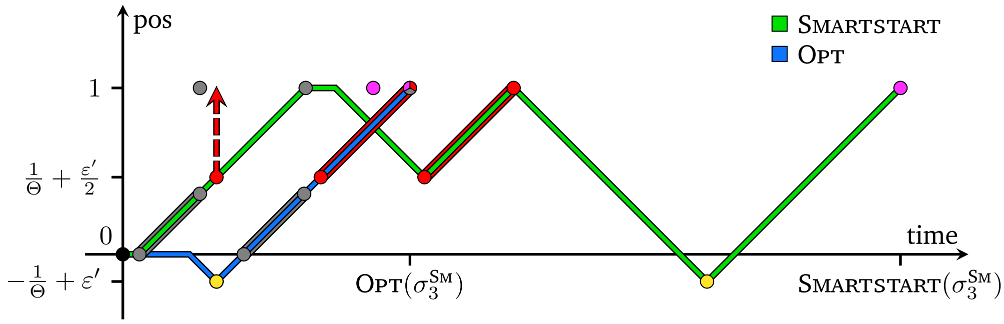

Figure 5.8: SmartStart's and Opt's walk serving σ_3^{SM} with $\varepsilon' = 0.2$ and $\Theta = 2.75$. Request $s_1^{(1)}$ is red ●, request $s_1^{(2)}$ is yellow ○ and request s_2 is violet ●. The requests of Lemma 5.6 are gray ●.

at position $p_2 = -\frac{1}{\Theta} + \varepsilon'$. For all $t \geq v_2$, we have

$$L(t, p_2, (s_2)) = D\left(-\frac{1}{\Theta} + \varepsilon' \to 1\right) = 1 + \frac{1}{\Theta} - \varepsilon'.$$

By assumption, we have $\Theta \geq 1 + \sqrt{2}$ and $\varepsilon' < \frac{2}{9} < \frac{3\Theta^2 - 2\Theta - 1}{\Theta}$. For the finishing time v_2 of schedule S_1 the inequality

$$\begin{aligned} w_2 &= \frac{1+\frac{1}{\Theta}}{\Theta - 1} - \frac{\varepsilon'}{\Theta - 1} \\ &= \frac{1+\frac{1}{\Theta}}{\Theta - 1} + \frac{(2\Theta - 1)\varepsilon'}{\Theta - 1} - \frac{2\Theta\varepsilon'}{\Theta - 1} \\ &\overset{\varepsilon' < \frac{3\Theta^2 - 2\Theta - 1}{\Theta}}{<} \frac{3\Theta - 1}{\Theta - 1} - \frac{2\varepsilon'\Theta}{\Theta - 1} \\ &= v_2. \end{aligned} \tag{5.26}$$

holds. Therefore the final schedule S_2 is started at time

$$t_2 = \max\{v_2, w_2\} \overset{(5.26)}{=} v_2 = \frac{3\Theta - 1}{\Theta - 1} - \frac{2\varepsilon'\Theta}{\Theta - 1}.$$

To sum it up, we have

$$\text{SmartStart}(\sigma_3^{\text{SM}}) = t_2 + L(t_2, p_2, (s_2)) = \frac{4\Theta^2 - \Theta - 1}{\Theta^2 - \Theta} - \frac{3\Theta - 1}{\Theta - 1}\varepsilon'. \qquad \square$$

Equipped with the Lemmas 5.18 and 5.19, we can compute lower bounds for the competitive ratio of SmartStart for online Dial-a-Ride for $\Theta \in [1 + \sqrt{2}, 3)$.

Lemma 5.20. *Let $1+\sqrt{2} \leq \Theta < 3$. For every sufficiently small $\varepsilon > 0$, we have*

$$\frac{\textsc{Smartstart}(\sigma_3^{\text{SM}})}{\textsc{Opt}(\sigma_3^{\text{SM}})} = \frac{4\Theta^2-\Theta-1}{\Theta^2+\Theta-2} - \varepsilon =: f_3^{\text{low}}(\Theta) - \varepsilon.$$

In particular, we have

$$\frac{\textsc{Smartstart}(\sigma_3^{\text{SM}})}{\textsc{Opt}(\sigma_3^{\text{SM}})} > \rho_{\text{SM}}^{\text{D},\mathbb{R}} \approx 2.93768$$

for $\Theta \in [1+\sqrt{2}, 3) \approx [2.414, 3)$.

Proof. Let $\varepsilon > 0$ with $\varepsilon < \min\{\frac{2}{9}(\frac{3\Theta^2-\Theta}{\Theta^2+\Theta-2}), \frac{1}{20}\}$ and $\varepsilon' = \frac{\Theta^2+\Theta-2}{3\Theta^2-\Theta}\varepsilon < \frac{2}{9}$. By Lemma 5.19 we have

$$\textsc{Smartstart}(\sigma_3^{\text{SM}}) = \frac{4\Theta^2-\Theta-1}{\Theta^2-\Theta} - \frac{3\Theta-1}{\Theta-1}\varepsilon'.$$

By Lemma 5.18 we have

$$\textsc{Opt}(\sigma_3^{\text{SM}}) = \frac{\Theta+2}{\Theta}.$$

We have

$$\begin{aligned}\frac{\textsc{Smartstart}(\sigma_3^{\text{SM}})}{\textsc{Opt}(\sigma_3^{\text{SM}})} &= \frac{4\Theta^2-\Theta-1}{\Theta^2+\Theta-2} - \frac{3\Theta^2-\Theta}{\Theta^2+\Theta-2}\varepsilon' \\ &= \frac{4\Theta^2-\Theta-1}{\Theta^2+\Theta-2} - \varepsilon \\ &= f_3^{\text{low}}(\Theta) - \varepsilon\end{aligned}$$

as claimed. The function f_3^{low} has exactly one local minimum in the interval $[1+\sqrt{2}, 3)$ at $\Theta = \frac{7}{5} + \frac{\sqrt{34}}{5}$. Therefore, we have

$$\frac{\textsc{Smartstart}(\sigma_3^{\text{SM}})}{\textsc{Opt}(\sigma_3^{\text{SM}})} \geq f_3^{\text{low}}\left(\frac{7}{5} + \frac{\sqrt{34}}{5}\right) - \varepsilon > 3 - \varepsilon > \rho_{\text{SM}}^{\text{D},\mathbb{R}} \approx 2.9377$$

for all $[1+\sqrt{2}, 3)$ and $\varepsilon < \frac{1}{20}$. □

Last, but not least, we examine request sequence σ_4^{SM}.

Definition 5.21. Let $\varepsilon' > 0$ with $\varepsilon' < \frac{\Theta-1}{2\Theta}$. We define

$$\sigma_4^{\text{SM}} := (\sigma_{1-\varepsilon',\varepsilon'/2}^{\text{lure}}, s_1^{(1)}, s_1^{(2)}, s_2),$$

where $\sigma^{\text{lure}}_{1-\varepsilon',\varepsilon'/2}$ is a request sequence resulting from the application of Lemma 5.6 with $p = 1 - \varepsilon'$ and $\mu = \frac{\varepsilon'}{2}$ and

$$\begin{aligned} s_1^{(1)} &= \left(\frac{\Theta+1}{2\Theta}, 1-\varepsilon'; \frac{1}{\Theta}+\varepsilon'\right), \\ s_1^{(2)} &= \left(\frac{1}{\Theta}, \frac{1}{\Theta}; \frac{1}{\Theta}+\varepsilon'\right), \\ s_2 &= \big(1-\varepsilon', 1-\varepsilon'; 1\big). \end{aligned}$$

Note that the requests $s_1^{(1)}$, $s_1^{(2)}$ and s_2 appear after time $\frac{1-\varepsilon'}{\Theta} + \frac{\varepsilon'}{2}$ and therefore do not interfere with the application of Lemma 5.6. Furthermore, note that we have $a_1^{(1)} = \frac{\Theta+1}{2\Theta} > \frac{1}{\Theta}$ and that $\varepsilon' < \frac{\Theta-1}{2\Theta}$ implies $a_1^{(1)} < 1-\varepsilon'$, i.e., the carrying path of $s_1^{(1)}$ and the area, where Lemma 5.6 is applied are disjoint and the starting position of request $s_1^{(1)}$ is on the left side of position $1-\varepsilon'$. We begin our analysis of σ_4^{SM} with the computation of $\text{OPT}(\sigma_4^{\text{SM}})$.

Lemma 5.22. *We have*

$$\text{OPT}(\sigma_4^{\text{SM}}) = 1.$$

Proof. OPT waits at the origin until time ε' and then performs the walk

$$0 \to 1 - \varepsilon'.$$

OPT's walk is presented in blue in Figure 5.9 for $\Theta = 3.25$. We show that all requests are served this way: Let q be the position of a request that has occurred by the application of Lemma 5.6. Then this requests is released earlier than time $q + \frac{\varepsilon'}{2}$. Since OPT reaches position q not earlier than time $q + \varepsilon' > q + \frac{\varepsilon'}{2}$, OPT can go straight from the origin to position $1-\varepsilon'$ collecting and delivering all requests that occur by the application of Lemma 5.6 as well as, $s_1^{(1)}$, $s_1^{(2)}$ and s_2. Note that $s_1^{(1)}$ can be served on the way since

$$a_1^{(1)} = \frac{\Theta+1}{2\Theta} > \frac{1}{\Theta} = r_1^{(1)} - \varepsilon'.$$

Therefore, we have

$$\text{OPT}(\sigma_4^{\text{SM}}) = \varepsilon' + D\big(0 \to 1+\varepsilon'\big) = 1.$$

Note that OPT can do this even if capacity $c = 1$ holds since no transportation requests need to be carried over $[0, \frac{1-\varepsilon'}{\Theta}] \cup \{1\}$, where the requests of the application of Lemma 5.6 appear. □

Lemma 5.23. *Let* $\Theta \in [3, \infty)$. *Then, we have*

$$\textsc{Smartstart}(\sigma_4^{\text{Sm}}) = \frac{4\Theta - 3}{\Theta} - \frac{9}{2}\varepsilon'.$$

Proof. SMARTSTART's walk is presented in green in Figure 5.9 for $\Theta = 3.25$. Algorithm SMARTSTART reaches position $p_1 = 1 - \varepsilon'$ at time $v_1 = 1 - \frac{\varepsilon'}{2}$. The shortest schedule serving $s_1^{(2)}$ before delivering $s_1^{(1)}$ has length

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

The shortest schedule that serves $s_1^{(2)}$ after delivering $s_1^{(1)}$ has length

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

Therefore SMARTSTART serves $s_1^{(2)}$ after delivering $s_1^{(1)}$ and for all $t \geq v_1$ we have

$$L(t, p_1, (s_1^{(1)}, s_1^{(2)})) = 2 - \frac{2}{\Theta} - 3\varepsilon'.$$

By assumption, we have $\Theta \geq 3$, which implies that for the finishing time $v_1 = 1 - \frac{\varepsilon'}{2}$ of schedule S_1 the inequality

$$w_1 = \frac{2\Theta - 2}{\Theta(\Theta - 1)} - \frac{3\varepsilon'}{\Theta - 1} \overset{\Theta \geq 3}{\leq} \frac{2}{3} - \frac{3}{2}\varepsilon' < 1 - \frac{\varepsilon'}{2} = v_1$$

holds. Thus, the schedule S_1 is started immediately after the application of Lemma 5.6 at time

$$t_1 = \max\{v_1, w_1\} = v_1 = 1 - \frac{\varepsilon'}{2}.$$

SMARTSTART finishes schedule S_1 at time

$$v_2 = t_1 + L(t_1, p_1, (s_1^{(1)}, s_1^{(2)})) = \frac{3\Theta - 2}{\Theta} - \frac{7}{2}\varepsilon'$$

at position $p_2 = \frac{1}{\Theta}$. For all $t \geq v_2$, we have

$$L(t, p_2, (s_2)) = D\left(\frac{1}{\Theta} \to 1 - \varepsilon'\right) = 1 - \frac{1}{\Theta} - \varepsilon'.$$

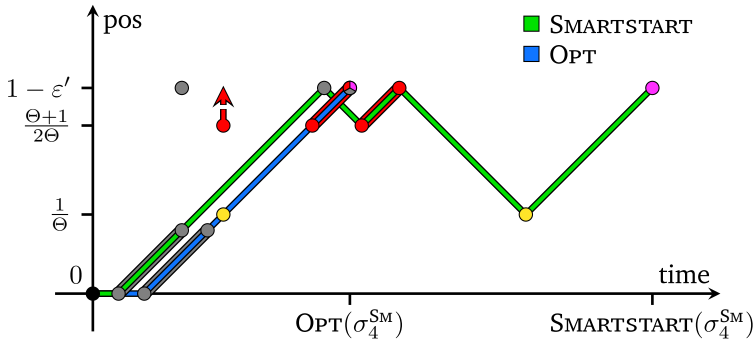

Figure 5.9: SMARTSTART's and OPT's walk serving σ_4^{SM} with $\varepsilon' = 0.2$ and $\Theta = 3.25$. Request $s_1^{(1)}$ is red ●, request $s_1^{(2)}$ is yellow ○ and request s_2 is violet ●. The requests of Lemma 5.6 are gray ●.

By assumption, we have $\Theta \geq 3$, which implies $\varepsilon' < \frac{\Theta-1}{2\Theta} < \frac{6(\Theta-1)^2}{7\Theta^2-9\Theta}$. For the finishing time $v_2 = \frac{3\Theta-2}{\Theta} - \frac{7}{2}\varepsilon'$ of schedule S_1 the inequality

$$\begin{aligned} w_2 &= \frac{1-\frac{1}{\Theta}-\varepsilon'}{\Theta-1} \\ &= \frac{1}{\Theta} - \frac{7}{2}\varepsilon' + \frac{7\Theta-9}{2\Theta-1}\varepsilon' \\ &\overset{\varepsilon' < \frac{6(\Theta-1)^2}{7\Theta^2-9\Theta}}{<} \frac{3\Theta-2}{\Theta} - \frac{7}{2}\varepsilon' \\ &= v_2. \end{aligned}$$

holds. Therefore the final schedule S_2 is started at time

$$t_2 = \max\{v_2, w_2\} = v_2 = \frac{3\Theta-2}{\Theta} - \frac{7}{2}\varepsilon'.$$

To sum it up, we have

$$\text{SMARTSTART}(\sigma_4^{\text{SM}}) = t_2 + L(t_2, p_2, (s_2)) = \frac{4\Theta-3}{\Theta} - \frac{9}{2}\varepsilon'. \qquad \square$$

Equipped with the Lemmas 5.22 and 5.23, we can compute lower bounds for the competitive ratio of SMARTSTART for online DIAL-A-RIDE for $\Theta \in [3, \infty)$.

Lemma 5.24. *Let $\Theta \geq 3$. For every sufficiently small $\varepsilon > 0$, we have*

$$\frac{\text{SMARTSTART}(\sigma_4^{\text{SM}})}{\text{OPT}(\sigma_4^{\text{SM}})} = \frac{4\Theta-3}{\Theta} - \varepsilon =: f_4^{\text{low}}(\Theta) - \varepsilon.$$

In particular, we have

$$\frac{\textsc{Smartstart}(\sigma_4^{\textsc{Sm}})}{\textsc{Opt}(\sigma_4^{\textsc{Sm}})} > \rho_{\textsc{Sm}}^{\mathrm{D},\mathbb{R}} \approx 2.93768$$

for $\Theta \in [3, \infty)$.

Proof. Let $\varepsilon > 0$ with $\varepsilon < \min\{\frac{9}{2}\left(\frac{\Theta-1}{2\Theta}\right), \frac{1}{20}\}$ and $\varepsilon' = \frac{2}{9}\varepsilon < \frac{\Theta-1}{2\Theta}$. By Lemma 5.23, we have

$$\textsc{Smartstart}(\sigma_4^{\textsc{Sm}}) = \frac{4\Theta - 3}{\Theta} - \frac{9}{2}\varepsilon'.$$

By Lemma 5.22, we have

$$\textsc{Opt}(\sigma_4^{\textsc{Sm}}) = 1.$$

We have

$$\frac{\textsc{Smartstart}(\sigma_4^{\textsc{Sm}})}{\textsc{Opt}(\sigma_4^{\textsc{Sm}})} = \frac{4\Theta - 3}{\Theta} - \frac{9}{2}\varepsilon' = \frac{4\Theta - 3}{\Theta} - \varepsilon = f_4^{\text{low}}(\Theta) - \varepsilon.$$

as claimed. The function f_4^{low} monotonically increasing on the interval $[3, \infty)$. Therefore, we have

$$\frac{\textsc{Smartstart}(\sigma_4^{\textsc{Sm}})}{\textsc{Opt}(\sigma_4^{\textsc{Sm}})} \geq f_4^{\text{low}}(3) - \varepsilon = 3 - \varepsilon' > \rho_{\textsc{Sm}}^{\mathrm{D},\mathbb{R}} \approx 2.93768$$

for all $[3, \infty)$ and $\varepsilon < \frac{1}{20}$. $\square$

We combine all lower bounds constructed in this section into one general lower bound. See Figure 5.10 for an illustration of all upper and lower bounds for the competitive ratio of online DIAL-A-RIDE on the line.

Theorem 5.25. *Let* $F_{\text{DaR}} : \mathbb{R}_{>1} \to \mathbb{R}_{>1}$ *be a function with*

$$F_{\text{DaR}}(\Theta) := \begin{cases} f_1^{\text{low}}(\Theta), & \textit{for } \Theta \in (1, 2], \\ f_1^{\text{up}}(\Theta), & \textit{for } \Theta \in (2, \Theta_{\textsc{Sm}}^{\mathrm{D},\mathbb{R}}], \\ f_2^{\text{up}}(\Theta), & \textit{for } \Theta \in (\Theta_{\textsc{Sm}}^{\mathrm{D},\mathbb{R}}, \frac{1}{2}(1+\sqrt{13})], \\ f_2^{\text{low}}(\Theta), & \textit{for } \Theta \in (\frac{1}{2}(1+\sqrt{13}), 1+\sqrt{2}), \\ f_3^{\text{low}}(\Theta), & \textit{for } \Theta \in [1+\sqrt{2}, 3), \\ f_4^{\text{low}}(\Theta), & \textit{for } \Theta \in [3, \infty). \end{cases}$$

Then F_{DaR} *is general lower bound for the competitive ratio of* SMARTSTART *for open online* DIAL-A-RIDE *on the line. The unique minimum of* F_{DaR} *lies in* $\Theta = \Theta_{\textsc{Sm}}^{\mathrm{D},\mathbb{R}} \approx 2.0526$ *and yields a lower bound of*

$$F_{\text{DaR}}(\Theta_{\textsc{Sm}}^{\mathrm{D},\mathbb{R}}) = \rho_{\textsc{Sm}}^{\mathrm{D},\mathbb{R}} \approx 2.9377$$

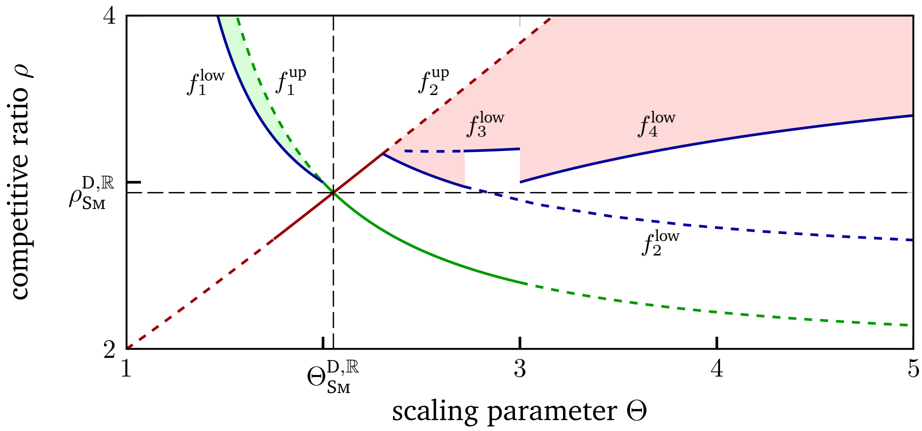

Figure 5.10: Functions f_1^{up} (green) / f_2^{up} (red): upper bounds for competitive ratio for postponing / non-postponing case, drawn solid if tight. Functions: f_1^{low} to f_4^{low} (blue): lower bounds for competitive ratio. Green / red area: possible values for the competitive ratio, bounded by f_1^{up} / f_2^{up} and f_1^{low} to f_4^{low}.

Proof. We have shown in Proposition 5.11 that $f_1^{\mathrm{up}}(\Theta)$ with $\Theta \in (2, \Theta_{\mathrm{SM}}^{\mathrm{D},\mathbb{R}}]$ is a lower bound for the competitive ratio of SMARTSTART for online DIAL-A-RIDE on the line and in Proposition 5.15 that $f_2^{\mathrm{up}}(\Theta)$ with $\Theta \in (\Theta_{\mathrm{SM}}^{\mathrm{D},\mathbb{R}}, \frac{1}{2}(1+\sqrt{13})]$ is a lower bound. Theorem 5.5 implies that F_{DaR} has unique minimum in the interval $(2, \frac{1}{2}(1+\sqrt{13})]$ at $\Theta = \Theta_{\mathrm{SM}}^{\mathrm{D},\mathbb{R}}$. It remains to show that $F_{\mathrm{DaR}}(\Theta) > F_{\mathrm{DaR}}(\Theta_{\mathrm{SM}}^{\mathrm{D},\mathbb{R}})$ for all $\Theta \in (1,2] \cup [\frac{1}{2}(1+\sqrt{13}), \infty)$. This immediately follows from Lemmas 5.10, 5.16, 5.20 and 5.24. $\square$

The main theorem of this section follows by combining Theorem 5.5 and Theorem 5.25.

Theorem 5.26. *The competitive ratio of* SMARTSTART *for open online* DIAL-A-RIDE *on the line with scaling parameter* $\Theta_{\mathrm{SM}}^{\mathrm{D},\mathbb{R}} \approx 2.0526$ *is exactly*

$$\rho_{\mathrm{SM}}^{\mathrm{D},\mathbb{R}} = f_1^{\mathrm{up}}(\Theta_{\mathrm{SM}}^{\mathrm{D},\mathbb{R}}) = f_2^{\mathrm{up}}(\Theta_{\mathrm{SM}}^{\mathrm{D},\mathbb{R}}) \approx 2.9377.$$

For every other $\Theta > 1$ *with* $\Theta \neq \Theta_{\mathrm{SM}}^{\mathrm{D},\mathbb{R}}$ *the competitive ratio of* SMARTSTART *is strictly larger than* $\rho_{\mathrm{SM}}^{\mathrm{D},\mathbb{R}}$.

5.3 Bounds for Open Online TSP on the Line

After we have thoroughly analyzed SMARTSTART for online DIAL-A-RIDE on the line, we now examine the algorithm for online TSP on the line. First we notice, that, since online TSP is a special case of online Dial-a-Ride, all upper bounds, i.e., the bounds provided by Proposition 5.3 and Proposition 5.4 for the competitive ratio of Smartstart for online Dial-a-Ride on the line are also valid for online TSP on the line. However, of the lower bounds, only the bounds obtained by the request sequences without transportation requests are valid for online TSP. To be more precise, only the bounds given by the request sequence σ_2^{SM} are valid, while the bounds given by σ_1^{SM}, σ_3^{SM} and σ_4^{SM} are not. Therefore, we have a lower bound of $f_2^{\text{up}}(\Theta)$ for $\Theta \in [\frac{7}{4}, \frac{1}{2}(1+\sqrt{13})]$ that tightly matches the upper bound provided by Proposition 5.4 for the case that the final schedule is not postponed and a lower bound of $f_2^{\text{low}}(\Theta)$ for $\Theta \in (\frac{1}{2}(1+\sqrt{13}), 4]$. We will see that the upper bound given in Proposition 5.3 for the case that the final schedule is postponed is not tight for online TSP. The reason for this is that online TSP allows a smaller bound for the length of a schedule.

Lemma 5.27. *For every schedule S_j of* Smartstart *for online* TSP*, we have*

$$L(t_j, p_j, \sigma_j) \leq \left(1 + \frac{\Theta}{2\Theta + 3}\right)\textsc{Opt}(\sigma).$$

Proof. By Lemma 4.6 and Lemma 5.1, we have

$$\begin{aligned}
L(t_j, p_j, \sigma_j) &\overset{\text{Lem 4.6}}{\leq} \min\left\{\textsc{Opt}(\sigma) + d(p_j, 0), \frac{3}{2}(\textsc{Opt}(\sigma) - t_{j-1})\right\} \\
&\overset{\text{Lem 5.1}}{\leq} \min\left\{\textsc{Opt}(\sigma) + d(p_j, 0), \frac{3}{2}\left(\textsc{Opt}(\sigma) - \frac{1}{\Theta}d(p_j, 0)\right)\right\} \\
&\leq \left(1 + \frac{\Theta}{2\Theta + 3}\right)\textsc{Opt}(\sigma)
\end{aligned}$$

since the minimum above is largest if the two terms are equal, which is the case for $d(p_j, 0) = \frac{\Theta}{2\Theta+3}\textsc{Opt}(\sigma)$. □

With the result of Lemma 5.27 we can improve the upper bound of Proposition 5.3.

Proposition 5.28. *In the case that* Smartstart *for online* TSP *postpones executing S_N, we have*

$$\frac{\textsc{Smartstart}(\sigma)}{\textsc{Opt}(\sigma)} \leq \frac{3\Theta^2 + 3\Theta}{2\Theta^2 + \Theta - 3} =: f_{1,\text{TSP}}^{\text{up}}(\Theta).$$

Proof. Assume Smartstart postpones the final schedule. Then we have

$$\textsc{Smartstart}(\sigma) \overset{(4.1)}{=} t_N + L(t_N, p_N, \sigma_N) \overset{(5.2)}{=} \frac{\Theta}{\Theta - 1}L(t_N, p_N, \sigma_N). \tag{5.27}$$

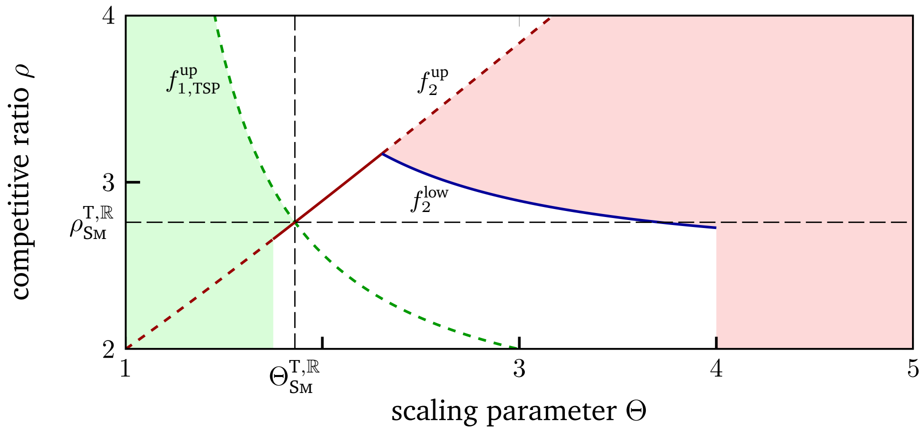

Figure 5.11: Functions $f_{1,\mathrm{TSP}}^{\mathrm{up}}$ (green) / f_2^{up} (red): upper bounds for competitive ratio for postponing / non-postponing case, drawn solid if tight. Function: f_2^{low} (blue): lower bound for competitive ratio. Green / red area: possible values for the competitive ratio, bounded by $f_{1,\mathrm{TSP}}^{\mathrm{up}}$ / f_2^{up} and f_2^{low}.

Lemma 5.27 thus yields the claimed bound:

$$\begin{aligned}\textsc{SmartStart}(\sigma) &\overset{(5.27)}{=} \frac{\Theta}{\Theta-1}L(t_N,p_N,\sigma_N)\\ &\overset{\text{Lem 5.27}}{\leq} \frac{\Theta}{\Theta-1}\left(1+\frac{\Theta}{2\Theta+3}\right)\textsc{Opt}(\sigma)\\ &= \frac{3\Theta^2+3\Theta}{2\Theta^2+\Theta-3}\textsc{Opt}(\sigma). \qquad \square\end{aligned}$$

We combine the results of Proposition 5.28 and Proposition 5.4 to obtain a general upper bound for the competitive ratio of SmartStart for online Dial-a-Ride on the line.

Theorem 5.29. *The function* $\max\{f_{1,\mathrm{TSP}}^{\mathrm{up}}, f_2^{\mathrm{up}}\}$ *gives an upper bound for the competitive ratio of* SmartStart *for open online* TSP *on the line for all* $\Theta > 1$. *Let* $\Theta_{\mathrm{SM}}^{\mathrm{T},\mathbb{R}} \approx 1.8607$ *be the unique solution of the equation* $f_{1,\mathrm{TSP}}^{\mathrm{up}}(\Theta) = f_2^{\mathrm{up}}(\Theta)$, *i.e., of*

$$\frac{3\Theta^2+3\Theta}{2\Theta^2+\Theta-3} = \frac{3\Theta^2+5\Theta+4}{3\Theta+3},$$

in the interval $(1,\infty)$. *Then,* $\Theta_{\mathrm{SM}}^{\mathrm{T},\mathbb{R}}$ *is the unique minimum of the function* $\max\{f_{1,\mathrm{TSP}}^{\mathrm{up}}, f_2^{\mathrm{up}}\}$ *and* SmartStart *with scaling parameter* $\Theta_{\mathrm{SM}}^{\mathrm{T},\mathbb{R}}$ *is* $\rho_{\mathrm{SM}}^{\mathrm{T},\mathbb{R}}$*-competitive with*

$$\rho_{\mathrm{SM}}^{\mathrm{T},\mathbb{R}} = f_{1,\mathrm{TSP}}^{\mathrm{up}}(\Theta_{\mathrm{SM}}^{\mathrm{T},\mathbb{R}}) = f_2^{\mathrm{up}}(\Theta_{\mathrm{SM}}^{\mathrm{T},\mathbb{R}}) \approx 2.7604.$$

Proof. For the case where SMARTSTART postpones the final schedule, we have established the upper bound

$$\frac{\text{SMARTSTART}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{3\Theta^2 + 3\Theta}{2\Theta^2 + \Theta - 3} = f_{1,\text{TSP}}^{\text{up}}(\Theta)$$

in Proposition 5.28 and for the case where SMARTSTART does not postpone final schedule, we have established the upper bound

$$\frac{\text{SMARTSTART}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{3\Theta^2 + 5\Theta + 4}{3\Theta + 3} = f_2^{\text{up}}(\Theta)$$

in Proposition 5.4. Thus, the maximum of both bounds is a general upper bound for the competitive ratio of SMARTSTART for open online TSP on the line that is independent of SMARTSTART's behavior before the final schedule.

Function $f_{1,\text{TSP}}^{\text{up}}$ is strictly decreasing for $\Theta > 1$ and function f_2^{up} is strictly increasing for $\Theta > 1$. Therefore the minimum of $\max\{f_{1,\text{TSP}}^{\text{up}}, f_2^{\text{up}}\}$ in the interval $(1, \infty)$ lies in the intersection point of $f_{1,\text{TSP}}^{\text{up}}$ and f_2^{up}, i.e., in $\Theta_{\text{SM}}^{\text{T},\mathbb{R}} \approx 1.8607$. The resulting upper bound for the competitive ratio is

$$\rho_{\text{SM}}^{\text{T},\mathbb{R}} = f_{1,\text{TSP}}^{\text{up}}(\Theta_{\text{SM}}^{\text{T},\mathbb{R}}) = f_2^{\text{up}}(\Theta_{\text{SM}}^{\text{T},\mathbb{R}}) \approx 2.7604. \qquad \square$$

See Figure 5.11 for a visualization of the upper bound for the competitive ratio of online DIAL-A-RIDE on the line presented in Theorem 5.5 together with the lower bound $f_2^{\text{low}}(\Theta)$.

In the following we will present two request sequences σ_5^{SM} and σ_6^{SM}. We will see that the ratio of SMARTSTART's and OPT's completion time of σ_5^{SM} tightly matches the upper bound of Proposition 5.28 for $\Theta \in (1, 4)$. The request sequence σ_6^{SM} will provide an additional lower bound for the competitive ratio of SMARTSTART for open online TSP on the line for larger values of Θ. But first, we take another look at the request sequence σ_2^{SM}. Since the upper bound for the competitive ratio $\rho_{\text{SM}}^{\text{T},\mathbb{R}}$ of SMARTSTART for the TSP version is slightly lower than the competitive ratio $\rho_{\text{SM}}^{\text{D},\mathbb{R}}$ of the DIAL-A-RIDE version, the lower bound $f_2^{\text{low}}(\Theta)$ provided by the request sequence σ_2^{SM} is useful for a slightly larger interval in the TSP version than in the DIAL-A-RIDE version.

Lemma 5.30. *We have*

$$\frac{\text{SMARTSTART}(\sigma_2^{\text{SM}})}{\text{OPT}(\sigma_2^{\text{SM}})} > \rho_{\text{SM}}^{\text{T},\mathbb{R}} \approx 2.7604$$

for $\Theta \in (\frac{1}{2}(1 + \sqrt{13}, \frac{7}{2}] \approx (2.303, 3.5]$.

Proof. According to Lemma 5.16, we have

$$\frac{\textsc{Smartstart}(\sigma_2^{\textsc{Sm}})}{\textsc{Opt}(\sigma_2^{\textsc{Sm}})} = \frac{5\Theta^2 + 3\Theta - 2}{2\Theta^2 + \Theta - 3} - \varepsilon = f_2^{\text{low}}(\Theta) - \varepsilon$$

for $\Theta \in (\frac{1}{2}(1 + \sqrt{13}), 4]$ and sufficiently small ε. Let $\varepsilon < \frac{1}{50}$. The function f_2^{low} is monotonically decreasing on $(\frac{1}{2}(1 + \sqrt{13}), \frac{7}{2}]$. Therefore, we have

$$\frac{\textsc{Smartstart}(\sigma_2^{\textsc{Sm}})}{\textsc{Opt}(\sigma_2^{\textsc{Sm}})} \geq f_2^{\text{low}}\left(\frac{7}{2}\right) - \varepsilon > 2.79 - \varepsilon' > \rho_{\textsc{Sm}}^{\mathrm{T},\mathbb{R}} \approx 2.7604$$

for all $(\frac{1}{2}(1 + \sqrt{13}), \frac{7}{2}]$ and $\varepsilon < \frac{1}{50}$. $\square$

We define the request sequence $\sigma_5^{\textsc{Sm}}$.

Definition 5.31. Let $\varepsilon' > 0$ with $\varepsilon' < \frac{\Theta+1}{\Theta}$. We define

$$\sigma_5^{\textsc{Sm}} := (\sigma_{1,\varepsilon'/2}^{\text{lure}}, s_1^{(1)}, s_1^{(2)}),$$

where $\sigma_{1,\varepsilon'/2}^{\text{lure}}$ is a subsequence of requests resulting from the application of Lemma 5.6 with $p = 1$ and $\mu = \frac{\varepsilon'}{2}$ and

$$\begin{aligned} s_1^{(1)} &= \left(\frac{2\Theta + 1}{\Theta} - \varepsilon'; \frac{1}{\Theta} + \varepsilon'\right), \\ s_1^{(2)} &= \left(-\frac{1}{\Theta}; \frac{1}{\Theta} + \varepsilon'\right). \end{aligned}$$

Note that the requests $s_1^{(1)}$ and $s_1^{(2)}$ appear after time $\frac{1}{\Theta} + \frac{\varepsilon'}{2}$ and therefore do not interfere with the application of Lemma 5.6. Furthermore, note that $\varepsilon' < \frac{\Theta+1}{\Theta}$ implies $a_1^{(1)} > 1$, i.e., the position of $s_1^{(1)}$ is on the right side of position 1. We begin our analysis of $\sigma_5^{\textsc{Sm}}$ with the computation of $\textsc{Opt}(\sigma_5^{\textsc{Sm}})$.

Lemma 5.32. *We have*

$$\textsc{Opt}(\sigma_5^{\textsc{Sm}}) = \frac{2\Theta + 3}{\Theta}.$$

Proof. $\textsc{Opt}$ waits at the origin until time ε' and then performs the walk

$$0 \to -\frac{1}{\Theta} \to \frac{2\Theta + 1}{\Theta} - \varepsilon'.$$

OPT's walk is shown in blue in Figure 5.12 for $\Theta = \Theta_{\text{SM}}^{\text{T},\mathbb{R}}$. We show that all requests are served this way: OPT collects $s_1^{(2)}$ at time $\frac{1}{\Theta} + \varepsilon'$ and returns to the origin at time $\frac{2}{\Theta} + \varepsilon'$. Let q be the position of a request that has occurred by the application of Lemma 5.6. Then this requests is released earlier than time $q + \frac{\varepsilon'}{2}$. Since OPT reaches position q not earlier than time $q + \frac{2}{\Theta} + \varepsilon' > q + \frac{\varepsilon'}{2}$, OPT can go straight from the origin to position $\frac{2\Theta+1}{\Theta} - \varepsilon'$ serving all requests that occur by the application of Lemma 5.6 as well as $s_1^{(1)}$. Therefore, we have

$$\text{OPT}(\sigma_5^{\text{SM}}) = \varepsilon' + D\left(0 \to -\frac{1}{\Theta} \to \frac{2\Theta+1}{\Theta} - \varepsilon'\right) = \frac{2\Theta+3}{\Theta}. \qquad \square$$

Next, we compute SMARTSTART's completion time.

Lemma 5.33. *Let $\Theta \leq 4$ and $\varepsilon' < \frac{\Theta-1}{2\Theta+6}$. Then, we have*

$$\text{SMARTSTART}(\sigma_5^{\text{SM}}) = \frac{3\Theta+3}{\Theta-1} - \frac{2\Theta}{\Theta-1}\varepsilon'.$$

Proof. SMARTSTART's walk is shown in green in Figure 5.12 for $\Theta = SMTL$. SMARTSTART reaches position $p_1 = 1$ at time $v_1 = 1 + \frac{\varepsilon'}{2}$. The shortest schedule serving $s_1^{(2)}$ before serving $s_1^{(1)}$ has length

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

The shortest schedule that serves $s_1^{(2)}$ after serving $s_1^{(1)}$ has length

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

Thus, SMARTSTART serves $s_1^{(2)}$ after serving $s_1^{(1)}$, and, for all $t \geq v_1$, we obtain

$$L\left(t, p_1, (s_1^{(1)}, s_1^{(2)})\right) = 3 + \frac{3}{\Theta} - 2\varepsilon'.$$

By assumption, we have $\Theta \leq 4$ and $\varepsilon' < \frac{\Theta-1}{2\Theta+6}$, which implies that for the time $v_1 = 1 + \frac{\varepsilon'}{2}$, when SMARTSTART reaches position $p_1 = 1$, the inequality

$$\begin{aligned} w_1 &= \frac{3\Theta+3}{\Theta(\Theta-1)} - \frac{2}{\Theta-1}\varepsilon' \\ &\overset{\Theta \leq 4}{\geq} \frac{5}{4} - \frac{2}{\Theta-1}\varepsilon' \end{aligned}$$

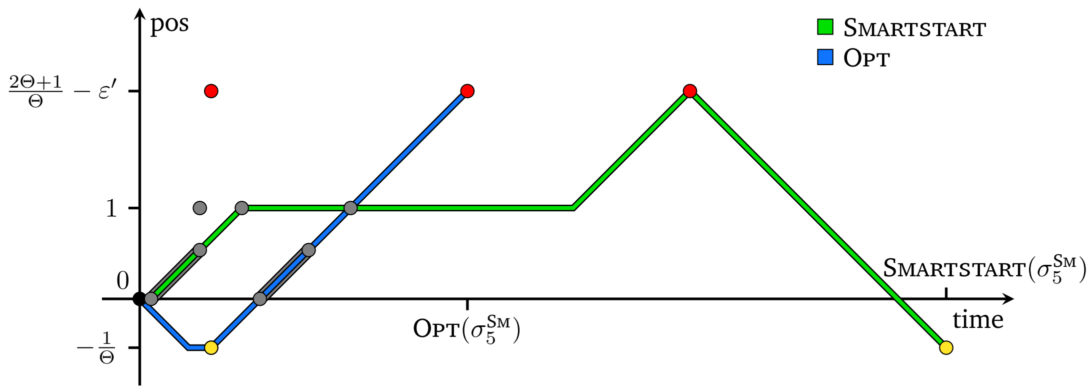

Figure 5.12: SMARTSTART's and OPT's walk serving σ_5^{SM} with $\varepsilon' = 0.25$ and $\Theta = \Theta_{\text{SM}}^{\text{T},\mathbb{R}}$. Request $s_1^{(1)}$ is red and request $s_1^{(2)}$ is yellow. The requests of Lemma 5.6 are gray.

$$\begin{aligned} &= \quad \frac{5}{4} - \frac{\Theta+3}{2\Theta-2}\varepsilon' + \frac{\varepsilon'}{2} \\ &\overset{\varepsilon' < \frac{\Theta-1}{2\Theta+6}}{>} \quad 1 + \frac{\varepsilon'}{2} \\ &= \quad v_1 \end{aligned} \tag{5.28}$$

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

$$t_1 = \max\{v_1, w_1\} \overset{(5.28)}{=} w_1 = \frac{3\Theta+3}{\Theta(\Theta-1)} - \frac{2}{\Theta-1}\varepsilon'.$$

To sum it up, we have

$$\text{SMARTSTART}(\sigma_5^{\text{SM}}) = t_1 + L(t_1, p_1, (s_1^{(1)}, s_1^{(2)})) = \frac{3\Theta+3}{\Theta-1} - \frac{2\Theta}{\Theta-1}\varepsilon'. \qquad \square$$

The following proposition shows that in the case $\Theta \in (1, 4]$ the ratio of SMARTSTART 's and OPT 's completion time for the request sequence σ_5^{SM} tightly matches the upper bound provided by Proposition 5.28.

Proposition 5.34. *Let $1 < \Theta \leq 4$. For every sufficiently small $\varepsilon > 0$, we have*

$$\frac{\text{SMARTSTART}(\sigma_5^{\text{SM}})}{\text{OPT}(\sigma_5^{\text{SM}})} \geq \frac{3\Theta^2+5\Theta+4}{3\Theta+3} - \varepsilon = f_{1,\text{TSP}}^{\text{up}}(\Theta) - \varepsilon$$

and SMARTSTART *postpones the final schedule, i.e., the upper bound established in Proposition 5.28 is tight for $\Theta \in (1, 4]$.*

Proof. Let $\varepsilon > 0$ with $\varepsilon < \frac{\Theta-1}{2\Theta+6}\left(\frac{2\Theta^2}{2\Theta^2+\Theta-3}\right)$ and $\varepsilon' = \frac{2\Theta^2+\Theta-3}{2\Theta^2}\varepsilon < \frac{\Theta-1}{2\Theta+6}$. Lemma 5.33 implies

$$\textsc{Smartstart}(\sigma_5^{\textsc{Sm}}) = \frac{3\Theta+3}{\Theta-1} - \frac{2\Theta}{\Theta-1}\varepsilon'.$$

By Lemma 5.32, we have

$$\textsc{Opt}(\sigma_5^{\textsc{Sm}}) = \frac{2\Theta+3}{\Theta}.$$

Since we have $\varepsilon' = \frac{2\Theta^2+\Theta-3}{2\Theta^2}\varepsilon$, we obtain

$$\frac{\textsc{Smartstart}(\sigma_5^{\textsc{Sm}})}{\textsc{Opt}(\sigma_5^{\textsc{Sm}})} = \frac{3\Theta^2+3\Theta}{2\Theta^2+\Theta-3} - \frac{2\Theta^2}{2\Theta^2+\Theta-3}\varepsilon' = \frac{3\Theta^2+3\Theta}{2\Theta^2+\Theta-3} - \varepsilon = f_{1,\text{TSP}}^{\text{up}}(\Theta) - \varepsilon.$$

□

Recall that the optimal parameter $\Theta_{\text{SM}}^{\text{T},\mathbb{R}}$ established in Theorem 5.29 is the only positive, real solution of the equation

$$\frac{3\Theta^2+3\Theta}{2\Theta^2+\Theta-3} = \frac{3\Theta^2+5\Theta+4}{3\Theta+3},$$

which is $\Theta_{\text{SM}}^{\text{T},\mathbb{R}} \approx 1.8607$. Therefore, by Proposition 5.34 and Proposition 5.15 the parameter $\Theta_{\text{SM}}^{\text{T},\mathbb{R}}$ lies in the interval where the upper bounds of Propositions 5.28 and 5.4 are both tight. Moreover, by Propositions 5.34 and 5.15 and by Lemma 5.30, there is no $\Theta \in (1, \frac{7}{2}]\setminus\{\Theta_{\text{SM}}^{\text{T},\mathbb{R}}\}$ that yields an equal or better competitive ratio than $\Theta_{\text{SM}}^{\text{T},\mathbb{R}}$ does. Therefore, it remains to make sure that there is no $\Theta > \frac{7}{2}$ that yields an equal or better competitive ratio than $\rho_{\text{SM}}^{\text{T},\mathbb{R}} \approx 2.76037$. For this we introduce the final request sequence $\sigma_6^{\textsc{Sm}}$. Figure 5.13 illustrates the upper and lower bounds for the compeititive ratio of Smartstart for online TSP on the line including the result of Proposition 5.34.

Definition 5.35. Let $\Theta > 3$ and $\varepsilon' > 0$ with $\varepsilon' < \min\{1, \frac{4}{\Theta}\}$. We define

$$\sigma_6^{\textsc{Sm}} := (\sigma_{1,\varepsilon'/4}^{\text{lure}}, s_1^{(1)}, s_1^{(2)}, s_2^{(1)}, s_2^{(2)}, s_3),$$

where $\sigma_{1,\varepsilon'/4}^{\text{lure}}$ is a subsequence of requests resulting from the application of Lemma 5.6 with $p = 1$ and $\mu = \frac{\varepsilon'}{4}$ and

$$s_1^{(1)} = \left(\max\left\{\frac{\Theta+1}{2}, \frac{\Theta^2-2\Theta+2}{\Theta}\right\}; \frac{1}{\Theta} + \frac{\varepsilon'}{2}\right),$$

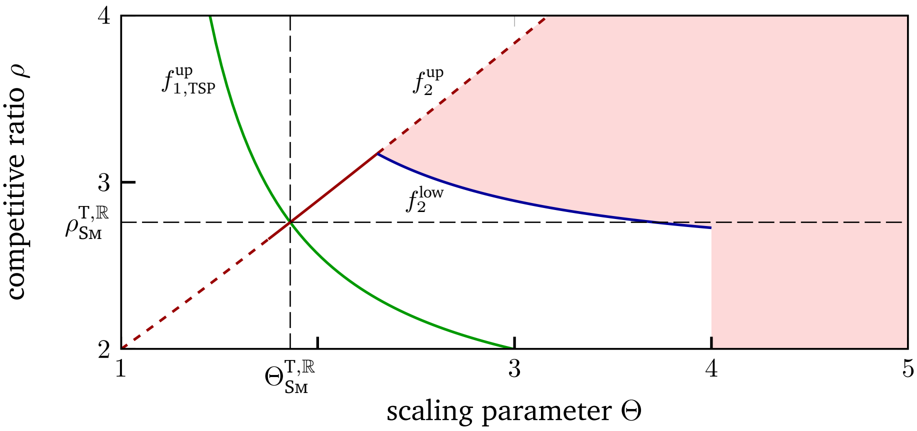

Figure 5.13: Functions $f_{1,\mathrm{TSP}}^{\mathrm{up}}$ (green) / f_2^{up} (red): upper bounds for competitive ratio for postponing / non-postponing case, drawn solid if tight. Function: f_2^{low} (blue): lower bound for competitive ratio. Green / red area: possible values for the competitive ratio, bounded by $f_{1,\mathrm{TSP}}^{\mathrm{up}}$ / f_2^{up} and f_2^{low}.

$$\begin{aligned}
s_1^{(2)} &= \left(\max\left\{\frac{5-\Theta}{4}, \frac{1}{\Theta}\right\} + \frac{\varepsilon'}{2}; \frac{1}{\Theta} + \frac{\varepsilon'}{2}\right), \\
s_2^{(1)} &= (1; 1+\varepsilon') \\
s_2^{(2)} &= \left(\max\left\{\Theta, \frac{2\Theta^2 - 5\Theta + 4}{\Theta}\right\} - \frac{\Theta - 1}{2}\varepsilon'; 1+\varepsilon'\right) \\
s_3 &= \left(\max\left\{\Theta, \frac{2\Theta^2 - 5\Theta + 4}{\Theta}\right\} - \varepsilon'; \Theta\right)
\end{aligned}$$

Note that the requests $s_1^{(1)}$, $s_1^{(2)}$, $s_2^{(1)}$, $s_2^{(2)}$ and s_3 appear after time $\frac{1}{\Theta} + \frac{\varepsilon'}{4}$ and therefore do not interfere with the application of Lemma 5.6. Furthermore, note that $\Theta > 3$ implies $a_3 > a_2^{(2)}$, i.e., the position of s_3 is to the right of the position of $s_2^{(2)}$ and

$$\varepsilon' < \min\left\{1, \frac{4}{\Theta}\right\} < \min\left\{\frac{\Theta - 1}{2}, \frac{2\Theta - 2}{\Theta}\right\}$$

implies

$$a_2^{(2)} \geq \Theta - \frac{\Theta - 1}{2}\varepsilon' \overset{\varepsilon' < \min\{1, \frac{4}{\Theta}\}}{>} a_1^{(1)} \overset{\Theta > 3}{>} 1 = a_2^{(1)} \overset{\varepsilon' < \min\{\frac{\Theta-1}{2}, \frac{2\Theta-2}{\Theta}\}}{>} = a_1^{(2)} > 0,$$

i.e., the position of s_3 is to the right of the position of $s_1^{(1)}$, which is to the right of the position of $s_2^{(1)}$ position, which is to the right of the position of $s_1^{(2)}$. Therefore, s_3 is the rightmost request. We begin our analysis of σ_6^{SM} with the computation of $\textsc{Opt}(\sigma_6^{\text{SM}})$.

Lemma 5.36. *We have*

$$\textsc{Opt}(\sigma_6^{\text{SM}}) = \max\left\{\Theta, \frac{2\Theta^2 - 5\Theta + 4}{\Theta}\right\}.$$

Proof. First we note that all position of the requests of σ_6^{SM} are to the right of the origin and to the left of a_3. $\textsc{Opt}$ waits at the origin until time ε' and then performs the walk

$$0 \to \max\left\{\Theta, \frac{2\Theta^2 - 5\Theta + 4}{\Theta}\right\} - \varepsilon'.$$

$\textsc{Opt}$'s walk is presented in blue in Figure 5.14 for $\Theta = 3.75$ and in blue in Figure 5.15 for $\Theta = 4.25$. We show that all requests are served this way: We need to show that the release time of every request is not more than ε' larger than their position. This is clear for $s_1^{(2)}$ and $s_2^{(1)}$. Let $q > 0$ be the position of a request that has occurred by the application of Lemma 5.6. Then this request is released earlier than time $q + \frac{\varepsilon'}{2}$ and $\textsc{Opt}$ reaches position q at time $q + \varepsilon'$. For $s_1^{(1)}$, we have

$$a_1^{(1)} = \max\left\{\frac{\Theta+1}{2}, \frac{\Theta^2 - 2\Theta + 2}{\Theta}\right\} \overset{\Theta>3}{>} \frac{1}{\Theta} = r_1^{(1)} - \varepsilon'.$$

Because of $\Theta > 3$, we have $\varepsilon' < 1 < \max\{2, \frac{\Theta^2 - 9\Theta + 16}{\Theta}\}$. For $s_2^{(2)}$, we obtain

$$a_2^{(2)} = \max\left\{\Theta, \frac{2\Theta^2 - 5\Theta + 4}{\Theta}\right\} - \frac{\Theta - 1}{2}\varepsilon' \overset{\varepsilon' < \max\{2, \frac{\Theta^2 - 9\Theta + 16}{\Theta}\}}{>} 1 = r_2^{(2)} - \varepsilon'.$$

It remains to examine the request s_3. We have

$$a_3 = \max\left\{\Theta, \frac{2\Theta^2 - 5\Theta + 4}{\Theta}\right\} - \varepsilon' \geq \Theta - \varepsilon' = r_3 - \varepsilon',$$

which proves the claim. Thus, in total, we have

$$\textsc{Opt}(\sigma_6^{\text{SM}}) = \varepsilon' + D\left(0 \to \max\left\{\Theta, \frac{2\Theta^2 - 5\Theta + 4}{\Theta}\right\} - \varepsilon'\right) = \max\left\{\Theta, \frac{2\Theta^2 - 5\Theta + 4}{\Theta}\right\}. \quad \square$$

Next, we examine $\textsc{Smartstart}$'s completion time.

Lemma 5.37. *Let* $\Theta \in (\frac{7}{2}, \infty)$. *Then, we have*

$$\textsc{SmartStart}(\sigma_6^{\mathrm{SM}}) = \max\left\{\frac{7\Theta - 5}{2}, \frac{6\Theta^2 - 15\Theta + 10}{\Theta}\right\} - \left(\Theta - \frac{11}{4}\right)\varepsilon'$$

Proof. SMARTSTART's walk serving σ_6^{SM} is presented in green in Figure 5.14 for $\Theta = 3.75$ and in green in Figure 5.15 for $\Theta = 4.25$. First, we note that

$$a_1^{(1)} = \max\left\{\frac{\Theta+1}{2}, \frac{\Theta^2 - 2\Theta + 2}{\Theta}\right\} = \frac{\Theta+1}{2}$$

holds if and only if we have

$$a_1^{(2)} - \frac{\varepsilon'}{2} = \max\left\{\frac{5-\Theta}{4}, \frac{1}{\Theta}\right\} = \frac{5-\Theta}{4}$$

and if and only if

$$a_1^{(2)} - \frac{\Theta-1}{2}\varepsilon' = a_3 - \varepsilon' = \max\left\{\Theta, \frac{2\Theta^2 - 5\Theta + 4}{\Theta}\right\} = \Theta.$$

SMARTSTART reaches position $p_1 = 1$ at time $v_1 = 1 + \frac{\varepsilon'}{4}$. The shortest schedule serving $s_1^{(2)}$ before serving $s_1^{(1)}$ has length

$$D\left(1 \to \max\left\{\frac{5-\Theta}{4}, \frac{1}{\Theta}\right\} + \frac{\varepsilon'}{2} \to \max\left\{\frac{\Theta+1}{2}, \frac{\Theta^2 - 2\Theta + 2}{\Theta}\right\}\right) = \Theta - 1 - \varepsilon'.$$

On the other hand, the shortest schedule that serves $s_1^{(1)}$ before serving $s_1^{(2)}$ has length

$$\begin{aligned} & D\left(1 \to \max\left\{\frac{\Theta+1}{2}, \frac{\Theta^2 - 2\Theta + 2}{\Theta}\right\} \to \max\left\{\frac{5-\Theta}{4}, \frac{1}{\Theta}\right\} + \frac{\varepsilon'}{2}\right) \\ = \;& \max\left\{\frac{5\Theta - 5}{4}, \frac{2\Theta^2 - 5\Theta + 3}{\Theta}\right\} - \frac{\varepsilon'}{2} \\ \overset{\Theta > \frac{7}{2}}{>} \;& \Theta - 1 - \varepsilon'. \end{aligned}$$

Therefore, SMARTSTART serves $s_1^{(2)}$ before serving $s_1^{(1)}$ and for all $t \geq v_1$, we have

$$L(t, p_1, (s_1^{(1)}, s_1^{(2)})) = \Theta - 1 - \varepsilon'$$

and thus

$$w_1 = 1 - \frac{\varepsilon'}{\Theta - 1} < 1 + \frac{\varepsilon'}{4} = v_1. \tag{5.29}$$

SMARTSTART starts the schedule S_1 at time

$$t_1 = \max\{v_1, w_1\} \overset{(5.29)}{=} v_1 = 1 + \frac{\varepsilon'}{4},$$

i.e., directly after the arrival at position $p_1 = 1$ and before $s_2^{(1)}$ and $s_2^{(2)}$ are released. SMARTSTART reaches position $p_2 = \max\left\{\frac{\Theta+1}{2}, \frac{\Theta^2-2\Theta+2}{\Theta}\right\}$ at time $v_2 = \Theta - \frac{3}{4}\varepsilon'$. The shortest schedule serving $s_2^{(1)}$ before serving $s_2^{(2)}$ has length

$$\begin{aligned} & D\left(\max\left\{\frac{\Theta+1}{2}, \frac{\Theta^2-2\Theta+2}{\Theta}\right\} \to 1 \to \max\left\{\Theta, \frac{2\Theta^2-5\Theta+4}{\Theta}\right\} - \tfrac{\Theta-1}{2}\varepsilon'\right) \\ = \ & \max\left\{\frac{3\Theta-3}{2}, \frac{3\Theta^2-9\Theta+6}{\Theta}\right\} - \frac{(\Theta-1)\varepsilon'}{2}. \end{aligned}$$

On the other hand, the shortest schedule that serves $s_2^{(2)}$ before serving $s_2^{(1)}$ has length

$$\begin{aligned} & D\left(\max\left\{\frac{\Theta+1}{2}, \frac{\Theta^2-2\Theta+2}{\Theta}\right\} \to \max\left\{\Theta, \frac{2\Theta^2-5\Theta+4}{\Theta}\right\} - \frac{\Theta-1}{2}\varepsilon' \to 1\right) \\ = \ & \max\left\{\frac{3\Theta-3}{2}, \frac{3\Theta^2-9\Theta+6}{\Theta}\right\} - (\Theta-1)\varepsilon'. \end{aligned}$$

Therefore, SMARTSTART serves $s_2^{(2)}$ before serving $s_2^{(1)}$ and for all $t \geq v_2$, we have

$$L(t, p_2, (s_2^{(1)}, s_2^{(2)})) = \max\left\{\frac{3\Theta-3}{2}, \frac{3\Theta^2-9\Theta+6}{\Theta}\right\} - (\Theta-1)\varepsilon'.$$

For the time $v_2 = \Theta - \frac{3}{4}\varepsilon'$ when SMARTSTART reaches position $p_2 = \max\left\{\frac{\Theta+1}{2}, \frac{\Theta^2-2\Theta+2}{\Theta}\right\}$, the inequality

$$w_2 = \max\left\{\frac{3}{2}, \frac{3\Theta^2-9\Theta+6}{\Theta^2-\Theta}\right\} - \varepsilon' \overset{\Theta>\frac{7}{2}}{<} \Theta - \frac{3}{4}\varepsilon' = v_2 \tag{5.30}$$

holds. Thus, SMARTSTART starts the schedule S_2 directly after finishing schedule S_1 at time

$$t_2 = \max\{v_2, w_2\} \overset{(5.30)}{=} v_2 = \Theta - \frac{3}{4}\varepsilon',$$

which is before the final request s_3 is released. Schedule S_2 ends in position $p_3 = 1$ at time

$$v_3 = t_2 + L(t_2, p_2, (s_2^{(1)}, s_2^{(2)})) = \max\left\{\frac{5\Theta - 3}{2}, \frac{4\Theta^2 - 9\Theta + 6}{\Theta}\right\} - \left(\Theta - \frac{7}{4}\right)\varepsilon'.$$

It remains to serve request s_3 starting from position $p_3 = 1$. For all $t \geq v_3$, we have

$$\begin{aligned} L(t, p_3, (s_3)) &= D\left(1 \to \max\left\{\Theta, \frac{2\Theta^2 - 5\Theta + 4}{\Theta}\right\} - \varepsilon'\right) \\ &= \max\left\{\Theta - 1, \frac{2\Theta^2 - 6\Theta + 4}{\Theta}\right\} - \varepsilon'. \end{aligned}$$

By assumption, we have $\Theta > \frac{7}{2}$ and thus $\varepsilon' < 1 < \frac{12\Theta^3 - 36\Theta^2 + 40\Theta - 16}{4\Theta^2 - 11\Theta + 3}$. For the case $\max\left\{\Theta - 1, \frac{2\Theta^2 - 6\Theta + 4}{\Theta}\right\} = \Theta - 1$, we get

$$\begin{aligned} w_3 &= 1 - \frac{\varepsilon'}{\Theta - 1} \\ &= 1 + \frac{4\Theta^2 - 11\Theta + 3}{4\Theta - 4}\varepsilon' - \left(\Theta - \frac{7}{4}\right)\varepsilon' \\ &\overset{\varepsilon' < \frac{12\Theta^3 - 36\Theta^2 + 40\Theta - 16}{4\Theta^2 - 11\Theta + 3}}{<} \frac{3\Theta^2 - 5\Theta + 4}{\Theta} - \left(\Theta - \frac{7}{4}\right)\varepsilon' \\ &< v_3. \end{aligned} \tag{5.31}$$

Again by assumption, we have $\Theta > \frac{7}{2}$ and thus $\varepsilon' < 1 < \frac{12\Theta^3 - 40\Theta^2 + 60\Theta - 32}{4\Theta^2 - 11\Theta + 3}$. For the case $\max\left\{\Theta - 1, \frac{2\Theta^2 - 6\Theta + 4}{\Theta}\right\} = \frac{2\Theta^2 - 6\Theta + 4}{\Theta}$, we get

$$\begin{aligned} w_3 &= \frac{2\Theta^2 - 6\Theta + 4}{\Theta} - \frac{\varepsilon'}{\Theta - 1} \\ &= \frac{2\Theta^2 - 6\Theta + 4}{\Theta} + \frac{4\Theta^2 - 11\Theta + 3}{4\Theta - 4}\varepsilon' - \left(\Theta - \frac{7}{4}\right)\varepsilon' \\ &\overset{\varepsilon' < \frac{12\Theta^3 - 40\Theta^2 + 60\Theta - 32}{4\Theta^2 - 11\Theta + 3}}{<} \frac{3\Theta^2 - 5\Theta + 4}{\Theta} - \left(\Theta - \frac{7}{4}\right)\varepsilon' \\ &< v_3. \end{aligned} \tag{5.32}$$

Because of the inequalities (5.31) and (5.32) SMARTSTART starts the schedule S_3 directly after finishing schedule S_2 at time

$$t_3 = \max\{v_3, w_3\} \overset{(5.31),(5.32)}{=} v_3 = \max\left\{\frac{5\Theta - 3}{2}, \frac{4\Theta^2 - 9\Theta + 6}{\Theta}\right\} - \left(\Theta - \frac{7}{4}\right)\varepsilon'.$$

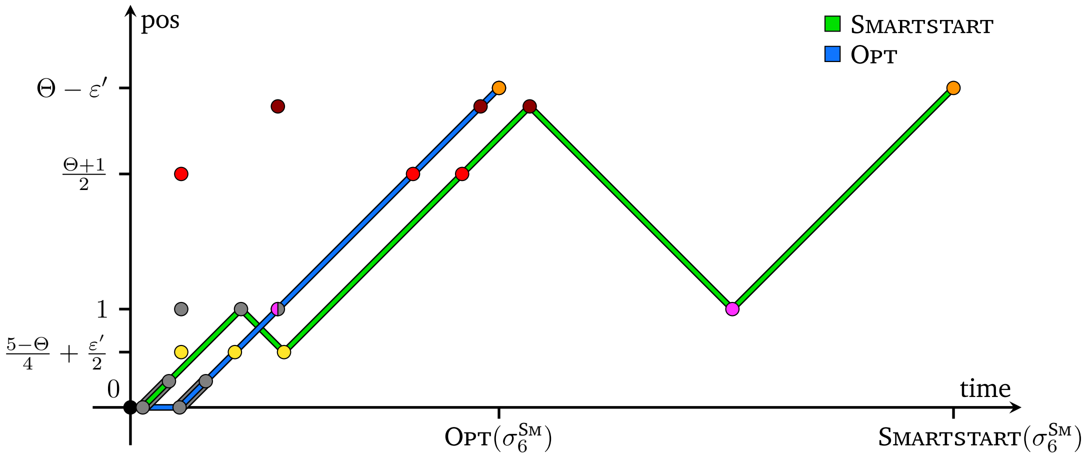

Figure 5.14: SmartStart's and Opt's walk serving σ_6^{SM} with $\varepsilon' = 0.5$ and $\Theta = 3.75$. Request $s_1^{(1)}$ is red ●, $s_1^{(2)}$ is yellow ●, $s_2^{(1)}$ is violet ●, $s_2^{(2)}$ is brown ● and s_3 is orange ●. The requests of Lemma 5.6 are gray ●.

To sum it up, we have

$$\begin{aligned}\textsc{SmartStart}(\sigma_6^{\mathrm{SM}}) &= t_3 + L(t_3, p_3, (s_3)) \\ &= \max\left\{\frac{7\Theta-5}{2}, \frac{6\Theta^2 - 15\Theta + 10}{\Theta}\right\} - \left(\Theta - \frac{11}{4}\right)\varepsilon'. \qquad \square\end{aligned}$$

Equipped with the Lemmas 5.36 and 5.37, we can compute lower bounds for the competitive ratio of SmartStart for online TSP for $\Theta \in (\frac{7}{2}, \infty)$. We start with the subinterval $(\frac{7}{2}, 4]$.

Lemma 5.38. *Let $\frac{7}{2} < \Theta \leq 4$. For every sufficiently small $\varepsilon > 0$, we have*

$$\frac{\textsc{SmartStart}(\sigma_6^{\mathrm{SM}})}{\textsc{Opt}(\sigma_6^{\mathrm{SM}})} = \frac{7\Theta - 5}{2\Theta} - \varepsilon =: f_{6.1}^{\mathrm{low}}(\Theta) - \varepsilon.$$

In particular, we have

$$\frac{\textsc{SmartStart}(\sigma_6^{\mathrm{SM}})}{\textsc{Opt}(\sigma_6^{\mathrm{SM}})} > \rho_{\mathrm{SM}}^{\mathrm{T},\mathbb{R}} \approx 2.7604$$

for $\Theta \in (\frac{7}{2}, 4]$.

Proof. Let $\varepsilon > 0$ with $\varepsilon < \min\{\frac{4\Theta-11}{4\Theta}, \frac{1}{40}\}$ and $\varepsilon' = \frac{4\Theta}{4\Theta-11}\varepsilon$. By Lemma 5.37, we have

$$\begin{aligned}\textsc{SmartStart}(\sigma_6^{\textsc{Sm}}) &\overset{\text{Lem 5.37}}{=} \max\left\{\frac{7\Theta-5}{2}, \frac{6\Theta^2-15\Theta+10}{\Theta}\right\} - \left(\Theta - \frac{11}{4}\right)\varepsilon' \\ &\overset{\Theta\leq 4}{=} \frac{7\Theta-5}{2} - \left(\Theta - \frac{11}{4}\right)\varepsilon'.\end{aligned}$$

Since we have $\varepsilon' < 1$, Lemma 5.36 implies

$$\textsc{Opt}(\sigma_6^{\textsc{Sm}}) = \max\left\{\Theta, \frac{2\Theta^2-5\Theta+4}{\Theta}\right\} \overset{\Theta\leq 4}{=} \Theta.$$

Since we have $\varepsilon' = \frac{4\Theta}{4\Theta-11}\varepsilon$, we obtain

$$\frac{\textsc{SmartStart}(\sigma_6^{\textsc{Sm}})}{\textsc{Opt}(\sigma_6^{\textsc{Sm}})} = \frac{7\Theta-5}{2\Theta} - \frac{4\Theta-11}{4\Theta}\varepsilon' = \frac{7\Theta-5}{2\Theta} - \varepsilon = f_{6.1}^{\text{low}}(\Theta) - \varepsilon,$$

as claimed. The function $f_{6.1}^{\text{low}}$ is strictly monotonically increasing on $[\frac{7}{2}, 4]$. Therefore, we have

$$\frac{\textsc{SmartStart}(\sigma_6^{\textsc{Sm}})}{\textsc{Opt}(\sigma_6^{\textsc{Sm}})} - \varepsilon > f_{6.1}^{\text{low}}\left(\frac{7}{2}\right) - \varepsilon > 2.7857 - \varepsilon > \rho_{\textsc{Sm}}^{\text{T},\mathbb{R}} \approx 2.7604$$

for all $\Theta \in (\frac{7}{2}, 4]$ and $\varepsilon < \frac{1}{40}$. ☐

Finally, we examine the subinterval $(4, \infty)$.

Lemma 5.39. *Let $\Theta > 4$. For every sufficiently small $\varepsilon > 0$, we have*

$$\frac{\textsc{SmartStart}(\sigma_6^{\textsc{Sm}})}{\textsc{Opt}(\sigma_6^{\textsc{Sm}})} = \frac{6\Theta^2-15\Theta+10}{2\Theta^2-5\Theta+4} - \varepsilon =: f_{6.2}^{\text{low}}(\Theta) - \varepsilon.$$

In particular, we have

$$\frac{\textsc{SmartStart}(\sigma_6^{\textsc{Sm}})}{\textsc{Opt}(\sigma_6^{\textsc{Sm}})} > \rho_{\textsc{Sm}}^{\text{T},\mathbb{R}} \approx 2.76037$$

for $\Theta \in (4, \infty)$.

Proof. Let $\varepsilon > 0$ with $\varepsilon < \min\{\frac{4\Theta^2-11\Theta}{8\Theta^2-20\Theta+16}, \frac{1}{20}\}$ and $\varepsilon' = \frac{8\Theta^2-20\Theta+16}{4\Theta^2-11\Theta}\varepsilon$. By Lemma 5.37, we have

$$\textsc{SmartStart}(\sigma_6^{\textsc{Sm}}) \overset{\text{Lem 5.37}}{=} \max\left\{\frac{7\Theta-5}{2}, \frac{6\Theta^2-15\Theta+10}{\Theta}\right\} - \left(\Theta - \frac{11}{4}\right)\varepsilon'$$

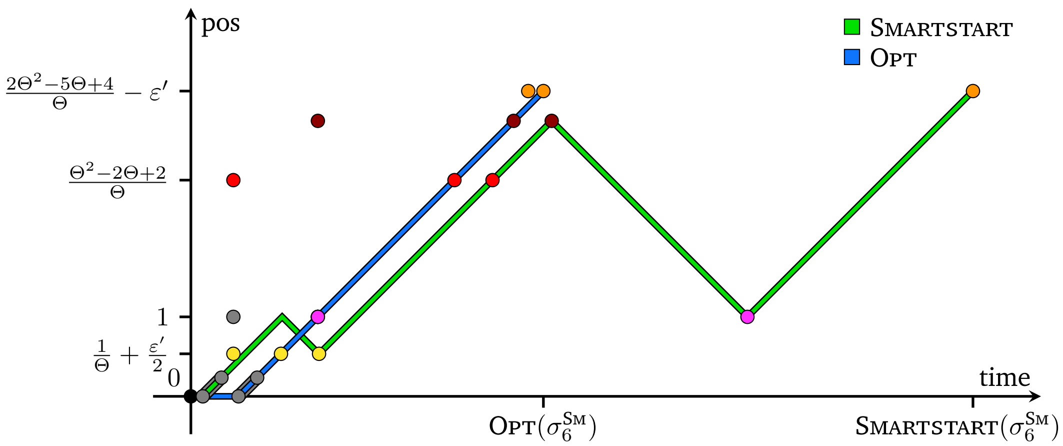

Figure 5.15: SmartStart's and Opt's walk serving σ_6^{SM} with $\varepsilon' = 0.5$ and $\Theta = 4.25$. Request $s_1^{(1)}$ is red ●, $s_1^{(2)}$ is yellow ●, $s_2^{(1)}$ is violet ●, $s_2^{(2)}$ is brown ● and s_3 is orange ●. The requests of Lemma 5.6 are gray ●.

$$\overset{\Theta \geq 4}{\geq} \frac{6\Theta^2 - 15\Theta + 10}{\Theta} - \left(\Theta - \frac{11}{4}\right)\varepsilon'.$$

Since we have $\varepsilon' < 1$, Lemma 5.36 implies

$$\textsc{Opt}(\sigma_6^{\mathrm{SM}}) = \max\left\{\Theta, \frac{2\Theta^2 - 5\Theta + 4}{\Theta}\right\} \overset{\Theta \geq 4}{=} \frac{2\Theta^2 - 5\Theta + 4}{\Theta}.$$

Since we have $\varepsilon' = \frac{8\Theta^2 - 20\Theta + 16}{4\Theta^2 - 11\Theta}\varepsilon$, we obtain

$$\begin{aligned}
\frac{\textsc{SmartStart}(\sigma_6^{\mathrm{SM}})}{\textsc{Opt}(\sigma_6^{\mathrm{SM}})} &\geq \frac{6\Theta^2 - 15\Theta + 10}{2\Theta^2 - 5\Theta + 4} - \frac{4\Theta^2 - 11\Theta}{8\Theta^2 - 20\Theta + 16}\varepsilon' \\
&= \frac{6\Theta^2 - 15\Theta + 10}{2\Theta^2 - 5\Theta + 4} - \varepsilon \\
&= f_{6.2}^{\mathrm{low}}(\Theta) - \varepsilon,
\end{aligned}$$

as claimed. The function $f_{6.2}^{\mathrm{low}}$ is strictly monotonically increasing on $[4, \infty)$. Therefore, we have

$$\frac{\textsc{SmartStart}(\sigma_6^{\mathrm{SM}})}{\textsc{Opt}(\sigma_6^{\mathrm{SM}})} - \varepsilon > f_{6.2}^{\mathrm{low}}(4) - \varepsilon = \frac{23}{8} - \varepsilon > \rho_{\mathrm{SM}}^{\mathrm{T},\mathbb{R}} \approx 2.7604$$

for all $\Theta \in (4, \infty)$ and $\varepsilon < \frac{1}{20}$. □

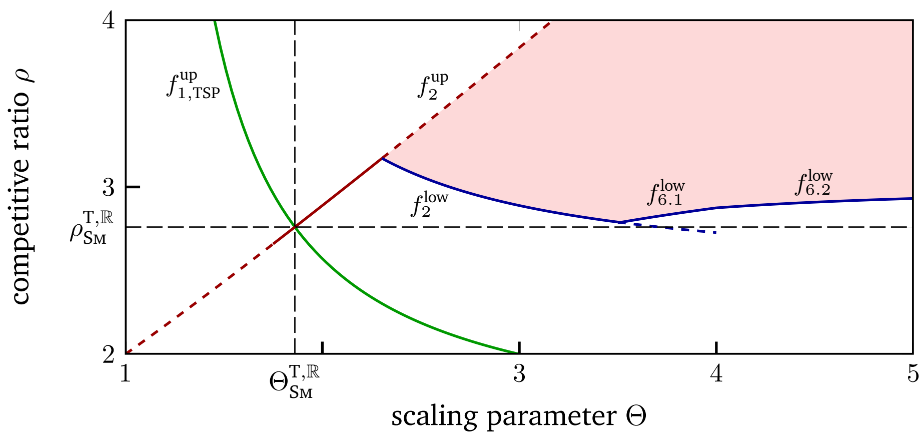

Figure 5.16: Functions $f_{1,\text{TSP}}^{\text{up}}$ (green) / f_2^{up} (red): upper bounds for competitive ratio for postponing / non-postponing case, drawn solid if tight. Functions: f_2^{low}, $f_{6.1}^{\text{low}}$, $f_{6.2}^{\text{low}}$ (blue): lower bound for competitive ratio. Green / red area: possible values for the competitive ratio, bounded by $f_{1,\text{TSP}}^{\text{up}}$ / f_2^{up} and f_2^{low}, $f_{6.1}^{\text{low}}$, $f_{6.2}^{\text{low}}$.

We combine all lower bounds constructed in this section into one general lower bound. See Figure 5.16 for an illustration of all upper and lower bounds for the competitive ratio of online TSP on the line.

Theorem 5.40. *Let* $F_{\text{TSP}} : \mathbb{R}_{>1} \to \mathbb{R}_{>1}$ *be a function with*

$$F_{\text{TSP}}(\Theta) := \begin{cases} f_{1,\text{TSP}}^{\text{up}}(\Theta), & \textit{for } \Theta \in (1, \Theta_{\text{SM}}^{\text{T},\mathbb{R}}], \\ f_2^{\text{up}}(\Theta), & \textit{for } \Theta \in (\Theta_{\text{SM}}^{\text{T},\mathbb{R}}, \frac{1}{2}(1+\sqrt{13})], \\ f_2^{\text{low}}(\Theta), & \textit{for } \Theta \in (\frac{1}{2}(1+\sqrt{13}), \frac{7}{2}], \\ f_{6.1}^{\text{low}}(\Theta), & \textit{for } \Theta \in (\frac{7}{2}, 4], \\ f_{6.2}^{\text{low}}(\Theta), & \textit{for } \Theta \in (4, \infty). \end{cases}$$

Then F_{TSP} *is general lower bound for the competitive ratio of* SMARTSTART *for online* TSP *on the line. The unique minimum of* F_{TSP} *lies in* $\Theta = \Theta_{\text{SM}}^{\text{T},\mathbb{R}} \approx 1.8607$ *and yields a lower bound of*

$$F_{\text{TSP}}(\Theta_{\text{SM}}^{\text{T},\mathbb{R}}) = \rho_{\text{SM}}^{\text{T},\mathbb{R}} \approx 2.7604.$$

Proof. We have shown in Proposition 5.34 that $f_{1,\text{TSP}}^{\text{up}}(\Theta)$ with $\Theta \in (1, \Theta_{\text{SM}}^{\text{T},\mathbb{R}}]$ is a lower bound for the competitive ratio of SMARTSTART for online TSP and in Proposition 5.15 that $f_2^{\text{up}}(\Theta)$ with $\Theta \in (\Theta_{\text{SM}}^{\text{T},\mathbb{R}}, \frac{1}{2}(1+\sqrt{13})]$ is a lower bound. Theorem 5.29 implies that

F_{TSP} has unique minimum in the interval $(1, \frac{1}{2}(1+\sqrt{13})]$ at $\Theta = \Theta_{\text{SM}}^{\text{T},\mathbb{R}}$. It remains to show that $F_{\text{TSP}}(\Theta) > F_{\text{TSP}}(\Theta_{\text{SM}}^{\text{T},\mathbb{R}})$ for all $\Theta \in (\frac{1}{2}(1+\sqrt{13}), \infty)$. This immediately follows from Lemmas 5.30, 5.38 and 5.39. □

The main theorem of this section follows by combining Theorem 5.29 and Theorem 5.40.

Theorem 5.41. *The competitive ratio of* Smartstart *for open online* TSP *on the line with scaling parameter* $\Theta_{\text{SM}}^{\text{T},\mathbb{R}} \approx 1.8607$ *is exactly*

$$\rho_{\text{SM}}^{\text{T},\mathbb{R}} = f_{1,\text{TSP}}^{\text{up}}(\Theta_{\text{SM}}^{\text{T},\mathbb{R}}) = f_2^{\text{up}}(\Theta_{\text{SM}}^{\text{T},\mathbb{R}}) \approx 2.7604.$$

For every other $\Theta > 1$ *with* $\Theta \neq \Theta_{\text{SM}}^{\text{T},\mathbb{R}}$ *the competitive ratio of* Smartstart *is strictly larger than* $\rho_{\text{SM}}^{\text{T},\mathbb{R}}$.

5.4 Upper Bounds for the General Setting

After having examined Smartstart on the real line thoroughly, we now focus on the general setting, i.e., on arbitrary continuous metric spaces. Since the real line $\mathbb{R}$ is a specific continuous metric space, every lower bound established in the previous sections also holds in the general setting. We also note that the upper bounds in the postponing case, i.e., the bound $f_1^{\text{up}}(\Theta)$ from Proposition 5.3 for open online Dial-a-Ride and the bound $f_{1,\text{TSP}}^{\text{up}}(\Theta)$ from Proposition 5.28 for online TSP, also hold in the general setting. The upper bound $f_2^{\text{up}}(\Theta)$ from Proposition 5.4 for the non-postponing case however uses Proposition 4.10, which relies on line-specific features. Therefore, $f_2^{\text{up}}(\Theta)$ is not a valid upper bound in the general setting. We compute a new bound.

Proposition 5.42. *If* Smartstart *for online* Dial-a-Ride *does not postpone the final schedule* S_N*, we have*

$$\frac{\textsc{Smartstart}(\sigma)}{\textsc{Opt}(\sigma)} \leq \Theta + 1 =: f_{2,X}^{\text{up}}(\Theta).$$

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

Let the latter be the case, then the final schedule is started at the release time r_n of the last request. Since Opt also has to serve the last request, we have $\textsc{Opt}(\sigma) \geq r_n$ and since the execution of the final schedule is not postponed, we have $r_n > \frac{1}{\Theta-1} L(t_N, p_N, \sigma_N)$, i.e.,

$$L(t_N, p_N, \sigma_N) < (\Theta - 1)\textsc{Opt}(\sigma). \tag{5.33}$$

In total we have

$$\begin{aligned}\textsc{Smartstart}(\sigma) &\overset{(4.1)}{=} t_N + L(t_N, p_N, \sigma_N)\\ &\overset{(5.4)}{=} r_n + L(t_N, p_N, \sigma_N)\\ &\overset{(5.33)}{<} \Theta\textsc{Opt}(\sigma)\\ &< (\Theta+1)\textsc{Opt}(\sigma).\end{aligned}$$

Now let the final schedule be started immediately after the second to final schedule. Let $s_N^{\textsc{Opt}}$ be the first request of σ_N that is served by Opt and let $a_N^{\textsc{Opt}}$ be its starting position and $r_N^{\textsc{Opt}}$ be its release time. We have

$$\begin{aligned}\textsc{Smartstart}(\sigma) &\overset{(4.1)}{=} t_N + L(t_N, p_N, \sigma_N)\\ &\overset{(5.3)}{=} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + L(t_N, p_N, \sigma_N)\\ &\overset{(4.2)}{\leq} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + L(r_N^{\textsc{Opt}}, p_N, \sigma_N)\\ &\overset{(5.1)}{\leq} \Theta t_{N-1} + L(r_N^{\textsc{Opt}}, p_N, \sigma_N).\end{aligned} \tag{5.34}$$

Since Opt serves all requests of σ_N after time $r_N^{\textsc{Opt}}$, starting with a request with starting position $a_N^{\textsc{Opt}}$, we have

$$\textsc{Opt}(\sigma) \geq r_N^{\textsc{Opt}} + L(r_N^{\textsc{Opt}}, a_N^{\textsc{Opt}}, \sigma_N). \tag{5.35}$$

Furthermore, we have

$$r_N^{\textsc{Opt}} > t_{N-1} \tag{5.36}$$

since otherwise $s_N^{\textsc{Opt}} \in \sigma_{N-1}$ would hold. We have $t_{N-1} < \textsc{Opt}(\sigma)$ since at least one request needs to be released after time t_{N-1}. This gives us

$$\begin{aligned}\textsc{Smartstart}(\sigma) &\overset{(5.34)}{\leq} \Theta t_{N-1} + L(r_N^{\textsc{Opt}}, p_N, \sigma_N)\\ &\overset{(4.3)}{\leq} \Theta t_{N-1} + d(a_N^{\textsc{Opt}}, p_N) + L(r_N^{\textsc{Opt}}, a_N^{\textsc{Opt}}, \sigma_N)\\ &\overset{(5.35)}{\leq} \Theta t_{N-1} + d(a_N^{\textsc{Opt}}, p_N) + \textsc{Opt}(\sigma) - r_N^{\textsc{Opt}}\\ &\overset{(5.36)}{<} (\Theta-1) t_{N-1} + d(a_N^{\textsc{Opt}}, p_N) + \textsc{Opt}(\sigma)\\ &\overset{t_{N-1} < \textsc{Opt}(\sigma)}{<} \Theta\textsc{Opt}(\sigma) + d(a_N^{\textsc{Opt}}, p_N).\end{aligned} \tag{5.37}$$

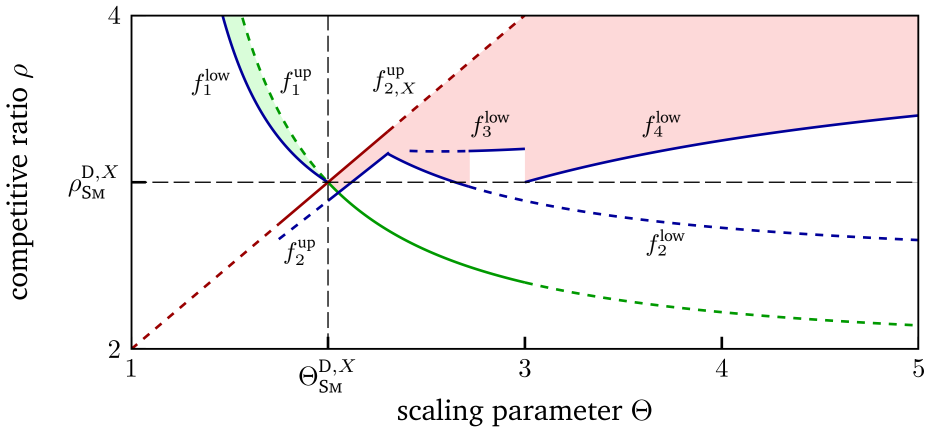

Figure 5.17: Functions f_1^{up} (green) / $f_{2,X}^{\mathrm{up}}$ (red): upper bounds for competitive ratio for postponing / non-postponing case, drawn solid if tight. Functions: f_1^{low} to f_4^{low} and f_2^{up} (blue): lower bounds for competitive ratio. Green / red area: possible values for the competitive ratio, bounded by f_1^{up} / f_2^{up} and f_1^{low} to f_4^{low} as well as f_2^{up}.

Since Opt has to visit both $a_N^{\textsc{Opt}}$ and p_N, we have $d(a_N^{\textsc{Opt}}, p_N) \leq \textsc{Opt}(\sigma)$, i.e.,

$$\textsc{SmartStart}(\sigma) \overset{(5.37)}{<} \Theta\textsc{Opt}(\sigma) + d(a_N^{\textsc{Opt}}, p_N) \leq (\Theta + 1)\textsc{Opt}(\sigma). \qquad \square$$

The upper bound $f_{2,X}^{\mathrm{up}}(\Theta)$ is slightly weaker than the upper bound $f_2^{\mathrm{up}}(\Theta)$. We use Proposition 5.42 to compute a new general upper bound for open online Dial-a-Ride.

Theorem 5.43. *The function* $\max\{f_1^{\mathrm{up}}, f_{2,X}^{\mathrm{up}}\}$ *gives an upper bound for the competitive ratio of* SmartStart *for open online* Dial-a-Ride *in the general setting for all* $\Theta > 1$. *Let* $\Theta_{\mathrm{SM}}^{\mathrm{D},X} = 2$ *be the unique solution of the equation* $f_1^{\mathrm{up}}(\Theta) = f_{2,X}^{\mathrm{up}}(\Theta)$, *i.e., of*

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

in the interval $(1, \infty)$. *Then,* $\Theta_{\mathrm{SM}}^{\mathrm{D},X}$ *is the unique minimum of the function* $\max\{f_1^{\mathrm{up}}, f_{2,X}^{\mathrm{up}}\}$ *and* SmartStart *with scaling parameter* $\Theta_{\mathrm{SM}}^{\mathrm{D},X}$ *is* $\rho_{\mathrm{SM}}^{\mathrm{D},X}$*-competitive with*

$$\rho_{\mathrm{SM}}^{\mathrm{D},X} = f_1^{\mathrm{up}}(\Theta_{\mathrm{SM}}^{\mathrm{D},X}) = f_{2,X}^{\mathrm{up}}(\Theta_{\mathrm{SM}}^{\mathrm{D},X}) = 3.$$

Proof. For the case where SmartStart postpones the final schedule, we have established the upper bound

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

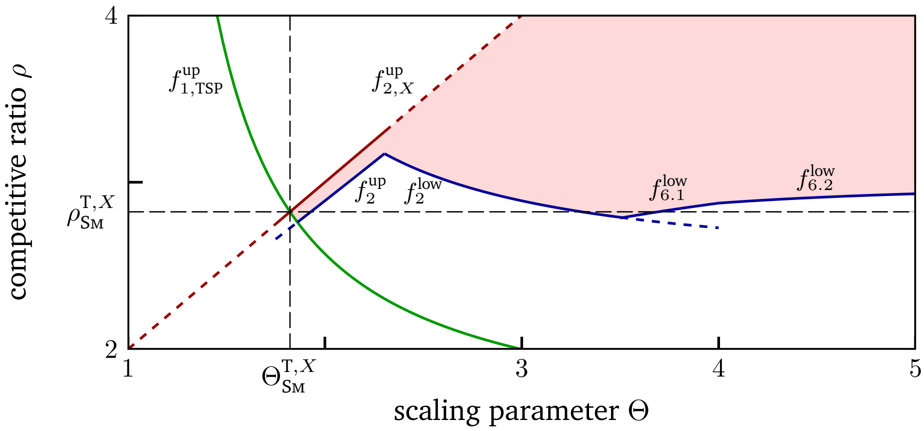

Figure 5.18: Functions $f_{1,\text{TSP}}^{\text{up}}$ (green) / $f_{2,X}^{\text{up}}$ (red): upper bounds for competitive ratio for postponing / non-postponing case, drawn solid if tight. Functions: f_2^{up}, f_2^{low}, $f_{6.1}^{\text{low}}$, $f_{6.2}^{\text{low}}$ (blue): lower bound for competitive ratio. Green / red area: possible values for the competitive ratio, bounded by $f_{1,\text{TSP}}^{\text{up}}$ / $f_{2,X}^{\text{up}}$ and f_2^{up}, f_2^{low}, $f_{6.1}^{\text{low}}$, $f_{6.2}^{\text{low}}$.

in Proposition 5.3, and for the case where SmartStart does not postpone final schedule we have established the upper bound

$$\frac{\textsc{SmartStart}(\sigma)}{\textsc{Opt}(\sigma)} \leq \Theta + 1 = f_{2,X}^{\text{up}}(\Theta)$$

in Proposition 5.42. Thus, the maximum of both bounds is a general upper bound for the competitive ratio of SmartStart for open online Dial-a-Ride in the general setting that is independent of SmartStart's behavior before the final schedule.

Function f_1^{up} is strictly decreasing for $\Theta > 1$ and function $f_{2,X}^{\text{up}}$ is strictly increasing for $\Theta > 1$. Therefore, the minimum of $\max\{f_1^{\text{up}}, f_{2,X}^{\text{up}}\}$ in the interval $(1, \infty)$ lies in the intersection point of f_1^{up} and $f_{2,X}^{\text{up}}$, i.e., in $\Theta_{\text{SM}}^{\text{D},X} = 2$. The resulting upper bound for the competitive ratio is

$$\rho_{\text{SM}}^{\text{D},X} = f_1^{\text{up}}(\Theta_{\text{SM}}^{\text{D},X}) = f_{2,X}^{\text{up}}(\Theta_{\text{SM}}^{\text{D},X}) = 3. \qquad \square$$

See Figure 5.17 for a visualization of the upper bound for the competitive ratio of online Dial-a-Ride on the line presented in Theorem 5.43. Finally, we use Proposition 5.42 to compute a new general upper bound for open online TSP.

Theorem 5.44. *The function* $\max\{f_{1,\text{TSP}}^{\text{up}}, f_{2,X}^{\text{up}}\}$ *gives an upper bound for the competitive ratio of* SmartStart *for open online* TSP *in the general setting for all* $\Theta > 1$. *Let* $\Theta_{\text{SM}}^{\text{T},X} \approx$

1.8229 *be the unique solution of the equation* $f^{\text{up}}_{1,\text{TSP}}(\Theta) = f^{\text{up}}_{2,X}(\Theta)$*, i.e., of*

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

in the interval $(1, \infty)$*. Then,* $\Theta^{\text{T},X}_{\text{SM}}$ *is the unique minimum of the function* $\max\{f^{\text{up}}_{1,\text{TSP}}, f^{\text{up}}_{2,X}\}$ *and* SMARTSTART *with scaling parameter* $\Theta^{\text{T},X}_{\text{SM}}$ *is* $\rho^{\text{T},X}_{\text{SM}}$*-competitive with*

$$\rho^{\text{T},X}_{\text{SM}} = f^{\text{up}}_{1,\text{TSP}}(\Theta^{\text{T},X}_{\text{SM}}) = f^{\text{up}}_{2,X}(\Theta^{\text{T},X}_{\text{SM}}) \approx 2.8229.$$

Proof. For the case where SMARTSTART postpones the final schedule we have established the upper bound

$$\frac{\text{SMARTSTART}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{3\Theta^2 + 3\Theta}{2\Theta^2 + \Theta - 3} = f^{\text{up}}_{1,\text{TSP}}(\Theta)$$

in Proposition 5.28, and for the case where SMARTSTART does not postpone final schedule we have established the upper bound

$$\frac{\text{SMARTSTART}(\sigma)}{\text{OPT}(\sigma)} \leq \Theta + 1 = f^{\text{up}}_{2,X}(\Theta)$$

in Proposition 5.42. Thus, the maximum of both bounds is a general upper bound for the competitive ratio of SMARTSTART for open online TSP in the general setting that is independent of SMARTSTART's behavior before the final schedule.

Function $f^{\text{up}}_{1,\text{TSP}}$ is strictly decreasing for $\Theta > 1$ and function $f^{\text{up}}_{2,X}$ is strictly increasing for $\Theta > 1$. Therefore, the minimum of $\max\{f^{\text{up}}_{1,\text{TSP}}, f^{\text{up}}_{2,X}\}$ in the interval $(1, \infty)$ lies in the intersection point of $f^{\text{up}}_{1,\text{TSP}}$ and $f^{\text{up}}_{2,X}$, i.e., in $\Theta^{\text{T},X}_{\text{SM}} \approx 1.8229$. The resulting upper bound for the competitive ratio is

$$\rho^{\text{T},X}_{\text{SM}} = f^{\text{up}}_{1,\text{TSP}}(\Theta^{\text{T},X}_{\text{SM}}) = f^{\text{up}}_{2,X}(\Theta^{\text{T},X}_{\text{SM}}) \approx 2.8229. \qquad \square$$

See Figure 5.18 for a visualization of the upper bound for the competitive ratio of online DIAL-A-RIDE on the line presented in Theorem 5.44.

Conclusion and Outlook

We provided a conclusive analysis for SMARTSTART in this chapter. We computed tight bounds for the competitive ratio for open online DIAL-A-RIDE and open online TSP on the line and introduced new upper bounds for open online DIAL-A-RIDE and open online TSP in general continuous metric spaces. For the open version of online DIAL-A-RIDE on the line we have shown a tight competitive ratio of 2.9377: The upper bound was proven in Theorem 5.5 and the lower bound was proven in Theorem 5.25. For the open version of online TSP on the line we have shown a tight competitive ratio of 2.7604: The upper bound was proven in Theorem 5.29 and the lower bound was proven in Theorem 5.40. While we have tight results on the line for open online DIAL-A-RIDE and open online TSP, it remains unclear if SMARTSTART performs worse in the general setting: We provided an upper bound of 3 for the competitive ratio of SMARTSTART for open online DIAL-A-RIDE and an upper bound of 2.8229 for the competitive ratio of SMARTSTART for open online TSP. The lower bounds obtained on the real line remain in the general setting. For the closed version, we provided a matching lower bound of 2 in Theroem 4.3 for the upper bound provided by Ascheuer et al. in [5, Thm 6]. Consequently, SMARTSTART is a best possible schedule-based online algorithm for closed online DIAL-A-RIDE and closed online TSP on both, the real line and the general setting. See Table 2.7 for a summary of the results.

If we compare SMARTSTART's competitiveness with the general bounds for online DIAL-A-RIDE and online TSP from Table 2.1, we see that our analysis of SMARTSTART provides an improved upper bound for the competitive ratio of open online DIAL-A-RIDE on the real line as well as on general continuous metric spaces, improving the best known bound from 3.4142 to 2.9377 on the real line and to 3 on general continuous metric spaces. While this is a significant improvement, there is still a large gap between SMARTSTART's competitive ratios and the best known general lower bounds for open online DIAL-A-RIDE and open online TSP. Moreover, there is also a significant gap between SMARTSTART's competitive ratio and the best known lower bounds for open schedule-based algorithms. This indicates that the open version of SMARTSTART is a rather weak schedule-based algorithm. Indeed, SMARTSTART contains a major flaw: the luring mechanic introduced in Lemma 5.6. In the next chapter we will introduce an improved schedule-based algorithm that avoids luring by using the complete known request sequence instead of just the unserved requests as a basis for computing its waiting time. Fittingly, we call this algorithm SMARTERSTART.

6 Algorithm SMARTERSTART

Recall that SMARTSTART uses the length of the next schedule as basis for its waiting function. For the open version of online DIAL-A-RIDE and online TSP, this calculation is flawed in two ways: On one hand, SMARTSTART does not use all available information, i.e., it uses just the unserved requests and not all already released requests to calculate its waiting time. This leads to ignoring some requests when calculating the waiting time as shown in Lemma 5.6, where it forces SMARTSTART to walk to any position with essentially no waiting time. On the other hand, SMARTSTART's waiting routine is designed to keep SMARTSTART competitive in comparison to an optimum offline solution starting from its current postition, while OPT always starts from the origin. Therefore, dependent on SMARTSTART's distance to the origin, its waiting time can be a lot longer than necessary.

Algorithm 4 SMARTERSTART

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

We fix these two issues by slightly adjusting the waiting routine. Essentially, at time t, SMARTERSTART (see Algorithm 4) waits before starting an optimum schedule to serve all currently unserved requests R_t at time

$$\min\left\{t' \in \mathbb{R}_{\geq 0} : t' \geq t \wedge t' \geq \frac{L(t', 0, \sigma_{\leq t'})}{\Theta - 1}\right\}, \tag{6.1}$$

where $\Theta > 1$ is again a scaling parameter. Formally SMARTERSTART is a schedule-based algorithm with waiting function

$$\text{wait}_{\mathrm{S+}}(t) := \begin{cases} \text{false}, & \text{if } R_t \neq \emptyset \text{ and } t \geq \frac{L(t,0,\sigma_{\leq t})}{\Theta-1}, \\ \text{true}, & \text{otherwise.} \end{cases}$$

In difference to SMARTSTART, the algorithm SMARTERSTART bases its waiting routine on the length of the optimum offline schedule serving *all* known requests starting from the origin instead of basing it on the length of the next executed schedule. Recall that the luring weakness of SMARTSTART is triggered by iteratively releasing requests that are close to each other, exploiting that SMARTSTART only takes the distance to the next request as basis for the computation of its waiting time (see Lemma 5.6). To be more precise, SMARTSTART is lured to position q by releasing requests $s_i = (i\varepsilon, i\varepsilon; \frac{\varepsilon}{\Theta-1} + (i-1)\varepsilon)$ for every $i \in \{1, \ldots, \frac{q}{\Theta\varepsilon}\}$, which are all served in seperate schedules. While SMARTSTART only uses the distance to the next request, which is always ε, to compute its waiting time, SMARTERSTART bases its waiting time on the optimum offline schedule serving all known requests from the origin, which increases by ε for every released request. This makes SMARTERSTART not vulnerable to this luring mechanic.

We analyze SMARTERSTART's competitiveness. Regarding open online DIAL-A-RIDE, we show that SMARTERSTART has a competitive ratio of $\rho_{\mathrm{S+}}^{\mathrm{D},\mathbb{R}} \approx 2.6662$ on the line for parameter value $\Theta_{\mathrm{S+}}^{\mathrm{D},\mathbb{R}} \approx 1.7125$ (Thm 6.20) and is $\rho_{\mathrm{S+}}^{\mathrm{D},X}$-competitive with $\rho_{\mathrm{S+}}^{\mathrm{D},X} = 2.6956$ in the general setting for parameter value $\Theta_{\mathrm{S+}}^{\mathrm{D},X} \approx 1.6956$ (Thm 6.36). For open online TSP, we show that SMARTERSTART achieves a competitive ratio of $\rho_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}} \approx 2.6288$ on the line for parameter value $\Theta_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}} \approx 1.6789$ (Thm 6.34) and is $\rho_{\mathrm{S+}}^{\mathrm{T},X}$-competitive with $\rho_{\mathrm{S+}}^{\mathrm{T},X} \approx 2.6625$ in the general setting for parameter value $\Theta_{\mathrm{S+}}^{\mathrm{T},X} \approx 1.6625$ (Thm 6.37).

We published the results of the first two sections also in [12]. Similar to SMARTSTART we show an upper bounds for the competitive ratio of SMARTERSTART for open online DIAL-A-RIDE on the line by deriving two separate upper bounds depending on Θ: an upper bound for the case that SMARTERSTART postpones starting its final schedule and an upper bound for the case that SMARTERSTART does not postpone its final schedule (see Section 6.1). We complement the upper bounds with matching lower bounds in Section 6.2. For online TSP on the line we show a slightly stronger upper bound for the competitive ratio for the case that the final schedule is postponed. This improves the general upper bound in comparison to the DIAL-A-RIDE version (see Section 6.3). In the same section, we complement this slightly stronger upper bound with a matching lower bound. For the general setting, we show slightly weaker upper bounds for the competitive ratio of SMARTERSTART for open online DIAL-A-RIDE and online TSP in the case that the final schedule is not postponed. This yields slightly weaker general upper bounds than on the real line (see Section 6.4). For the closed version of SMARTERSTART, we provide an upper bound for the competitive ratio of 2, which is complemented by the general lower bound for schedule-based algorithms provided by Theorem 4.3 (see Section 6.5).

We start with the computation of an upper bound for the competitive ratio of SMARTERSTART. Similar to SMARTSTART and independent of which version, open or closed, we examine, we distinguish between three different cases concerning the starting time of the final schedule. If SMARTERSTART postpones the execution of the final schedule S_N (i.e., it waits even though there are unserved requests), the starting time of schedule S_N is given by

$$t_N = \frac{1}{\Theta - 1} L(t_N, 0, \sigma_{\leq t_N}). \tag{6.2}$$

If SMARTERSTART does not postpone the final schedule, we have

$$t_N = t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) \tag{6.3}$$

if the final schedule S_N is executed directly after the second to final schedule and

$$t_N = r_n. \tag{6.4}$$

if there are no unserved requests at the point of time the execution of S_{N-1} is finished and the last requests are released at time $r_n > \frac{1}{\Theta-1} L(t_N, 0, \sigma_{\leq t_N})$.

First, we examine the case that the final schedule is postponed. We start by giving a lower bound for the starting time of a schedule.

6.1 Upper Bound for the Open Online DIAL-A-RIDE on the Line

We start by giving a lower bound for the starting time of a schedule. For SMARTSTART, a schedule S_j is never started earlier than time $\frac{1}{\Theta} d(0, p_{j+1})$ (see Lemma 5.1). This changes slightly for SMARTERSTART.

Lemma 6.1. *Algorithm* SMARTERSTART *for open online* DIAL-A-RIDE *does not start schedule S_j earlier than time $\frac{1}{\Theta-1} d(0, p_{j+1})$, i.e., we have $t_j \geq \frac{1}{\Theta-1} d(0, p_{j+1})$.*

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

$$L(t_j, 0, \sigma_{\leq t_j}) \geq d(0, p_{j+1}) \tag{6.5}$$

and therefore

$$t_j \overset{(6.1)}{\geq} \frac{L(t_j, 0, \sigma_{\leq t_j})}{\Theta - 1} \overset{(6.5)}{\geq} \frac{1}{\Theta - 1} d(0, p_{j+1}). \qquad \square$$

Using Lemma 6.1 and Lemma 4.5, we can give an upper bound for the length of SMARTERSTART's schedules, which is an essential ingredient in our upper bounds. A similar bound for SMARTSTART was proven in Lemma 5.2.

Lemma 6.2. *For every schedule S_j of* SMARTERSTART *for open online* DIAL-A-RIDE*, we have*

$$L(t_j, p_j, \sigma_j) \leq \left(1 + \frac{\Theta - 1}{\Theta + 1}\right)\textsc{Opt}(\sigma).$$

Proof. By Lemma 4.5 and Lemma 6.1 we have

$$\begin{aligned} L(t_j, p_j, \sigma_j) &\overset{\text{Lem 4.5}}{\leq} \min\{\textsc{Opt}(\sigma) + d(p_j, 0), 2(\textsc{Opt}(\sigma) - t_{j-1})\} \\ &\overset{\text{Lem 6.1}}{\leq} \min\left\{\textsc{Opt}(\sigma) + d(p_j, 0), 2\left(\textsc{Opt}(\sigma) - \frac{1}{\Theta - 1} d(p_j, 0)\right)\right\} \\ &\leq \left(1 + \frac{\Theta - 1}{\Theta + 1}\right)\textsc{Opt}(\sigma) \end{aligned}$$

since the minimum above is largest if the two terms are equal, which is the case for $d(p_j, 0) = \frac{\Theta-1}{\Theta+1}\textsc{Opt}(\sigma)$. □

The following proposition uses Lemma 6.2 to provide an upper bound for the competitive ratio of SMARTERSTART, in the case that SMARTERSTART does have a waiting period before starting the final schedule.

Proposition 6.3. *In case* SMARTERSTART *for open online* DIAL-A-RIDE *postpones executing S_N, we have*

$$\frac{\textsc{SmarterStart}(\sigma)}{\textsc{Opt}(\sigma)} \leq \frac{2\Theta^2 - \Theta + 1}{\Theta^2 - 1} =: g_1^{\text{up}}(\Theta).$$

Proof. Assume SMARTERSTART postpones the final schedule. Then Lemma 6.2 yields the claimed bound:

$$\begin{aligned} \textsc{SmarterStart}(\sigma) &\overset{(4.1)}{=} t_N + L(t_N, p_N, \sigma_N) \\ &\overset{(6.2)}{=} \frac{1}{\Theta - 1} L(t_N, 0, \sigma_{\leq t_N}) + L(t_N, p_N, \sigma_N) \\ &\overset{(4.4)}{\leq} \frac{1}{\Theta - 1}\textsc{Opt}(\sigma) + L(t_N, p_N, \sigma_N) \\ &\overset{\text{Lem 6.2}}{\leq} \left(\frac{1}{\Theta - 1} + 1 + \frac{\Theta - 1}{\Theta + 1}\right)\textsc{Opt}(\sigma) \\ &= \frac{2\Theta^2 - \Theta + 1}{\Theta^2 - 1}\textsc{Opt}(\sigma). \end{aligned}$$

□

In comparison, the upper bound for the competitive ratio of SMARTSTART, in the case that the final schedule is postponed, is $\frac{2\Theta^2+2\Theta}{\Theta^2+\Theta-2}$ (see Proposition 5.3). Note that SMARTER-START's bound is better than SMARTSTART's bound for $\Theta > 1$.

Using the bound established by Proposition 4.10, we can give an upper bound for the competitive ratio of SMARTERSTART for open online DIAL-A-RIDE on the line if the server is not waiting before starting the final schedule. Note that by using Proposition 4.10, the resulting upper bound is not valid for general continuous metric spaces.

Proposition 6.4. *If* SMARTERSTART *for open online* DIAL-A-RIDE *on the line does not postpone executing S_N, we have*

$$\frac{\textsc{SmarterStart}(\sigma)}{\textsc{Opt}(\sigma)} \leq \frac{3\Theta^2+3}{2\Theta+1} =: g_2^{\text{up}}(\Theta).$$

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

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

$$\textsc{Opt}(\sigma) \geq r_n. \tag{6.6}$$

In total we have

$$\begin{aligned}
\textsc{SmarterStart}(\sigma) &\overset{(4.1)}{=} t_N + L(t_N, p_N, \sigma_N)\\
&\overset{(6.4)}{=} r_n + L(t_N, p_N, \sigma_N)\\
&\overset{(6.6)}{\leq} \textsc{Opt}(\sigma) + L(t_N, p_N, \sigma_N)\\
&\overset{\text{Lem 6.2}}{\leq} \left(2 + \frac{\Theta-1}{\Theta+1}\right)\textsc{Opt}(\sigma)\\
&\overset{\Theta \geq 1}{<} \frac{3\Theta^2+3}{2\Theta+1}\textsc{Opt}(\sigma).
\end{aligned}$$

Now, consider the case that the final schedule is started immediately after the second to final schedule. Let $s_N^{\textsc{Opt}}$ be the first request of σ_N that is served by OPT and let $a_N^{\textsc{Opt}}$ be its starting position and $r_N^{\textsc{Opt}}$ be its release time. We have

$$\begin{aligned}
\textsc{SmarterStart}(\sigma) &\overset{(4.1)}{=} t_N + L(t_N, p_N, \sigma_N)\\
&\overset{(6.3)}{=} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + L(t_N, p_N, \sigma_N)
\end{aligned}$$

$$\overset{t_N \geq r_N^{\text{OPT}}}{\leq} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + L(r_N^{\text{OPT}}, p_N, \sigma_N). \tag{6.7}$$

Since OPT serves all requests of σ_N after time r_N^{OPT}, starting with a request with starting position a_N^{OPT}, we have

$$\text{OPT}(\sigma) \geq r_N^{\text{OPT}} + L(r_N^{\text{OPT}}, a_N^{\text{OPT}}, \sigma_N). \tag{6.8}$$

Furthermore, we have

$$r_N^{\text{OPT}} > t_{N-1} \tag{6.9}$$

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

$$\begin{aligned}
\text{SMARTERSTART}(\sigma) &\overset{(6.7)}{\leq} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + L(r_N^{\text{OPT}}, p_N, \sigma_N) \\
&\overset{(4.3)}{\leq} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + |a_N^{\text{OPT}} - p_N| \\
&\quad + L(r_N^{\text{OPT}}, a_N^{\text{OPT}}, \sigma_N) \\
&\overset{(6.8)}{\leq} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + |a_N^{\text{OPT}} - p_N| + \text{OPT}(\sigma) - r_N^{\text{OPT}} \\
&\overset{(6.9)}{<} L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + |a_N^{\text{OPT}} - p_N| + \text{OPT}(\sigma) \qquad (6.10) \\
&\overset{(4.3)}{\leq} |p_{N-1}| + L(t_{N-1}, 0, \sigma_{N-1}) + |a_N^{\text{OPT}} - p_N| + \text{OPT}(\sigma) \\
&\overset{\text{Lem 6.1}}{\leq} (\Theta - 1)t_{N-2} + L(t_{N-1}, 0, \sigma_{N-1}) + |a_N^{\text{OPT}} - p_N| + \text{OPT}(\sigma). \qquad (6.11)
\end{aligned}$$

We have

$$\text{OPT}(\sigma) \geq t_{N-2} + |a_N^{\text{OPT}} - p_N|, \tag{6.12}$$

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

$$\begin{aligned}
\text{SMARTERSTART}(\sigma) &\overset{(6.11)}{<} (\Theta - 1)t_{N-2} + L(t_{N-1}, 0, \sigma_{N-1}) + |a_N^{\text{OPT}} - p_N| + \text{OPT}(\sigma) \\
&\overset{(6.12)}{\leq} 2\text{OPT}(\sigma) + L(t_{N-1}, 0, \sigma_{N-1}) + (\Theta - 2)t_{N-2}.. \qquad (6.13)
\end{aligned}$$

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

$$\text{SMARTERSTART}(\sigma) \overset{(6.13)}{<} 2\text{OPT}(\sigma) + L(t_{N-1}, 0, \sigma_{N-1}) + (\Theta - 2)t_{N-2}$$

$$\begin{aligned} &\overset{(4.4)}{\leq} (\Theta+1)\text{OPT}(\sigma) \\ &\overset{\Theta \geq 2}{\leq} \frac{3\Theta^2+3}{2\Theta+1}\text{OPT}(\sigma). \end{aligned}$$

Thus, we may assume $\Theta < 2$. Similarly as in inequality (6.13), we get

$$\begin{aligned} \text{SMARTERSTART}(\sigma) &\overset{(6.11)}{<} (\Theta-1)t_{N-2} + L(t_{N-1}, 0, \sigma_{N-1}) + |a_N^{\text{OPT}} - p_N| + \text{OPT}(\sigma) \\ &\overset{(6.12)}{\leq} \Theta\text{OPT}(\sigma) + L(t_{N-1}, 0, \sigma_{N-1}) + (2-\Theta)|a_N^{\text{OPT}} - p_N| \\ &\overset{(6.1)}{\leq} \Theta\text{OPT}(\sigma) + (\Theta-1)t_{N-1} + (2-\Theta)|a_N^{\text{OPT}} - p_N| \\ &\leq (2\Theta-1)\text{OPT}(\sigma) + (2-\Theta)|a_N^{\text{OPT}} - p_N|, \end{aligned} \tag{6.14}$$

where the last inequality follows, because of $\text{OPT}(\sigma) \geq t_{N-1}$. This means the claim is shown if we have

$$|a_N^{\text{OPT}} - p_N| \leq \left(1 - \frac{\Theta-1}{2\Theta+1}\right)\text{OPT}(\sigma) \tag{6.15}$$

since then we have

$$\begin{aligned} \text{SMARTERSTART}(\sigma) &\overset{(6.14)}{<} (2\Theta-1)\text{OPT}(\sigma) + (2-\Theta)|a_N^{\text{OPT}} - p_N| \\ &\overset{(6.15)}{\leq} (2\Theta-1)\text{OPT}(\sigma) + (2-\Theta)\left(1 - \frac{\Theta-1}{2\Theta+1}\right)\text{OPT}(\sigma) \\ &= \frac{3\Theta^2+3}{2\Theta+1}\text{OPT}(\sigma). \end{aligned}$$

Therefore, we may assume in the following that

$$|p_N - a_N^{\text{OPT}}| > \left(1 - \frac{\Theta-1}{2\Theta+1}\right)\text{OPT}(\sigma). \tag{6.16}$$

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

$$|p_N - a_N^{\text{OPT}}| + y \leq \text{OPT}(\sigma). \tag{6.17}$$

In the case that OPT visits position p_N before it collects s_N^{OPT}, we have

$$|a_N^{\text{OPT}} - p_N| + |p_N| \leq \text{OPT}(\sigma). \tag{6.18}$$

Similarly, if OPT collects s_N^{OPT} before it visits position p_N for the first time, we have

$$\text{OPT}(\sigma) \quad \geq \quad r_N^{\text{OPT}} + |a_N^{\text{OPT}} - p_N|$$

$$\begin{aligned}
&\overset{(6.9)}{>} \quad t_{N-1} + |a_N^{\text{OPT}} - p_N| \\
&\overset{\text{Lem 6.1}}{\geq} \quad \frac{|p_N|}{\Theta - 1} + |a_N^{\text{OPT}} - p_N| \\
&\overset{\Theta < 2}{\geq} \quad |p_N| + |a_N^{\text{OPT}} - p_N|.
\end{aligned}$$

Thus, inequality (6.18) holds in general. To sum it up, we may assume that

$$\max\{y, |p_N|, t_{N-2}\} \overset{(6.16),(6.17),(6.18),(6.12)}{<} \frac{\Theta - 1}{2\Theta + 1}\text{OPT}(\sigma) \tag{6.19}$$

holds. By Proposition 4.10 we have

$$\begin{aligned}
\text{SMARTERSTART}(\sigma) &\overset{(6.10)}{<} L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + |p_N - a_N^{\text{OPT}}| + \text{OPT}(\sigma) \\
&\overset{\text{Prop 4.10}}{\leq} 2|p_{N-1}| + |p_{N-1} - p_N| + 2y + |p_N - a_N^{\text{OPT}}| + \text{OPT}(\sigma) \\
&\overset{(6.17)}{\leq} 2|p_{N-1}| + |p_{N-1} - p_N| + y + 2\text{OPT}(\sigma) \\
&\leq 3|p_{N-1}| + |p_N| + y + 2\text{OPT}(\sigma) \\
&\overset{\text{Lem 6.1}}{\leq} (3\Theta - 3)t_{N-2} + |p_N| + y + 2\text{OPT}(\sigma) \\
&\overset{(6.19)}{\geq} \left((3\Theta - 3)\frac{\Theta - 1}{2\Theta + 1} + 2\frac{\Theta - 1}{2\Theta + 1} + 2\right)\text{OPT}(\sigma) \\
&= \frac{3\Theta^2 + 3}{2\Theta + 1}\text{OPT}(\sigma).
\end{aligned}$$

□

In comparison, the upper bound for the competitive ratio of SMARTSTART, in case that the final schedule is not postponed, is $\frac{3\Theta^2+5\Theta+4}{3\Theta+3}$ (see Proposition 5.4). Note that SMARTERSTART's bound is slightly worse than SMARTSTART's bound for $\Theta > 1.47$. However, in combination with the bound of Proposition 6.3, SMARTERSTART has a better worst-case than SMARTSTART. We compute a general upper bound for the competitive ratio of SMARTERSTART for open online DIAL-A-RIDE on the line.

Theorem 6.5. *The function* $\max\{g_1^{\text{up}}, g_2^{\text{up}}\}$ *gives an upper bound for the competitive ratio of* SMARTERSTART *for open online* DIAL-A-RIDE *on the line for all* $\Theta > 1$. *Let* $\Theta_{\text{S+}}^{\text{D},\mathbb{R}} \approx 1.7125$ *be the unique solution of* $g_1^{\text{up}}(\Theta) = g_2^{\text{up}}(\Theta)$, *i.e., of*

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

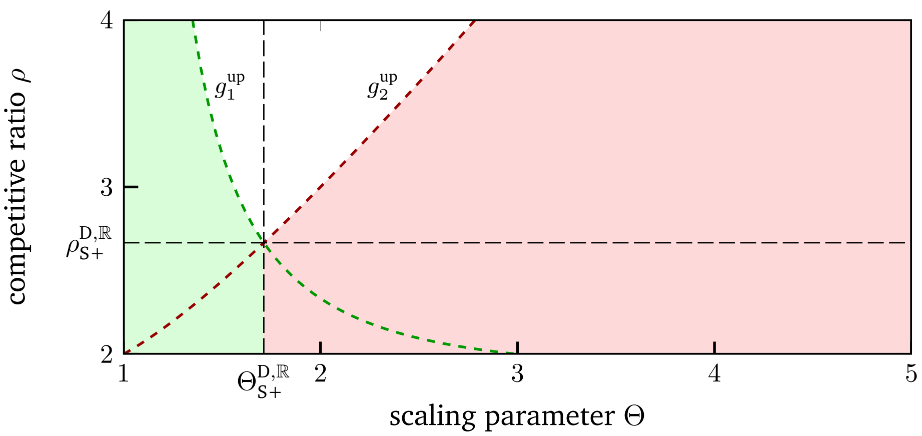

Figure 6.1: Functions g_1^{up} (green) / g_2^{up} (red): upper bounds for competitive ratio for postponing / non-postponing case. Green / red area: possible values for the competitive ratio, bounded by g_1^{up} / g_2^{up}.

in the interval $(1, \infty)$. *Then,* $\Theta_{S+}^{D,\mathbb{R}}$ *is the unique minimum of the function* $\max\{g_1^{up}, g_2^{up}\}$ *and* SMARTERSTART *with scaling parameter* $\Theta_{S+}^{D,\mathbb{R}}$ *is* $\rho_{S+}^{D,\mathbb{R}}$*-competitive with*

$$\rho_{S+}^{D,\mathbb{R}} := g_1^{up}(\Theta_{S+}^{D,\mathbb{R}}) = g_2^{up}(\Theta_{S+}^{D,\mathbb{R}}) \approx 2.6662.$$

Proof. For the case where SMARTERSTART postpones the final schedule we have established the upper bound

$$\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{2\Theta^2 - \Theta + 1}{\Theta^2 - 1} = g_1^{up}(\Theta)$$

in Proposition 6.3, and for the case where SMARTERSTART starts the final schedule immediately after the second to final schedule we have established the upper bound

$$\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{3\Theta^2 + 3}{2\Theta + 1} = g_2^{up}(\Theta)$$

in Proposition 6.4. Thus, the maximum of both bounds is a general upper bound for the competitive ratio of SMARTERSTART for open online DIAL-A-RIDE on the line that is independent of SMARTERSTART's behavior.

The function g_1^{up} is strictly decreasing for $\Theta > 1$ and the function g_2^{up} is strictly increasing for $\Theta > 1$. Therefore, the minimum of $\max\{g_1^{up}, g_2^{up}\}$ in the interval $(1, \infty)$ lies in the intersection point of g_1^{up} and g_2^{up}, i.e., in $\Theta_{S+}^{D,\mathbb{R}} \approx 1.7125$. The resulting upper bound for the competitive ratio is

$$\rho_{S+}^{D,\mathbb{R}} = g_1^{up}(\Theta_{S+}^{D,\mathbb{R}}) = g_2^{up}(\rho_{S+}^{D,\mathbb{R}}) \approx 2.6662. \qquad \square$$

See Figure 6.1 for a visualization of the upper bound for the competitive ratio of SmarterStart for open online Dial-a-Ride on the line presented in Theorem 6.5.

6.2 Lower Bound for the Open Online Dial-a-Ride on the Line

In this section we present matching lower bounds for the upper bounds provided by the Propositions 6.3 and 6.4. In particular, we show that for $\Theta \in (1, 2)$ there are request sequences where SmarterStart postpones its final schedule and has competitive ratio at least $g_1^{\mathrm{up}}(\Theta)$. Similarly, we show that for $\Theta \in [\frac{1}{2}(1+\sqrt{5}), 2]$ there are instances where SmarterStart does not postpone its final schedule and has competitive ratio at least $g_2^{\mathrm{up}}(\Theta)$. Together, this implies that the general upper bound of $\max\{g_1^{\mathrm{up}}, g_2^{\mathrm{up}}\}$ is tight for $\Theta \in (1, 2]$ and thus for $\Theta = \Theta_{\mathrm{S+}}^{\mathrm{D},\mathbb{R}}$ as defined in Theorem 6.5. Combining these lower bounds with additional lower bounds for $\Theta > 2$, we will show that $\Theta_{\mathrm{S+}}^{\mathrm{D},\mathbb{R}} \approx 1.7125$ is the optimum choice of the scaling parameter Θ, i.e., all $\Theta > 1$ with $\Theta \neq \Theta_{\mathrm{S+}}^{\mathrm{D},\mathbb{R}}$ yield competitive ratios larger than $\rho_{\mathrm{S+}}^{\mathrm{D},\mathbb{R}}$.

As before with algorithm SmartStart, we denote by w_j the earliest time, a potential waiting period before schedule S_j is over, i.e., $w_j = \frac{1}{\Theta-1}L(v_j, 0, \sigma_{\leq v_j})$. Consequently, if no new requests appear between the ending time of schedule S_{j-1} and the starting time of schedule S_j (which will be the case for all request sequences constructed in this section), we have $t_j = \max\{v_j, w_j\}$.

In the following, we will analyze three different request sequences $\sigma_1^{\mathrm{S+}}$, $\sigma_2^{\mathrm{S+}}$ and $\sigma_3^{\mathrm{S+}}$. We will see that the ratio of SmarterStart's and Opt's completion time of $\sigma_1^{\mathrm{S+}}$ tightly matches the upper bound of Proposition 6.3 for $\Theta \in (1, 2)$ and that the ratio of SmarterStart's and Opt's completion time of $\sigma_2^{\mathrm{S+}}$ tightly matches the upper bound of Proposition 6.4 for $\Theta \in [1.6180, 2]$. The request sequence $\sigma_3^{\mathrm{S+}}$ will provide an additional lower bound for the competitive ratio of SmarterStart for open online Dial-a-Ride on the line for larger values of Θ. We start with the request sequence $\sigma_1^{\mathrm{S+}}$.

Definition 6.6. Let $\varepsilon' > 0$ with $\varepsilon' < \frac{1}{\Theta-1}$. We define

$$\sigma_1^{\mathrm{S+}} := (s_1, s_2)$$

with

$$\begin{aligned} s_1 &= (1, 1; 0), \\ s_2 &= \left(-\frac{1}{\Theta-1} + \varepsilon', 1; \frac{1}{\Theta-1} + \varepsilon'\right). \end{aligned}$$

Note that $\varepsilon' < \frac{1}{\Theta-1}$ implies $a_2 < 0$, i.e., the request s_2 is on the left side of the origin. We begin our analysis with the computation of $\textsc{Opt}(\sigma_1^{\mathrm{S+}})$.

Lemma 6.7. *We have*

$$\textsc{Opt}(\sigma_1^{\mathrm{S+}}) = \frac{\Theta+1}{\Theta-1}.$$

Proof. $\textsc{Opt}$ waits at the origin until time $2\varepsilon'$ and then performs the walk

$$0 \to -\frac{1}{\Theta-1} + \varepsilon' \to 1.$$

$\textsc{Opt}$'s walk is presented in blue in Figure 6.2 for $\Theta = \Theta_{\mathrm{S+}}^{\mathrm{D},\mathbb{R}}$. We show that all requests are served this way: $\textsc{Opt}$ collects s_2 at position $a_1 = -\frac{1}{\Theta-1} + \varepsilon'$ time $\frac{1}{\Theta-1} + \varepsilon'$ and reaches position $a_2 = 1$ at time $\frac{\Theta+1}{\Theta-1} > \frac{1}{\Theta-1} + \varepsilon' = r_1$. Therefore, we have

$$\textsc{Opt}(\sigma_1^{\mathrm{S+}}) = 2\varepsilon' + D\left(0 \to -\frac{1}{\Theta-1} + \varepsilon' \to 1\right) = \frac{\Theta+1}{\Theta-1}. \qquad \square$$

Next, we compute $\textsc{SmarterStart}(\sigma_1^{\mathrm{S+}})$.

Lemma 6.8. *Let $\Theta \in (1,2)$ and $\varepsilon' < \frac{1}{2}$. Then we have*

$$\textsc{SmarterStart}(\sigma_1^{\mathrm{S+}}) = \frac{2\Theta^2 - \Theta + 1}{(\Theta-1)^2} - \frac{2\Theta}{\Theta-1}\varepsilon'.$$

Proof. $\textsc{SmarterStart}$'s walk is presented in green in Figure 6.2 for $\Theta = \Theta_{\mathrm{S+}}^{\mathrm{D},\mathbb{R}}$. For all $t \geq 0$ we have $L(t, 0, (s_1)) = 1$. Thus, $\textsc{SmarterStart}$ starts its first schedule S_1 at time $t_1 = \frac{1}{\Theta-1}$ and reaches position $p_2 = 1$ at time $v_2 = \frac{\Theta}{\Theta-1}$. For $t \geq v_2$, we have

$$L(t, 0, (s_1, s_2)) = D\left(0 \to -\frac{1}{\Theta-1} + \varepsilon' \to 1\right) = \frac{\Theta+1}{\Theta-1} - 2\varepsilon'.$$

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

$$w_2 = \frac{L(t, 0, (s_1, s_2))}{\Theta-1} = \frac{\Theta+1}{(\Theta-1)^2} - \frac{2}{\Theta-1}\varepsilon'.$$

By assumption, we have $\Theta < 2$ and $\varepsilon' < \frac{1}{2}$, which implies that for the time $v_2 = \frac{\Theta}{\Theta-1}$, when $\textsc{SmarterStart}$ reaches position $p_2 = 1$, the inequality

$$w_2 = \frac{\Theta+1}{(\Theta-1)^2} - \frac{2}{\Theta-1}\varepsilon' \overset{\varepsilon'<\frac{1}{2}}{>} \frac{2}{(\Theta-1)^2} \overset{\Theta<2}{>} \frac{\Theta}{\Theta-1} = v_2 \tag{6.20}$$

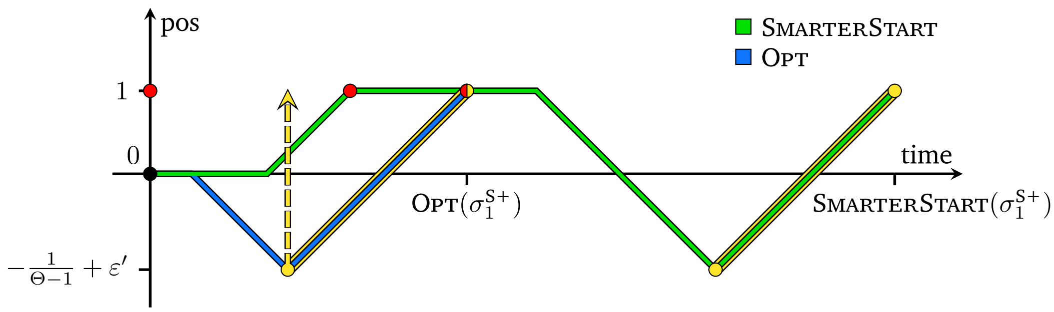

Figure 6.2: SMARTERSTART's and OPT's walk serving σ_1^{S+} with $\varepsilon' = 0.25$ and $\Theta = \Theta_{S+}^{D,\mathbb{R}}$. Request s_1 is red ● and request s_2 is yellow ●.

holds. Note that inequality (6.20) also holds for slightly larger Θ if we let $\varepsilon \to 0$. Because of inequality (6.20), SMARTERSTART has a waiting period and starts the schedule S_2 at time

$$t_2 = \max\{v_2, w_2\} \overset{(6.20)}{=} w_2 = \frac{\Theta+1}{(\Theta-1)^2} - \frac{2}{\Theta-1}\varepsilon'.$$

Serving s_2 from position $p_2 = 1$ takes time

$$L(t_2, p_2, (s_2)) = D\left(1 \to -\frac{1}{\Theta-1} + \varepsilon \to 1\right) = \frac{2\Theta}{\Theta-1} - 2\varepsilon'.$$

To sum it up, we have

$$\text{SMARTERSTART}(\sigma_1^{S+}) = t_2 + L(t_2, p_2, (s_2)) = \frac{2\Theta^2-\Theta+1}{(\Theta-1)^2} - \frac{2\Theta}{\Theta-1}\varepsilon'. \qquad \square$$

Eqipped with Lemmas 6.7 and 6.8, we can compute a lower bound for the competitive of SMARTERSTART for open online DIAL-A-RIDE on the line for $\Theta \in (1,2)$.

Proposition 6.9. *Let $1 < \Theta < 2$. For every sufficiently small $\varepsilon > 0$, we have*

$$\frac{\text{SMARTERSTART}(\sigma_1^{S+})}{\text{OPT}(\sigma_1^{S+})} = \frac{2\Theta^2-\Theta+1}{\Theta^2-1} - \varepsilon = g_1^{\text{up}}(\Theta) - \varepsilon,$$

i.e., the upper bound established in Proposition 6.3 is tight for $\Theta \in (1,2)$.

Proof. Let $\varepsilon > 0$ with $\varepsilon < \frac{\Theta}{\Theta+1}$ and $\varepsilon' = \frac{\Theta+1}{2\Theta}\varepsilon < \frac{1}{2}$. By Lemma 6.8, we have

$$\text{SMARTERSTART}(\sigma_1^{S+}) = \frac{2\Theta^2-\Theta+1}{(\Theta-1)^2} - \frac{2\Theta}{\Theta-1}\varepsilon'.$$

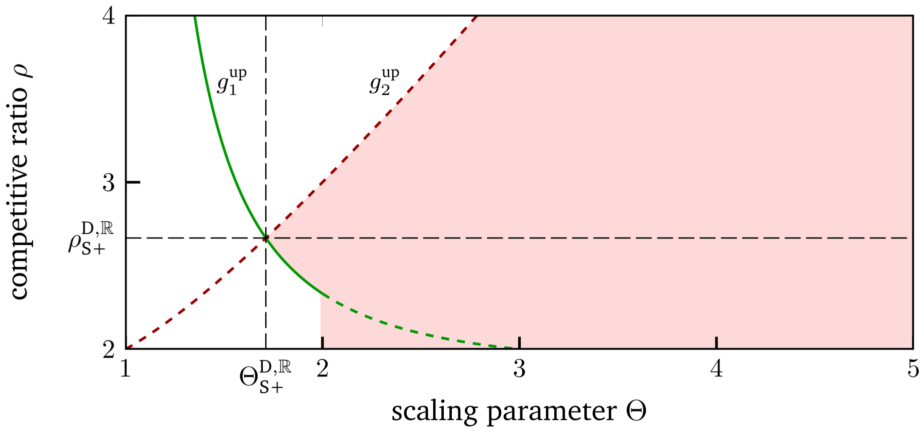

Figure 6.3: Functions g_1^{up} (green) / g_2^{up} (red): upper bounds for competitive ratio for postponing / non-postponing case, drawn solid if tight. Red area: possible values for the competitive ratio, bounded by g_2^{up}.

By Lemma 6.7, we have

$$\textsc{Opt}(\sigma_1^{\mathrm{S+}}) = \frac{\Theta + 1}{\Theta - 1}.$$

Since we have $\varepsilon' = \frac{\Theta+1}{2\Theta}\varepsilon$, we obtain

$$\frac{\textsc{SmarterStart}(\sigma_1^{\mathrm{S+}})}{\textsc{Opt}(\sigma_1^{\mathrm{S+}})} = \frac{2\Theta^2 - \Theta + 1}{\Theta^2 - 1} - \frac{2\Theta}{\Theta + 1}\varepsilon' = \frac{2\Theta^2 - \Theta + 1}{\Theta^2 - 1} - \varepsilon = g_1^{\mathrm{up}}(\Theta) - \varepsilon. \quad \square$$

Figure 6.3 is a visualization of the upper bound for the competitive ratio of online DIAL-A-RIDE on the line presented in Theorem 6.5 together with the lower bound of Proposition 6.9. Next, we examine the request sequence $\sigma_2^{\mathrm{S+}}$.

Definition 6.10. Let $\varepsilon' > 0$ with $\varepsilon' < \frac{\Theta}{\Theta-1}$. We define

$$\sigma_2^{\mathrm{S+}} = \{s_1, s_2^{(1)}, s_2^{(2)}, s_3\}$$

with

$$s_1 = (1, 1; 0),$$
$$s_2^{(1)} = \left(\frac{2\Theta - 1}{\Theta - 1} - \varepsilon', \frac{2\Theta - 1}{\Theta - 1} - \varepsilon'; \frac{1}{\Theta - 1} + \varepsilon'\right),$$

$$\begin{aligned}
s_2^{(2)} &= \left(-\frac{1}{\Theta-1}, -\frac{1}{\Theta-1}; \frac{1}{\Theta-1}+\varepsilon'\right),\\
s_3 &= \left(\max\left\{\frac{3}{(\Theta-1)^2}, \frac{2\Theta-1}{\Theta-1}\right\}-\varepsilon', \max\left\{\frac{3}{(\Theta-1)^2}, \frac{2\Theta-1}{\Theta-1}\right\}-\varepsilon'; \frac{2\Theta+1}{(\Theta-1)^2}\right).
\end{aligned}$$

Note $\varepsilon' < \frac{\Theta}{\Theta-1}$, which implies $a_3 \geq a_2^{(1)} = \frac{2\Theta-1}{\Theta-1} - \varepsilon' > 1 = a_1$, i.e., the positions of $a_2^{(1)}$ and a_3 are to the right of position $a_1 = 1$. We start the examination of $\sigma_2^{\mathsf{S}+}$ with computing $\textsc{Opt}(\sigma_2^{\mathsf{S}+})$.

Lemma 6.11. *We have*

$$\textsc{Opt}(\sigma_2^{\mathsf{S}+}) = \max\left\{\frac{2\Theta+1}{(\Theta-1)^2}, \frac{2\Theta+1}{\Theta-1}\right\}.$$

Proof. Opt waits at the origin until time ε' and then performs the walk

$$0 \to -\frac{1}{\Theta-1} \to \max\left\{\frac{3}{(\Theta-1)^2}, \frac{2\Theta-1}{\Theta-1}\right\} - \varepsilon'.$$

Opt's walk is presented in blue in Figure 6.4 for $\Theta = \Theta_{\mathsf{S}+}^{\mathsf{D},\mathbb{R}}$ and in blue in Figure 6.5 for $\Theta = 2.5$. We show that all requests are seved this way: Opt serves request $s_2^{(2)}$ at time $r_2^{(2)} = \frac{1}{\Theta-1}+\varepsilon'$. The release time of $s_2^{(1)}$ is the same as of $s_2^{(2)}$ and thus Opt can serve $s_2^{(2)}$ at arrival. Opt reaches position $a_3 = \max\left\{\frac{3}{(\Theta-1)^2}, \frac{2\Theta-1}{\Theta-1}\right\} - \varepsilon'$ at time $\max\left\{\frac{2\Theta+1}{(\Theta-1)^2}, \frac{2\Theta+1}{\Theta-1}\right\}$, which is after time r_3, i.e., request s_3 is also served at arrival. To sum it up, we have

$$\begin{aligned}
\textsc{Opt}(\sigma_2^{\mathsf{S}+}) &= \varepsilon' + D\left(0 \to -\frac{1}{\Theta-1} \to \max\left\{\frac{3}{(\Theta-1)^2}, \frac{2\Theta-1}{\Theta-1}\right\} - \varepsilon'\right)\\
&= \max\left\{\frac{2\Theta+1}{(\Theta-1)^2}, \frac{2\Theta+1}{\Theta-1}\right\}.
\end{aligned}$$

□

Next, we compute SmarterStart's completion time.

Lemma 6.12. *Let* $\frac{1}{2}(1+\sqrt{5}) \leq \Theta \leq 3$ *and* $\varepsilon' < \frac{1}{2}$. *Then, we have*

$$\textsc{SmarterStart}(\sigma_2^{\mathsf{S}+}) = \max\left\{\frac{3\Theta^2+3}{(\Theta-1)^2}, \frac{5\Theta^2-3\Theta+1}{(\Theta-1)^2}\right\} - \frac{3\Theta-2}{\Theta-1}\varepsilon'.$$

Proof. SmarterStart's walk is presented in green in Figure 6.4 for $\Theta = \Theta_{\mathsf{S}+}^{\mathsf{D},\mathbb{R}}$ and in green in Figure 6.5 for $\Theta = 2.5$. For all $t \geq 0$, we have $L(t, 0, (s_1)) = 1$. Thus, SmarterStart starts its first schedule S_1 at time $t_1 = \frac{1}{\Theta-1}$ and reaches position $p_2 = 1$ at time $v_2 = \frac{\Theta}{\Theta-1}$. For $t \geq v_2$ we have

$$L(t, 0, (s_1, s_2^{(1)}, s_2^{(2)})) = D\left(0 \to -\frac{1}{\Theta - 1} \to \frac{2\Theta - 1}{\Theta - 1} - \varepsilon'\right) = \frac{2\Theta + 1}{\Theta - 1} - \varepsilon'.$$

Thus, the second schedule S_2 is not started before time

$$w_2 = \frac{L\left(t, 0, (s_1, s_2^{(1)}, s_2^{(2)})\right)}{\Theta - 1} = \frac{2\Theta + 1}{(\Theta - 1)^2} - \frac{\varepsilon'}{\Theta - 1}.$$

By assumption, we have $\Theta \leq 3$ and $\varepsilon' < \frac{1}{2}$, which implies that for the time $v_2 = \frac{\Theta}{\Theta-1}$, when SMARTERSTART reaches position $p_2 = 1$, the inequality

$$w_2 = \frac{2\Theta + 1}{(\Theta - 1)^2} - \frac{\varepsilon'}{\Theta - 1} \overset{\varepsilon' < \frac{1}{2}}{>} \frac{\frac{3}{2}\Theta + \frac{3}{2}}{(\Theta - 1)^2} \overset{\Theta \leq 3}{\geq} \frac{\Theta}{\Theta - 1} = v_2 \tag{6.21}$$

holds. Note that inequality (6.21) also holds for slightly larger Θ if we let $\varepsilon \to 0$. Because of inequality (6.21), SMARTERSTART has a waiting period and starts the schedule S_2 at time

$$t_2 = \max\{v_2, w_2\} \overset{(6.21)}{=} w_2 = \frac{2\Theta + 1}{(\Theta - 1)^2} - \frac{\varepsilon'}{\Theta - 1}$$

before the request s_3 is released. If SMARTERSTART serves $s_2^{(2)}$ before serving $s_2^{(1)}$ the time it needs is at least

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

The best schedule that serves $s_2^{(2)}$ after serving $s_2^{(1)}$ needs time

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

Thus, SMARTERSTART serves $s_2^{(2)}$ after serving $s_2^{(1)}$ and finishes S_2 at position $p_3 = -\frac{1}{\Theta-1}$ at time

$$v_3 = t_2 + L(t_2, p_2, (s_2^{(1)}, s_2^{(2)})) = \frac{3\Theta^2 - \Theta + 1}{(\Theta - 1)^2} - \frac{2\Theta - 1}{\Theta - 1}\varepsilon'.$$

For all $t \geq v_3$ we have

$$L(t, 0, (s_1, s_2^{(1)}, s_2^{(2)}, s_3)) = D\left(0 \to -\frac{1}{\Theta - 1} \to \max\left\{\frac{3}{(\Theta - 1)^2}, \frac{2\Theta - 1}{\Theta - 1}\right\} - \varepsilon'\right)$$

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

Therefore the final schedule is not started before time

$$w_3 = \frac{L(t,0,(s_1,s_2^{(1)},s_2^{(2)},s_3))}{\Theta-1} = \max\left\{\frac{2\Theta+1}{(\Theta-1)^3}, \frac{2\Theta+1}{(\Theta-1)^2}\right\} - \frac{\varepsilon'}{\Theta-1}.$$

However, by assumption, we have $\Theta \geq \frac{1}{2}\big(1+\sqrt{5}\big)$ and $\varepsilon' < \frac{1}{2}$, which implies

$$\begin{aligned}
v_3 &= \frac{3\Theta^2-\Theta+1}{(\Theta-1)^2} - \frac{2\Theta-1}{\Theta-1}\varepsilon' \\
&= \frac{3\Theta^2-\Theta+1}{(\Theta-1)^2} - 2\varepsilon' - \frac{\varepsilon'}{\Theta-1} \\
&\overset{\varepsilon'<\frac{1}{2}}{>} \frac{2\Theta^2+\Theta}{(\Theta-1)^2} - \frac{\varepsilon'}{\Theta-1} \\
&\overset{\Theta\geq\frac{1}{2}(1+\sqrt{5})}{\geq} \max\left\{\frac{2\Theta+1}{(\Theta-1)^3}, \frac{2\Theta+1}{(\Theta-1)^2}\right\} - \frac{\varepsilon'}{\Theta-1} \\
&= w_3,
\end{aligned} \tag{6.22}$$

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

$$t_3 = \max\{v_3, w_3\} \overset{(6.22)}{=} v_3 = \frac{3\Theta^2-\Theta+1}{(\Theta-1)^2} - \frac{2\Theta-1}{\Theta-1}\varepsilon'.$$

The schedule S_3 needs time

$$\begin{aligned}
L(t_3,p_3,(s_3)) &= D\left(\frac{1}{\Theta-1} \to \max\left\{\frac{3}{(\Theta-1)^2}, \frac{2\Theta-1}{\Theta-1}\right\} - \varepsilon'\right) \\
&= \max\left\{\frac{\Theta+2}{(\Theta-1)^2}, \frac{2\Theta}{\Theta-1}\right\} - \varepsilon'.
\end{aligned}$$

To sum it up, we have

$$\textsc{SmarterStart}(\sigma_2^{\mathrm{S}+}) = t_3 + L(t_3,p_3,(s_3)) = \max\left\{\frac{3\Theta^2+3}{(\Theta-1)^2}, \frac{5\Theta^2-3\Theta+1}{(\Theta-1)^2}\right\} - \frac{3\Theta-2}{\Theta-1}\varepsilon'.$$

□

Eqipped with Lemmas 6.11 and 6.12, we can compute lower bounds for the competitive of SmarterStart for open online Dial-a-Ride on the line for $\Theta \in [\frac{1}{2}(1+\sqrt{5}), 3]$. We begin with the subinterval $[\frac{1}{2}(1+\sqrt{5}), 2]$.

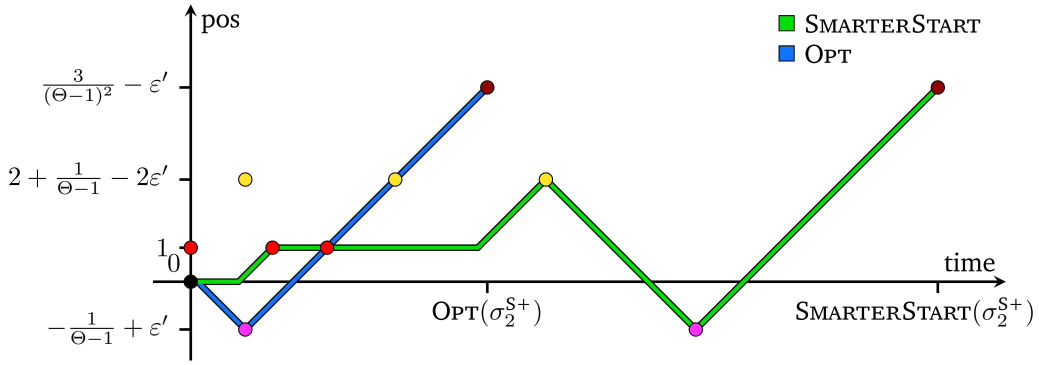

Figure 6.4: SMARTERSTART's and OPT's walk serving σ_2^{S+} with $\varepsilon' = 0.2$ and $\Theta = \Theta_{S+}^{D,\mathbb{R}}$. Request s_1 is red ●, request $s_2^{(1)}$ is yellow ●, request $s_2^{(2)}$ is violet ● and request s_3 is brown ●.

Proposition 6.13. *Let $\frac{1}{2}(1+\sqrt{5}) \leq \Theta \leq 2$. For every sufficiently small $\varepsilon > 0$ we have*

$$\frac{\text{SMARTERSTART}(\sigma_2^{S+})}{\text{OPT}(\sigma_2^{S+})} = \frac{3\Theta^2+3}{2\Theta+1} - \varepsilon = g_2^{\text{up}}(\Theta) - \varepsilon,$$

i.e., the upper bound established in Proposition 6.4 is tight for $\Theta \in [\frac{1}{2}(1+\sqrt{5}), 2] \approx [1.6180, 2]$.

Proof. Let $\varepsilon > 0$ with $\varepsilon < \frac{1}{2}(\frac{3\Theta^2-5\Theta+2}{2\Theta+1})$ and $\varepsilon' = \frac{2\Theta+1}{3\Theta^2-5\Theta+2}\varepsilon < \frac{1}{2}$. By Lemma 6.12, we have

$$\begin{aligned} \text{SMARTERSTART}(\sigma_2^{S+}) &\overset{\text{Lem 6.12}}{=} \max\left\{\frac{3\Theta^2+3}{(\Theta-1)^2}, \frac{5\Theta^2-3\Theta+1}{(\Theta-1)^2}\right\} - \frac{3\Theta-2}{\Theta-1}\varepsilon' \\ &\overset{\Theta \leq 2}{=} \frac{3\Theta^2+3}{(\Theta-1)^2} - \frac{3\Theta-2}{\Theta-1}\varepsilon'. \end{aligned}$$

Lemma 6.11 implies

$$\text{OPT}(\sigma_2^{S+}) = \max\left\{\frac{2\Theta+1}{(\Theta-1)^2}, \frac{2\Theta+1}{\Theta-1}\right\} \overset{\Theta \leq 2}{=} \frac{2\Theta+1}{(\Theta-1)^2}.$$

Since we have $\varepsilon' = \frac{2\Theta+1}{3\Theta^2-5\Theta+2}\varepsilon$, we finally obtain

$$\frac{\text{SMARTERSTART}(\sigma_2^{S+})}{\text{OPT}(\sigma_2^{S+})} = \frac{3\Theta^2+3}{2\Theta+1} - \frac{3\Theta^2-5\Theta+2}{2\Theta+1}\varepsilon' = \frac{3\Theta^2+3}{2\Theta+1} - \varepsilon = g_2^{\text{up}}(\Theta) - \varepsilon. \qquad \square$$

Next, we compute a lower bound for the competitive of SMARTERSTART for open online DIAL-A-RIDE on the line for $\Theta \in (2, 3]$.

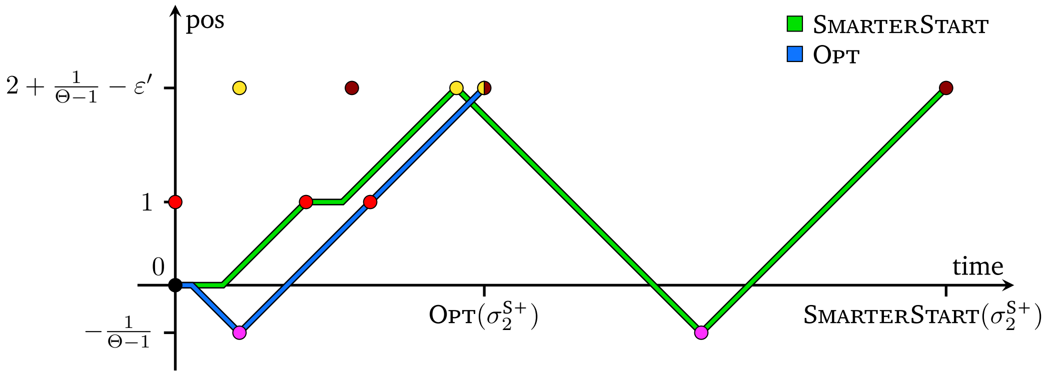

Figure 6.5: SMARTERSTART's and OPT's walk serving σ_2^{S+} with $\varepsilon' = 0.2$ and $\Theta = 2.5$. Request s_1 is red, request $s_2^{(1)}$ is yellow, request $s_2^{(2)}$ is violet and request s_3 is brown.

Lemma 6.14. *Let $2 < \Theta \le 3$ and $\varepsilon > 0$ sufficiently small. Then, we have*

$$\frac{\textsc{SmarterStart}(\sigma_2^{S+})}{\textsc{Opt}(\sigma_2^{S+})} = \frac{5\Theta^2 - 3\Theta + 1}{2\Theta^2 - \Theta - 1} - \varepsilon =: g_2^{\text{low}}(\Theta) - \varepsilon.$$

In particular, we have

$$\frac{\textsc{SmarterStart}(\sigma_2^{S+})}{\textsc{Opt}(\sigma_2^{S+})} > \rho_{S+}^{D,\mathbb{R}} \approx 2.6662$$

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

Proof. Let $\varepsilon > 0$ with $\varepsilon < \min\{\frac{1}{2}(\frac{3\Theta-2}{2\Theta+1}), \frac{1}{10}\}$ and $\varepsilon' = \frac{2\Theta+1}{3\Theta-2}\varepsilon < \frac{1}{2}$. By Lemma 6.12, we have

$$\begin{aligned}\textsc{SmarterStart}(\sigma_2^{S+}) &\overset{\text{Lem 6.12}}{=} \max\left\{\frac{3\Theta^2+3}{(\Theta-1)^2}, \frac{5\Theta^2-3\Theta+1}{(\Theta-1)^2}\right\} - \frac{3\Theta-2}{\Theta-1}\varepsilon' \\ &\overset{\Theta \ge 2}{=} \frac{5\Theta^2-3\Theta+1}{(\Theta-1)^2} - \frac{3\Theta-2}{\Theta-1}\varepsilon'.\end{aligned}$$

By Lemma 6.11 we have

$$\textsc{Opt}(\sigma_2^{S+}) = \max\left\{\frac{2\Theta+1}{(\Theta-1)^2}, \frac{2\Theta+1}{\Theta-1}\right\} \overset{\Theta\ge 2}{=} \frac{2\Theta+1}{\Theta-1}.$$

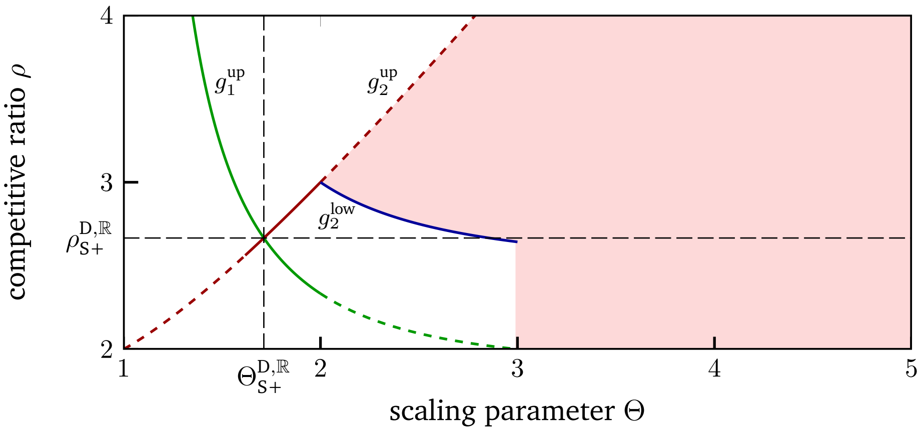

Figure 6.6: Functions g_1^{up} (green) / g_2^{up} (red): upper bounds for competitive ratio for postponing / non-postponing case, drawn solid if tight. Function: g_2^{low} (blue): lower bound for competitive ratio. Red area: possible values for the competitive ratio, bounded by g_2^{up} and g_2^{low}.

Since we have $\varepsilon' = \frac{2\Theta+1}{3\Theta-2}\varepsilon$, we finally obtain

$$\frac{\text{SMARTERSTART}(\sigma_2^{\text{S+}})}{\text{OPT}(\sigma_2^{\text{S+}})} = \frac{5\Theta^2 - 3\Theta + 1}{2\Theta^2 - \Theta - 1} - \frac{3\Theta - 2}{2\Theta + 1}\varepsilon' = \frac{5\Theta^2 - 3\Theta + 1}{2\Theta^2 - \Theta - 1} - \varepsilon = g_2^{\text{low}}(\Theta) - \varepsilon,$$

as claimed. The function g_2^{low} is monotonically increasing on $(2, 1+\sqrt{2}]$. Therefore, we have

$$\frac{\text{SMARTERSTART}(\sigma_2^{\text{S+}})}{\text{OPT}(\sigma_2^{\text{S+}})} - \varepsilon > g_2^{\text{low}}(1+\sqrt{2}) - \varepsilon = \frac{11}{\sqrt{2}} - 5 - \varepsilon > \rho_{\text{S+}}^{\text{D},\mathbb{R}}$$

for all $\Theta \in (2, 1+\sqrt{2}]$ and $\varepsilon < \frac{1}{10}$. $\square$

Figure 6.6 is a visualization of the upper bound for the competitive ratio of online DIAL-A-RIDE on the line presented in Theorem 6.5 together with the lower bounds of Propositions 6.9 and 6.13 as well as Lemma 6.14.

Recall that the optimal parameter $\Theta_{\text{S+}}^{\text{D},\mathbb{R}}$ established in Theorem 6.5 is the only positive, real solution of the equation

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

which is $\Theta_{\text{S+}}^{\text{D},\mathbb{R}} \approx 1.7125$. Therefore, according to Proposition 6.9 and Proposition 6.13 the parameter $\Theta_{\text{S+}}^{\text{D},\mathbb{R}}$ lies in the interval where the upper bounds of Propositions 6.3 and 6.4

are both tight. It remains to make sure that for all Θ that lie outside of this interval the competitive ratio of SMARTERSTART is larger than $\rho_{\mathrm{S+}}^{\mathrm{D},\mathbb{R}} \approx 2.6662$. For this, we examine the request sequence $\sigma_3^{\mathrm{S+}}$.

Definition 6.15. Let $\Theta > 1 + \sqrt{2}$ and $\varepsilon' > 0$ with $\varepsilon' < \min\{\frac{\Theta}{2\Theta-2}, \frac{1}{\Theta-1}\}$. We define

$$\sigma_3^{\mathrm{S+}} := \{s_1, s_2^{(1)}, s_2^{(2)}, s_3\}$$

with

$$\begin{aligned} s_1 &= (1, 1; 0), \\ s_2^{(1)} &= \left(\frac{\Theta - 2}{2\Theta - 2} + \varepsilon', 1; \frac{1}{\Theta - 1} + \varepsilon'\right), \\ s_2^{(2)} &= \left(-\frac{1}{\Theta - 1} + \varepsilon', -\frac{1}{\Theta - 1} + \varepsilon'; \frac{1}{\Theta - 1} + \varepsilon'\right), \\ s_3 &= \left(1, 1; \frac{\Theta + 1}{\Theta - 1}\right). \end{aligned}$$

We have $\varepsilon' < \frac{\Theta}{2\Theta-2}$, which implies $0 < \frac{\Theta-2}{2\Theta-2} + \varepsilon' < 1$ for $\Theta > 1 + \sqrt{2}$, i.e., the starting position of $s_2^{(1)}$ is between 0 and 1. Furthermore, we have $\varepsilon' < \frac{1}{\Theta-1}$, which implies $a_2^{(2)} = -\frac{1}{\Theta-1} + \varepsilon' < 0$, i.e., the starting position of $s_2^{(2)}$ is to the left of the origin. We start the examination of $\sigma_3^{\mathrm{S+}}$ with computing $\textsc{Opt}(\sigma_3^{\mathrm{S+}})$.

Lemma 6.16. *We have*

$$\textsc{Opt}(\sigma_3^{\mathrm{S+}}) = \frac{\Theta + 1}{\Theta - 1}.$$

Proof. OPT waits at the origin until time $2\varepsilon'$ and then performs the walk

$$0 \to -\frac{1}{\Theta - 1} + \varepsilon' \to 1.$$

An illustration of OPT's walk is presented in blue in Figure 6.7 for $\Theta = 2.75$. We show that all requests are served this way: OPT serves request $s_2^{(2)}$ at time $r_2^{(2)} = \frac{1}{\Theta-1} + \varepsilon'$. When OPT returns to the origin at time $\frac{2}{\Theta-1}$, the requests s_1 and $s_2^{(1)}$ already have been released and can thus be served on the way to position 1. OPT reaches position 1 at time $\frac{\Theta+1}{\Theta-1}$ and can thus serve s_3 at arrival. To sum it up, we have

$$\textsc{Opt}(\sigma_3^{\mathrm{S+}}) = 2\varepsilon' + D\left(0 \to -\frac{1}{\Theta - 1} + \varepsilon' \to 1\right) = \frac{\Theta + 1}{\Theta - 1}.$$

OPT can do this even if $c = 1$ since $s_2^{(1)}$ is the only transportation request and no other request lies between its starting position and destination. □

Next, we compute SMARTERSTART's completion time.

Lemma 6.17. *Let $\Theta > 1 + \sqrt{2}$. Then, we have*

$$\text{SMARTERSTART}(\sigma_3^{\text{S+}}) = \frac{4\Theta}{\Theta - 1} - 4\varepsilon'.$$

Proof. SMARTERSTART's walk is presented in green in Figure 6.7 for $\Theta = 2.75$. For all $t \geq 0$, we have $L(t, 0, (s_1)) = 1$. Thus, SMARTERSTART waits at position $p_1 = 0$ at least until time $\frac{1}{\Theta-1}$. Since no other request are released until SMARTERSTART's waiting period is over the first schedule S_1 is started at time $t_1 = \frac{1}{\Theta-1}$. SMARTERSTART reaches position $p_2 = 1$ at time $v_2 = \frac{\Theta}{\Theta-1}$. For $t \geq v_2$ we have

$$L(t, 0, (s_1, s_2^{(1)}, s_2^{(2)})) = D\left(0 \to -\frac{1}{\Theta - 1} + \varepsilon' \to 1\right) = \frac{\Theta + 1}{\Theta - 1} - 2\varepsilon'.$$

Thus, SMARTERSTART postpones the second schedule S_2 at least until

$$w_2 = \frac{\Theta + 1}{(\Theta - 1)^2} - \frac{2}{\Theta - 1}\varepsilon'.$$

By assumption, we have $\Theta > 1 + \sqrt{2}$, which implies

$$w_2 = \frac{\Theta + 1}{(\Theta - 1)^2} - \frac{2}{\Theta - 1}\varepsilon' \overset{\Theta > 1+\sqrt{2}}{<} \frac{\Theta}{\Theta - 1} = v_2. \tag{6.23}$$

Because of inequality (6.23), SMARTERSTART starts schedule S_2 at time

$$t_2 = \max\{v_2, w_2\} \overset{(6.23)}{=} v_2 = \frac{\Theta}{\Theta - 1},$$

which is before request s_3 is released. The shortest schedule serving $s_2^{(2)}$ before serving $s_2^{(1)}$ has length

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

The shortest schedule that serves $s_2^{(2)}$ after serving $s_2^{(1)}$ has length

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

Thus, SMARTERSTART serves $s_2^{(2)}$ after serving $s_2^{(1)}$ and finishes S_2 at position $p_3 = -\frac{1}{\Theta-1}+\varepsilon'$ at time

$$v_3 = t_2 + L(t_2, p_2, (s_2^{(1)}, s_2^{(2)})) = \frac{3\Theta}{\Theta-1} - 3\varepsilon'$$

after s_3 is released. We have for all $t \geq v_3$ the equation

$$L(t, 0, (s_1, s_2^{(1)}, s_2^{(2)}, s_3)) = D\left(0 \to -\frac{1}{\Theta-1}\varepsilon' \to 1\right) = \frac{\Theta+1}{\Theta-1} - 2\varepsilon'.$$

Therefore the final schedule is not started before time

$$w_3 = \frac{\Theta+1}{(\Theta-1)^2} - \frac{2}{\Theta-1}\varepsilon',$$

which is equal to w_2 and thus smaller than t_3, which again is smaller than v_3. Therefore, the starting time of the schedule S_3 is the ending time of the schedule S_2 and we have

$$t_3 = v_3 = \frac{3\Theta}{\Theta-1} - 3\varepsilon'.$$

The schedule S_3 has length

$$L(t_3, p_3, (s_3)) = D\left(-\frac{1}{\Theta-1} + \varepsilon \to 1\right) = \frac{\Theta}{\Theta-1} - \varepsilon'.$$

To sum it up, we have

$$\textsc{SmarterStart}(\sigma_3^{\mathsf{S+}}) = t_3 + L(t_3, p_3, (s_3)) = \frac{4\Theta}{\Theta-1} - 4\varepsilon'. \qquad \square$$

Equipped with the Lemmas 6.16 and 6.17, we can compute a lower bound for the competitive ratio of SMARTERSTART for open online DIAL-A-RIDE on the line for $\Theta > 1+\sqrt{2}$.

Lemma 6.18. *Let $\Theta > 1+\sqrt{2}$ and $\varepsilon > 0$ sufficiently small. Then, we have*

$$\frac{\textsc{SmarterStart}(\sigma_3^{\mathsf{S+}})}{\textsc{Opt}(\sigma_3^{\mathsf{S+}})} = \frac{4\Theta}{\Theta+1} - \varepsilon =: g_3^{\text{low}}(\Theta) - \varepsilon.$$

In particular, we have

$$\frac{\textsc{SmarterStart}(\sigma_3^{\mathsf{S+}})}{\textsc{Opt}(\sigma_3^{\mathsf{S+}})} > \rho_{\mathsf{S+}}^{\mathrm{D},\mathbb{R}} \approx 2.6662$$

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

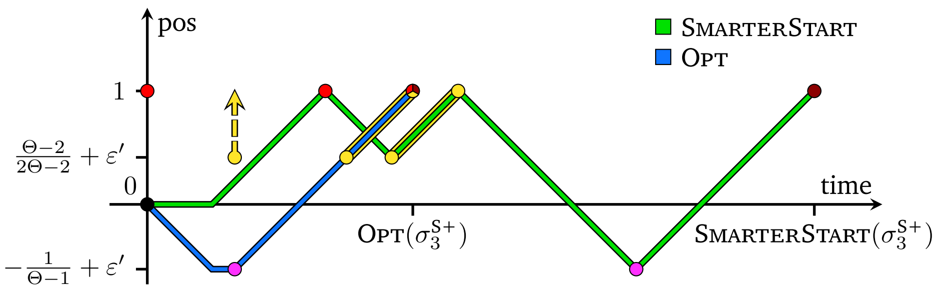

Figure 6.7: SMARTERSTART's and OPT's walk serving $\sigma_3^{\text{S+}}$ with $\varepsilon' = 0.2$ and $\Theta = 2.75$. Request s_1 is red ●, request $s_2^{(1)}$ is yellow ●, request $s_2^{(2)}$ is violet ● and request s_3 is brown ●.

Proof. Let $\varepsilon > 0$ with $\varepsilon < \frac{1}{10}$ and $\varepsilon' = \frac{\Theta+1}{4\Theta-4}$. By Lemma 6.17, we have

$$\text{SMARTERSTART}(\sigma_3^{\text{S+}}) \overset{\text{Lem 6.17}}{=} \frac{4\Theta}{\Theta-1} - 4\varepsilon'.$$

By Lemma 6.16, we have

$$\text{OPT}(\sigma_3^{\text{S+}}) = \frac{\Theta+1}{\Theta-1}.$$

Since we have $\varepsilon' = \frac{\Theta+1}{4\Theta-4}\varepsilon$, we finally obtain

$$\frac{\text{SMARTERSTART}(\sigma_3^{\text{S+}})}{\text{OPT}(\sigma_3^{\text{S+}})} = \frac{4\Theta}{\Theta+1} - \frac{4\Theta-4}{\Theta+1}\varepsilon' = \frac{4\Theta}{\Theta+1} - \varepsilon = g_3^{\text{low}}(\Theta) - \varepsilon.$$

The function g_3^{low} is monotonically increasing on $[1+\sqrt{2}, \infty)$. Therefore, we have

$$\frac{\text{SMARTERSTART}(\sigma_3^{\text{S+}})}{\text{OPT}(\sigma_3^{\text{S+}})} - \varepsilon > g_3^{\text{low}}(1+\sqrt{2}) - \varepsilon = 2\sqrt{2} - \varepsilon > \rho_{\text{S+}}^{\text{D},\mathbb{R}}$$

for all $\Theta \in (1+\sqrt{2}, \infty)$ and $\varepsilon < \frac{1}{10}$. □

We combine all lower bounds constructed in this section into one general lower bound. See Figure 6.8 for an illustration of all upper and lower bounds for the competitive ratio of online DIAL-A-RIDE on the line.

Theorem 6.19. *Let $G_{\text{DaR}} : \mathbb{R}_{>1} \to \mathbb{R}_{>1}$ be a function with*

$$G_{\text{DaR}}(\Theta) := \begin{cases} g_1^{\text{up}}(\Theta), & \text{for } \Theta \in (1, \Theta_{\text{S+}}^{\text{D},\mathbb{R}}], \\ g_2^{\text{up}}(\Theta), & \text{for } \Theta \in (\Theta_{\text{S+}}^{\text{D},\mathbb{R}}, 2], \\ g_2^{\text{low}}(\Theta), & \text{for } \Theta \in (2, 1+\sqrt{2}), \\ g_3^{\text{low}}(\Theta), & \text{for } \Theta \in [1+\sqrt{2}, \infty). \end{cases}$$

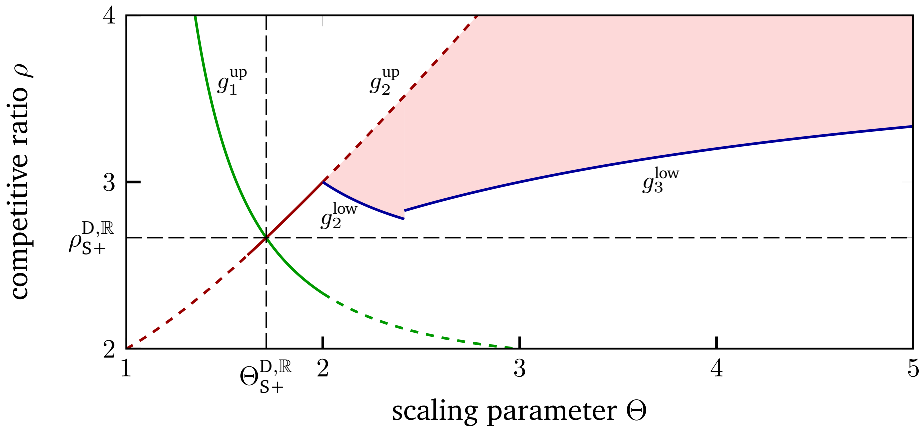

Figure 6.8: Functions g_1^{up} (green) / g_2^{up} (red): upper bounds for competitive ratio for postponing / non-postponing case, drawn solid if tight. Functions: g_2^{low} and g_3^{low} (blue): lower bounds for competitive ratio. Red area: possible values for the competitive ratio, bounded by g_2^{up} and g_2^{low} / g_3^{low}.

Then G_{DaR} is general lower bound for the competitive ratio of SmarterStart *for online* Dial-a-Ride *on the line. The unique minimum of G_{DaR} lies in $\Theta = \Theta_{\text{S+}}^{\text{D},\mathbb{R}}$ and yields a lower bound of*

$$G_{\text{DaR}}(\Theta_{\text{S+}}^{\text{D},\mathbb{R}}) = \rho_{\text{S+}}^{\text{D},\mathbb{R}} \approx 2.6662.$$

Proof. We have shown in Proposition 6.9 that $g_1^{\text{up}}(\Theta)$ with $\Theta \in (1, \Theta_{\text{S+}}^{\text{D},\mathbb{R}}]$ is a lower bound for the competitive ratio of SmarterStart for online Dial-a-Ride and in Proposition 6.13 that $g_2^{\text{up}}(\Theta)$ with $\Theta \in (\Theta_{\text{S+}}^{\text{D},\mathbb{R}}, 2]$ is a lower bound. Theorem 6.5 implies that G_{DaR} has unique minimum in the interval $(1, 2]$ at $\Theta = \Theta_{\text{S+}}^{\text{D},\mathbb{R}}$. It remains to show that $G_{\text{DaR}}(\Theta) > G_{\text{DaR}}(\Theta_{\text{S+}}^{\text{D},\mathbb{R}})$ for all $\Theta \in (2, \infty)$. This immediately follows from Lemmas 6.14 and 6.18. □

The main theorem of this section follows by combining Theorem 6.5 and Theorem 6.19.

Theorem 6.20. *The competitive ratio of* SmarterStart *for open online* Dial-a-Ride *on the line with scaling parameter $\Theta_{\text{S+}}^{\text{D},\mathbb{R}} \approx 1.7125$ is exactly*

$$\rho_{\text{S+}}^{\text{D},\mathbb{R}} = g_1^{\text{up}}(\Theta_{\text{S+}}^{\text{D},\mathbb{R}}) = g_2^{\text{up}}(\Theta_{\text{S+}}^{\text{D},\mathbb{R}}) \approx 2.6662.$$

For every scaling parameter $\Theta > 1$ with $\Theta \neq \Theta_{\text{S+}}^{\text{D},\mathbb{R}}$ the competitive ratio of SmarterStart *is strictly larger than $\rho_{\text{S+}}^{\text{D},\mathbb{R}}$.*

6.3 Bounds for the Open Online TSP on the Line

In the case of SMARTSTART, the DIAL-A-RIDE and the TSP version of the algorithm had different competitive ratios. This is also true for SMARTERSTART. Since online TSP is a special case of online DIAL-A-RIDE, all upper bounds, i.e., the bounds provided by Proposition 6.3 and Proposition 6.4 on the competitive ratio of SMARTERSTART for open online DIAL-A-RIDE on the line are also valid for open online TSP on the line. However, of the lower bounds, only the bounds obtained by request sequences without transportation requests are valid for open online TSP. To be more precise, only the bounds given by the request sequence $\sigma_2^{\text{S}+}$ are valid, while the bounds given by $\sigma_1^{\text{S}+}$, $\sigma_3^{\text{S}+}$ are not. Therefore, we have only the lower bound $g_2^{\text{up}}(\Theta)$ for $\Theta \in [\frac{1}{2}(1+\sqrt{5}), 2]$ that tightly matches the upper bound from Proposition 6.4 for the case that the final schedule is not postponed as well as the lower bound $g_2^{\text{low}}(\Theta)$ for $\Theta \in (2, 3]$. We will see that the upper bound given in Proposition 6.3 for the case that the final schedule is postponed is not tight for online TSP. The reason for this is that online TSP allows a smaller bound for the length of a schedule.

Lemma 6.21. *For every schedule S_j of* SMARTERSTART *for open online* TSP*, we have*

$$L(t_j, p_j, \sigma_j) \leq \left(1 + \frac{\Theta - 1}{2\Theta + 1}\right)\textsc{Opt}(\sigma).$$

Proof. By Lemma 4.5 and Lemma 6.1 we have

$$\begin{aligned} L(t_j, p_j, \sigma_j) &\overset{\text{Lem 4.6}}{\leq} \min\left\{\textsc{Opt}(\sigma) + d(p_j, 0), \frac{3}{2}(\textsc{Opt}(\sigma) - t_{j-1})\right\} \\ &\overset{\text{Lem 6.1}}{\leq} \min\left\{\textsc{Opt}(\sigma) + d(p_j, 0), \frac{3}{2}\left(\textsc{Opt}(\sigma) - \frac{1}{\Theta - 1}d(p_j, 0)\right)\right\} \\ &\leq \left(1 + \frac{\Theta - 1}{2\Theta + 1}\right)\textsc{Opt}(\sigma) \end{aligned}$$

since the minimum above is largest if the two terms are equal, which is the case for $d(p_j, 0) = \frac{\Theta-1}{2\Theta+1}\textsc{Opt}(\sigma)$. □

Using Lemma 6.21, we can improve the bound of Proposition 6.3.

Proposition 6.22. *In case* SMARTERSTART *for open online* TSP *postpones executing S_N, we have*

$$\frac{\textsc{SmarterStart}(\sigma)}{\textsc{Opt}(\sigma)} \leq \frac{3\Theta^2 - \Theta + 1}{2\Theta^2 - \Theta - 1} =: g_{1,\text{TSP}}^{\text{up}}(\Theta).$$

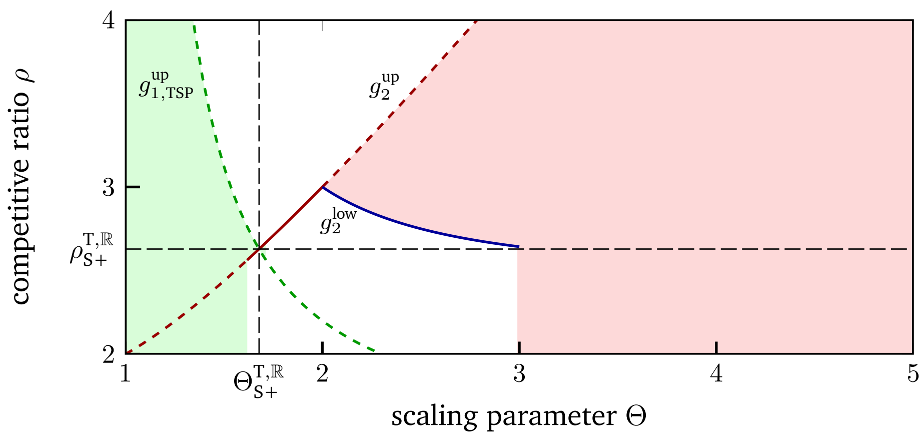

Figure 6.9: Functions $g_{1,\mathrm{TSP}}^{\mathrm{up}}$ (green) / g_2^{up} (red): upper bounds for competitive ratio for postponing / non-postponing case, drawn solid if tight. Function: g_2^{low} (blue): lower bound for competitive ratio. Green / red area: possible values for the competitive ratio, bounded by g_2^{up} and g_2^{low}.

Proof. Assume SMARTERSTART postpones the final schedule. Then Lemma 6.21 yields the claimed bound:

$$\begin{aligned}
\textsc{SmarterStart}(\sigma) &\overset{(4.1)}{=} t_N + L(t_N, p_N, \sigma_N) \\
&\overset{(6.2)}{=} \frac{1}{\Theta - 1} L(t_N, 0, \sigma_{\leq t_N}) + L(t_N, p_N, \sigma_N) \\
&\overset{(4.4)}{\leq} \frac{1}{\Theta - 1} \textsc{Opt}(\sigma) + L(t_N, p_N, \sigma_N) \\
&\overset{\text{Lem 6.21}}{\leq} \left(\frac{1}{\Theta - 1} + 1 + \frac{\Theta - 1}{2\Theta + 1} \right) \textsc{Opt}(\sigma) \\
&= \frac{3\Theta^2 - \Theta + 1}{2\Theta^2 - \Theta - 1} \textsc{Opt}(\sigma).
\end{aligned}$$ □

Consequently, we obtain a general upper bound for the competitive ratio of SMARTERSTART for online TSP on the line that is slightly stronger than our bound for the competitive ratio of SMARTERSTART for online DIAL-A-RIDE on the line.

Theorem 6.23. *The function* $\max\{g_{1,\mathrm{TSP}}^{\mathrm{up}}, g_2^{\mathrm{up}}\}$ *gives an upper bound for the competitive ratio of* SMARTERSTART *for open online* TSP *on the line for all* $\Theta > 1$. *Let* $\Theta_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}} \approx 1.6789$ *be the unique solution of* $g_{1,\mathrm{TSP}}^{\mathrm{up}}(\Theta) = g_2^{\mathrm{up}}(\Theta)$, *i.e., of*

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

in the interval $(1,\infty)$. *Then,* $\Theta_{\text{S+}}^{\text{T},\mathbb{R}}$ *is the unique minimum of th function* $\max\{g_{1,\text{TSP}}^{\text{up}}, g_2^{\text{up}}\}$. SMARTERSTART *with scaling parameter* $\Theta_{\text{S+}}^{\text{T},\mathbb{R}}$ *is* $\rho_{\text{S+}}^{\text{T},\mathbb{R}}$*-competitive with*

$$\rho_{\text{S+}}^{\text{T},\mathbb{R}} := g_{1,\text{TSP}}^{\text{up}}(\Theta_{\text{S+}}^{\text{T},\mathbb{R}}) = g_2^{\text{up}}(\Theta_{\text{S+}}^{\text{T},\mathbb{R}}) \approx 2.6288.$$

Proof. For the case where SMARTERSTART postpones the final schedule we have established the upper bound

$$\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{3\Theta^2 - \Theta + 1}{2\Theta^2 - \Theta - 1} = g_{1,\text{TSP}}^{\text{up}}(\Theta)$$

in Proposition 6.22, and for the case where SMARTERSTART starts the final schedule immediately after the second to final schedule we have established the upper bound

$$\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{3\Theta^2 + 3}{2\Theta + 1} = g_2^{\text{up}}(\Theta)$$

in Proposition 6.4. Thus, the maximum of both bounds is a general upper bound for the competitive ratio of SMARTERSTART for open online DIAL-A-RIDE on the line that is independent of SMARTERSTART's behavior.

Function $g_{1,\text{TSP}}^{\text{up}}$ is strictly decreasing for $\Theta > 1$ and function g_2^{up} is strictly increasing for $\Theta > 1$. Therefore, the minimum of $\max\{g_{1,\text{TSP}}^{\text{up}}, g_2^{\text{up}}\}$ in the interval $(1,\infty)$ lies in the intersection point of $g_{1,\text{TSP}}^{\text{up}}$ and g_2^{up}, i.e., in $\Theta_{\text{S+}}^{\text{T},\mathbb{R}} \approx 1.6789$. The resulting upper bound for the competitive ratio is

$$\rho_{\text{S+}}^{\text{T},\mathbb{R}} = g_{1,\text{TSP}}^{\text{up}}(\Theta_{\text{S+}}^{\text{T},\mathbb{R}}) = g_2^{\text{up}}(\rho_{\text{S+}}^{\text{T},\mathbb{R}}) \approx 2.6288. \qquad \square$$

See Figure 6.9 for a visualization of the upper bound for the competitive ratio of online TSP on the line presented in Theorem 6.23 and the lower bound provided by Lemma 6.14.

In the following, we will present two request sequences $\sigma_4^{\text{S+}}$ and $\sigma_5^{\text{S+}}$. We complement the upper bound of Proposition 6.22 with a matching lower bound by computing the ratio of SMARTERSTART's and OPT's completion time of $\sigma_4^{\text{S+}}$ for $\Theta \in (1, 1+\sqrt{3}]$. The request sequence $\sigma_5^{\text{S+}}$ provides an additional lower bound for the competitive ratio of SMARTERSTART for open online TSP on the line. However, first we take another look at the request sequence $\sigma_2^{\text{S+}}$: Since then competitive ratio $\rho_{\text{S+}}^{\text{D},\mathbb{R}}$ of the DIAL-A-RIDE version of SMARTERSTART is slightly larger than the upper bound $\rho_{\text{S+}}^{\text{T},\mathbb{R}}$ for the competitive ratio of the TSP version, the lower bound $g_2^{\text{low}}(\Theta)$ provided by the request sequence $\sigma_2^{\text{S+}}$ holds for slightly larger scaling parameters Θ then in the DIAL-A-RIDE version.

Lemma 6.24. *We have*

$$\frac{\textsc{SmarterStart}(\sigma_2^{\mathrm{S+}})}{\textsc{Opt}(\sigma_2^{\mathrm{S+}})} > \rho_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}} \approx 2.6288$$

for $\Theta \in (2,3]$.

Proof. According to Lemma 6.14, we have

$$\frac{\textsc{SmarterStart}(\sigma_2^{\mathrm{S+}})}{\textsc{Opt}(\sigma_2^{\mathrm{S+}})} = \frac{5\Theta^2 - 3\Theta + 1}{2\Theta^2 - \Theta - 1} - \varepsilon = g_2^{\mathrm{low}}(\Theta) - \varepsilon$$

for $\Theta \in (2,3]$ and sufficiently small $\varepsilon > 0$. Let $\varepsilon < \frac{1}{50}$. The function g_2^{low} is monotonically decreasing on $(2,3]$. Therefore, we have

$$\frac{\textsc{SmarterStart}(\sigma_2^{\mathrm{S+}})}{\textsc{Opt}(\sigma_2^{\mathrm{S+}})} = g_2^{\mathrm{low}}(3) - \varepsilon > 2.642 - \varepsilon > \rho_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}}$$

for all $\Theta \in (2,3]$ and $\varepsilon < \frac{1}{50}$. □

We define the request sequence $\sigma_4^{\mathrm{S+}}$.

Definition 6.25. Let $\varepsilon' > 0$ with $\varepsilon' < \frac{\Theta}{\Theta-1}$. We define

$$\sigma_4^{\mathrm{S+}} := (s_1, s_2^{(1)}, s_2^{(2)})),$$

with

$$\begin{aligned} s_1 &= (1,1;0) \\ s_2^{(1)} &= \left(2 + \frac{1}{\Theta - 1} - \varepsilon'; \frac{1}{\Theta - 1} + \varepsilon'\right), \\ s_2^{(2)} &= \left(-\frac{1}{\Theta - 1}; \frac{1}{\Theta - 1} + \varepsilon'\right). \end{aligned}$$

Note that $\varepsilon' < \frac{\Theta}{\Theta-1}$ implies $a_2^{(1)} > 1$, i.e., the request $s_2^{(1)}$ appears to the right of the request s_1. We begin our analysis with the computation of $\textsc{Opt}(\sigma_4^{\mathrm{S+}})$.

Lemma 6.26. *We have*

$$\textsc{Opt}(\sigma_4^{\mathrm{S+}}) = \frac{2\Theta + 1}{\Theta - 1}.$$

Proof. Opt waits at the origin until time ε' and then performs the walk

$$0 \to -\frac{1}{\Theta-1} \to 2 + \frac{1}{\Theta-1} - \varepsilon'.$$

An illustration of Opt's walk is presented in blue in Figure 6.10 for $\Theta = \Theta_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}}$. We show that all requests can be served this way: Opt serves $s_2^{(2)}$ at time $r_2^{(2)} = \frac{1}{\Theta-1} + \varepsilon'$ and then walks straight towards position $2 + \frac{1}{\Theta-1} - \varepsilon'$. At the time $\frac{2}{\Theta-1} + \varepsilon'$, when Opt is at the origin again, all remaining requests are already released. Therefore all remaining requests can be served at arrival and we have

$$\textsc{Opt}(\sigma_4^{\mathrm{S+}}) = \varepsilon' + D\left(0 \to -\frac{1}{\Theta-1} \to 2 + \frac{1}{\Theta-1} - \varepsilon'\right) = \frac{2\Theta+1}{\Theta-1}. \qquad \square$$

Next, we compute SmarterStart's completion time.

Lemma 6.27. *Let* $\Theta \leq 1+\sqrt{3}$ *and* $\varepsilon' < 1$. *Then, we have*

$$\textsc{SmarterStart}(\sigma_4^{\mathrm{S+}}) = \frac{3\Theta^2 - \Theta + 1}{(\Theta-1)^2} - \frac{2\Theta-1}{\Theta-1}\varepsilon'.$$

Proof. SmarterStart's walk is presented in green in Figure 6.10 for $\Theta = \Theta_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}}$. For all $t \geq 0$, we have $L(t, 0, (s_1)) = 1$. Thus, SmarterStart starts its first schedule S_1 at time $t_1 = \frac{1}{\Theta-1}$ and reaches position $p_2 = 1$ at time $v_2 = \frac{\Theta}{\Theta-1}$. For $t \geq v_2$ we have

$$L(t, 0, (s_1, s_2^{(1)}, s_2^{(2)})) = D\left(0 \to -\frac{1}{\Theta-1} \to 2 + \frac{1}{\Theta-1} - \varepsilon'\right) = \frac{2\Theta+1}{\Theta-1} - \varepsilon'.$$

Thus, the second schedule S_2 is not started before time

$$w_2 = \frac{L\left(t, 0, (s_1, s_2^{(1)}, s_2^{(2)})\right)}{\Theta-1} = \frac{2\Theta+1}{(\Theta-1)^2} - \frac{\varepsilon'}{\Theta-1}.$$

By assumption, we have $\Theta \leq 1+\sqrt{3}$ and $\varepsilon' < 1$, which implies that for the time $v_2 = \frac{\Theta}{\Theta-1}$, when SmarterStart reaches position $p_2 = 1$, the inequality

$$w_2 = \frac{2\Theta+1}{(\Theta-1)^2} - \frac{\varepsilon'}{\Theta-1} \overset{\varepsilon'<1}{>} \frac{\Theta+2}{(\Theta-1)^2} \overset{\Theta\leq 1+\sqrt{3}}{\geq} \frac{\Theta}{\Theta-1} = v_2 \tag{6.24}$$

holds. Note that inequality (6.24) also holds for slightly larger Θ if we let $\varepsilon \to 0$. Because of inequality (6.24), SmarterStart has a waiting period and starts the schedule S_2 at time

$$t_2 = \max\{v_2, w_2\} \overset{(6.24)}{=} w_2 = \frac{2\Theta+1}{(\Theta-1)^2} - \frac{\varepsilon'}{\Theta-1}.$$

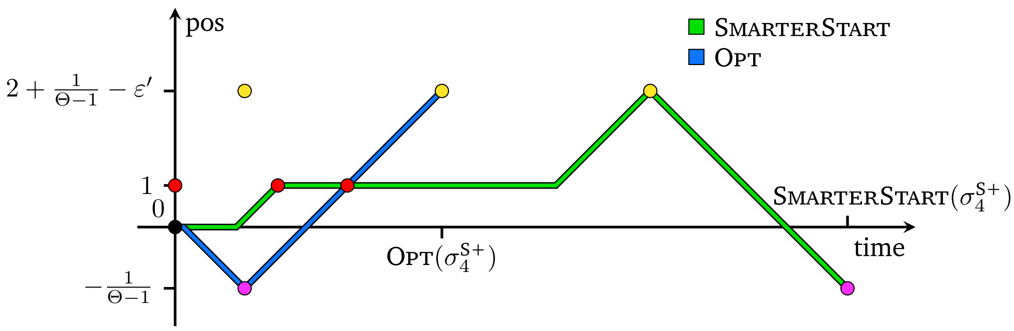

Figure 6.10: SmarterStart's and Opt's walk serving σ_4^{S+} with $\varepsilon' = 0.2$ and $\Theta = \Theta_{S+}^{T,\mathbb{R}}$. Request s_1 is red ●, request $s_2^{(1)}$ is yellow ○ and request $s_2^{(2)}$ is violet ○.

If SmarterStart serves $s_2^{(2)}$ before serving $s_2^{(1)}$ the time it needs is at least

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

The best schedule that serves $s_2^{(2)}$ after serving $s_2^{(1)}$ needs time

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

Thus, SmarterStart serves $s_2^{(2)}$ after serving $s_2^{(1)}$ and we have

$$L(t_2, p_2, (s_2^{(1)}, s_2^{(2)})) = \frac{3\Theta}{\Theta - 1} - 2\varepsilon'.$$

To sum it up, we have

$$\textsc{SmarterStart}(\sigma_4^{S+}) = t_2 + L(t_2, p_2, (s_2^{(1)}, s_2^{(2)})) = \frac{3\Theta^2 - \Theta + 1}{(\Theta - 1)^2} - \frac{2\Theta - 1}{\Theta - 1}\varepsilon'. \qquad \square$$

Equipped with the Lemmas 6.26 and 6.27, we compute a lower bound for the competitive ratio of SmarterStart for open online TSP on the line for $1 < \Theta \leq 1 + \sqrt{3}$.

Proposition 6.28. *Let $1 < \Theta \leq 1 + \sqrt{3}$. For every sufficiently small $\varepsilon > 0$ there is a request sequence σ_4^{S+} such that SmarterStart postpones the final schedule S_N and such that*

$$\frac{\textsc{SmarterStart}(\sigma_4^{S+})}{\textsc{Opt}(\sigma_4^{S+})} = \frac{3\Theta^2 - \Theta + 1}{2\Theta^2 - \Theta - 1} - \varepsilon =: g_{1,\text{TSP}}^{\text{up}}(\Theta) - \varepsilon,$$

i.e., the upper bound established in Proposition 6.4 is tight for $\Theta \in (1, 1 + \sqrt{3}] \approx (1, 2.7321]$.

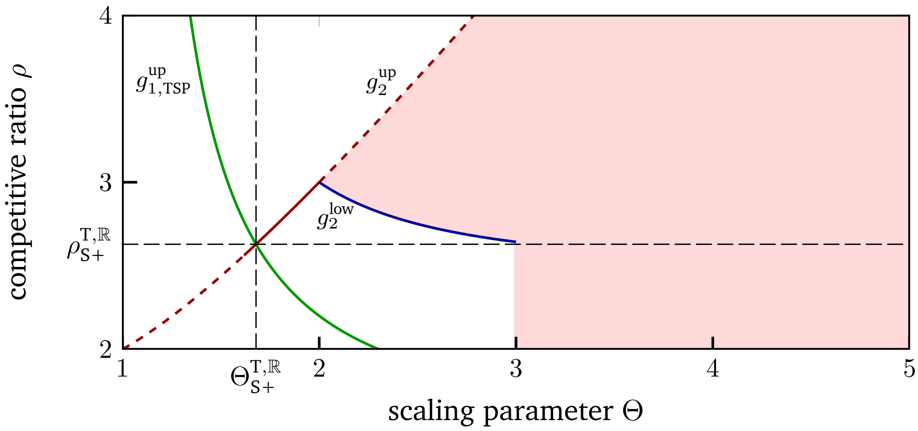

Figure 6.11: Functions $g^{\mathrm{up}}_{1,\mathrm{TSP}}$ (green) / g^{up}_{2} (red): upper bounds for competitive ratio for postponing / non-postponing case, drawn solid if tight. Function: g^{low}_{2} (blue): lower bound for competitive ratio. Red area: possible values for the competitive ratio, bounded by g^{up}_{2} and g^{low}_{2}.

Proof. Let $\varepsilon > 0$ with $\varepsilon < \frac{2\Theta-1}{2\Theta+1}$ and $\varepsilon' = \frac{2\Theta+1}{2\Theta-1}\varepsilon < 1$. By Lemma 6.27, we have

$$\textsc{SmarterStart}(\sigma_4^{\mathrm{S+}}) = \frac{3\Theta^2-\Theta+1}{(\Theta-1)^2} - \frac{2\Theta-1}{\Theta-1}\varepsilon'.$$

By Lemma 6.26, we have

$$\textsc{Opt}(\sigma_4^{\mathrm{S+}}) = \frac{2\Theta+1}{\Theta-1}.$$

Since we have $\varepsilon' = \frac{2\Theta+1}{2\Theta-1}\varepsilon$, we finally obtain

$$\frac{\textsc{SmarterStart}(\sigma_4^{\mathrm{S+}})}{\textsc{Opt}(\sigma_4^{\mathrm{S+}})} = \frac{3\Theta^2-\Theta+1}{2\Theta^2-\Theta-1} - \frac{2\Theta-1}{2\Theta+1}\varepsilon' = \frac{3\Theta^2-\Theta+1}{2\Theta^2-\Theta-1} - \varepsilon = g^{\mathrm{up}}_{1,\mathrm{TSP}}(\Theta) - \varepsilon. \quad \square$$

Figure 6.13 is a visualization of the upper bound for the competitive ratio of online TSP on the line presented in Theorem 6.23 together with the lower bounds of Propositions 6.28 and 6.13 as well as Lemma 6.24.

Recall that the optimal parameter $\Theta^{\mathrm{T},\mathbb{R}}_{\mathrm{S+}}$ established in Theorem 6.23 is the only positive, real solution of the equation

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

which is $\Theta_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}} \approx 1.6789$. Therefore, by Proposition 6.28 and Proposition 6.13 the parameter $\Theta_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}}$ lies in the interval where the upper bounds of Propositions 6.22 and 6.4 are both tight. It remains to make sure that for all Θ that lie outside of this interval the competitive ratio of SMARTERSTART is larger than $\rho_{\mathrm{S+}}^{\mathrm{D},\mathbb{R}} \approx 2.6288$. For this, we examine the request sequence $\sigma_5^{\mathrm{S+}}$.

Definition 6.29. Let $\Theta > 3$ and $\varepsilon' > 0$ with $\varepsilon' < \frac{\Theta-2}{\Theta-1}$. We define

$$\sigma_5^{\mathrm{S+}} := (s_1, s_2^{(1)}, s_2^{(1)}, s_3)$$

with

$$\begin{aligned}
s_1 &= (1;0),\\
s_2^{(1)} &= \left(\frac{2\Theta-3}{\Theta-1} - \varepsilon'; \frac{1}{\Theta-1} + \varepsilon'\right),\\
s_2^{(2)} &= \left(\frac{1}{\Theta-1}; \frac{1}{\Theta-1} + \varepsilon'\right),\\
s_3 &= \left(\frac{2\Theta-3}{\Theta-1} - \varepsilon'; \frac{\Theta}{\Theta-1} + \varepsilon'\right).
\end{aligned}$$

Note that $\varepsilon' < \frac{\Theta-2}{\Theta-1}$ implies $\frac{2\Theta-3}{\Theta-1} - \varepsilon' > 1$, i.e., the position of requests $s_2^{(1)}$ and s_3 to the right of the position of request s_1. We begin our analysis with the computation of $\mathrm{OPT}(\sigma_5^{\mathrm{S+}})$.

Lemma 6.30. *Let $\varepsilon' < \frac{\Theta-3}{\Theta-1}$. We have*

$$\mathrm{OPT}(\sigma_5^{\mathrm{S+}}) = \frac{2\Theta-3}{\Theta-1}.$$

Proof. An illustration of OPT's walk is presented in blue in Figure 6.12 for $\Theta = 3.5$. OPT waits at the origin until time ε' and then performs the walk

$$0 \to \frac{2\Theta-3}{\Theta-1} - \varepsilon'.$$

We show that all requests are served this way: This is clear for the requests s_1 and $s_2^{(2)}$. The position of the remaining two requests $s_2^{(1)}$ and s_3 is reached at time $\frac{2\Theta-3}{\Theta-1}$. Since we have

$$\frac{2\Theta-3}{\Theta-1} \overset{\varepsilon' < \frac{\Theta-3}{\Theta-1}}{>} \frac{\Theta}{\Theta-1} + \varepsilon' = r_3 > r_2^{(1)},$$

both requests can be served at arrival and we have

$$\textsc{Opt}(\sigma_5^{\mathrm{S}+}) = \varepsilon' + D\left(0 \to \frac{2\Theta - 3}{\Theta - 1} - \varepsilon'\right) = \frac{2\Theta - 3}{\Theta - 1}. \qquad \square$$

Next, we compute SmarterStart's completion time.

Lemma 6.31. *We have*

$$\textsc{SmarterStart}(\sigma_5^{\mathrm{S}+}) = \frac{6\Theta - 10}{\Theta - 1} - 3\varepsilon'.$$

Proof. SmarterStart's walk is presented in green in Figure 6.12 for $\Theta = 3.5$. For all $t \geq 0$, we have $L(t, 0, (s_1)) = 1$. Thus, SmarterStart starts its first schedule S_1 at time $t_1 = \frac{1}{\Theta - 1}$ and reaches position $p_2 = 1$ at time $v_2 = \frac{\Theta}{\Theta - 1}$. For $t \geq v_2$ we have

$$L(t, 0, (s_1, s_2^{(1)}, s_2^{(2)})) = D\left(0 \to \frac{2\Theta - 3}{\Theta - 1} - \varepsilon'\right) = \frac{2\Theta - 3}{\Theta - 1} - \varepsilon'.$$

Thus, the second schedule S_2 is not started before time

$$w_2 = \frac{L\Big(t, 0, (s_1, s_2^{(1)}, s_2^{(2)})\Big)}{\Theta - 1} = \frac{2\Theta - 3}{(\Theta - 1)^2} - \frac{\varepsilon'}{\Theta - 1}.$$

For the time $v_2 = \frac{\Theta}{\Theta - 1}$, when SmarterStart reaches position $p_2 = 1$, we have

$$w_2 = \frac{2\Theta - 3}{(\Theta - 1)^2} - \frac{\varepsilon'}{\Theta - 1} \overset{\Theta > 1}{<} \frac{\Theta}{\Theta - 1} = v_2. \tag{6.25}$$

Because of inequality (6.25), SmarterStart does not postpone the schedule S_2 at time

$$t_2 = \max\{v_2, w_2\} \overset{(6.25)}{=} v_2 = \frac{\Theta}{\Theta - 1}.$$

The shortest schedule serving $s_2^{(2)}$ before serving $s_2^{(1)}$ has length

$$D\left(1 \to \frac{1}{\Theta - 1} \to \frac{2\Theta - 3}{\Theta - 1} - \varepsilon'\right) = \frac{3\Theta - 6}{\Theta - 1} - \varepsilon'.$$

The shortest schedule that serves $s_2^{(2)}$ after serving $s_2^{(1)}$ needs time

$$D\left(1 \to \frac{2\Theta - 3}{\Theta - 1} - \varepsilon' \to \frac{1}{\Theta - 1}\right) = \frac{3\Theta - 6}{\Theta - 1} - 2\varepsilon'.$$

Thus, SmarterStart serves $s_2^{(2)}$ after serving $s_2^{(1)}$ and we have

$$L(t_2, p_2, (s_2^{(1)}, s_2^{(2)})) = \frac{3\Theta - 6}{\Theta - 1} - 2\varepsilon'.$$

Schedule S_2 ends at time

$$v_3 = t_2 + L(t_2, p_2, (s_2^{(1)}, s_2^{(2)})) = \frac{4\Theta - 6}{\Theta - 1} - 2\varepsilon'$$

at position $p_3 = \frac{1}{\Theta - 1}$. For $t \geq v_3$ we have

$$L(t, 0, (s_1, s_2^{(1)}, s_2^{(2)}, s_3)) = D\left(0 \to \frac{2\Theta - 3}{\Theta - 1} - \varepsilon'\right) = \frac{2\Theta - 3}{\Theta - 1} - \varepsilon'.$$

Thus, we have $w_3 = w_2$, which is smaller than v_2 by inequality (6.25), which again is smaller than v_3. Therefore, the final schedule S_3 is started at time

$$t_3 = v_3 = \frac{4\Theta - 6}{\Theta - 1} - 2\varepsilon'.$$

For all $t \geq v_3$, we have

$$L(t, p_3, (s_3)) = D\left(\frac{1}{\Theta - 1} \to \frac{2\Theta - 3}{\Theta - 1} - \varepsilon'\right) = \frac{2\Theta - 4}{\Theta - 1} - \varepsilon'.$$

To sum it up, we have

$$\textsc{SmarterStart}(\sigma_5^{\mathsf{S+}}) = t_3 + L(t_3, p_3, (s_3)) = \frac{6\Theta - 10}{\Theta - 1} - 3\varepsilon'. \qquad \square$$

Equipped with the Lemmas 6.30 and 6.31, we compute a lower bound for the competitive ratio of SmarterStart for open online TSP on the line for $\Theta > 3$.

Lemma 6.32. *Let $\Theta > 3$ and $\varepsilon > 0$ sufficiently small. Then, we have*

$$\frac{\textsc{SmarterStart}(\sigma_5^{\mathsf{S+}})}{\textsc{Opt}(\sigma_5^{\mathsf{S+}})} = \frac{6\Theta - 10}{2\Theta - 3} - \varepsilon =: g_5^{\text{low}}(\Theta) - \varepsilon.$$

In particular, we have

$$\frac{\textsc{SmarterStart}(\sigma_5^{\mathsf{S+}})}{\textsc{Opt}(\sigma_5^{\mathsf{S+}})} > \rho_{\mathsf{S+}}^{\mathrm{T},\mathbb{R}} \approx 2.6288.$$

for $\Theta \in (3, \infty)$ and sufficiently small ε.

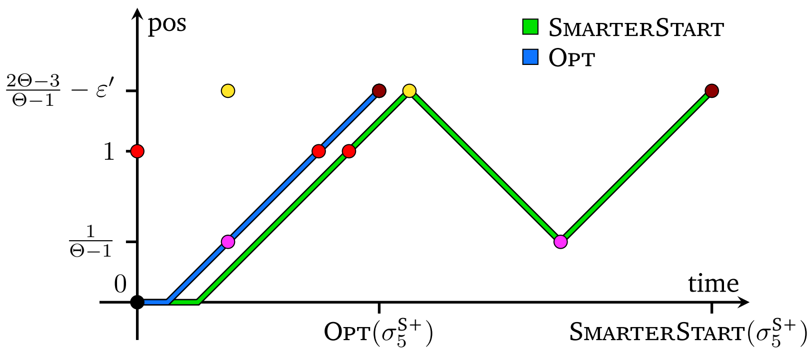

Figure 6.12: SmarterStart's and Opt's walk serving σ_5^{S+} with $\varepsilon' = 0.2$ and $\Theta = 3.5$. Request s_1 is red ●, request $s_2^{(1)}$ is yellow ●, request $s_2^{(2)}$ is violet ● and request s_3 is brown ●.

Proof. Let $\varepsilon > 0$ with $\varepsilon < \min\{\frac{3\Theta-3}{2\Theta-3}(\frac{\Theta-3}{\Theta-1}), \frac{1}{50}\}$ and $\varepsilon' = \frac{2\Theta-3}{3\Theta-3}\varepsilon < \frac{\Theta-3}{\Theta-1}$. By Lemma 6.31, we have

$$\mathrm{SmarterStart}(\sigma_5^{S+}) = \frac{6\Theta-10}{\Theta-1} - 3\varepsilon'.$$

By Lemma 6.30, we have

$$\mathrm{Opt}(\sigma_5^{S+}) = \frac{2\Theta-3}{\Theta-1}.$$

Since we have $\varepsilon' = \frac{2\Theta-3}{3\Theta-3}$, we obtain

$$\frac{\mathrm{SmarterStart}(\sigma_5^{S+})}{\mathrm{Opt}(\sigma_5^{S+})} = \frac{6\Theta-10}{2\Theta-3} - \frac{3\Theta-3}{2\Theta-3}\varepsilon' = \frac{6\Theta-10}{2\Theta-3} - \varepsilon = g_5^{\mathrm{low}}(\Theta) - \varepsilon.$$

The function g_5^{low} is monotonically increasing on the interval $[3, \infty)$. Therefore, we have

$$\frac{\mathrm{SmarterStart}(\sigma_5^{S+})}{\mathrm{Opt}(\sigma_5^{S+})} > g_5^{\mathrm{low}}(3) - \varepsilon = \frac{8}{3} - \varepsilon > \rho_{S+}^{T,\mathbb{R}} \approx 2.6288$$

for all $\Theta > 3$ and $\varepsilon < \frac{1}{50}$. □

We combine all lower bounds constructed in this section into one general lower bound. See Figure 6.13 for an illustration of all upper and lower bounds for the competitive ratio og SmarterStart for open online TSP on the line.

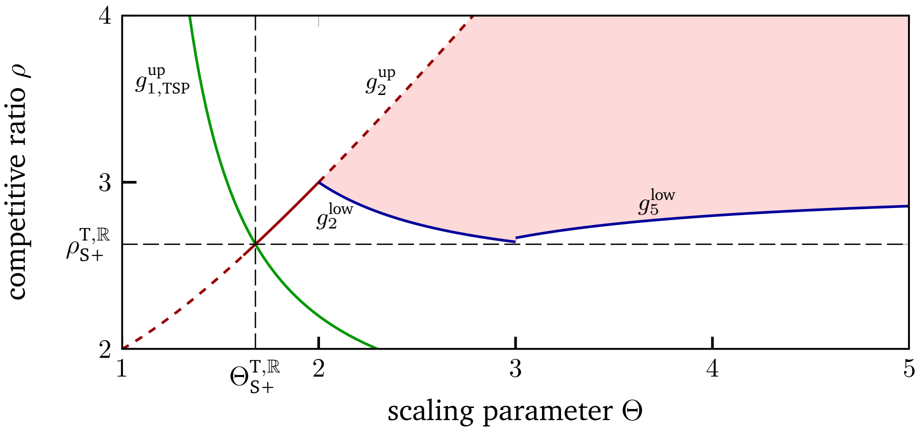

Figure 6.13: Functions $g_{1,\mathrm{TSP}}^{\mathrm{up}}$ (green) / g_2^{up} (red): upper bounds for competitive ratio for postponing / non-postponing case, drawn solid if tight. Functions: g_2^{low} and g_5^{low} (blue): lower bounds for competitive ratio. Red area: possible values for the competitive ratio, bounded by g_2^{up} and g_2^{low} / g_5^{low}.

Theorem 6.33. *Let $G_{\mathrm{TSP}} : \mathbb{R}_{>1} \to \mathbb{R}_{>1}$ be a function with*

$$G_{\mathrm{TSP}}(\Theta) := \begin{cases} g_{1,\mathrm{TSP}}^{\mathrm{up}}(\Theta), & \textit{for } \Theta \in (1, \Theta_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}}], \\ g_2^{\mathrm{up}}(\Theta), & \textit{for } \Theta \in (\Theta_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}}, 2], \\ g_2^{\mathrm{low}}(\Theta), & \textit{for } \Theta \in (2, 3], \\ g_5^{\mathrm{low}}(\Theta), & \textit{for } \Theta \in (3, \infty). \end{cases}$$

Then G_{TSP} is a general lower bound for the competitive ratio of SmarterStart *for open online* TSP *on the line. The unique minimum of G_{TSP} lies in $\Theta = \Theta_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}}$ and yields a lower bound of*

$$G_{\mathrm{TSP}}(\Theta_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}}) = \rho_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}} \approx 2.6288.$$

Proof. We have shown in Proposition 6.28 that $g_{1,\mathrm{TSP}}^{\mathrm{up}}(\Theta)$ with $\Theta \in (1, 2)$ is a lower bound for the competitive ratio of SmarterStart for open online TSP on the line and in Proposition 6.13 that $g_2^{\mathrm{up}}(\Theta)$ with $\Theta \in [\frac{1}{2}(1+\sqrt{5}), 2]$ is a lower bound. Theorem 6.23 implies that G_{TSP} has its unique minimum in the interval $(1, 2]$ at $\Theta = \Theta_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}}$. It remains to show that $G_{\mathrm{TSP}}(\Theta) > G_{\mathrm{TSP}}(\Theta_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}})$ for all $\Theta > 2$. This immediately follows from the Lemmas 6.24 and 6.32. □

The main theorem of this section follows by combining Theorem 6.23 and Theorem 6.33.

Theorem 6.34. *The competitive ratio of* SMARTERSTART *for open online* TSP *on the line with scaling parameter* $\Theta_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}} \approx 1.6789$ *is exactly*

$$\rho_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}} = g_{1,\mathrm{TSP}}^{\mathrm{up}}(\Theta_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}}) = g_2^{\mathrm{up}}(\Theta_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}}) \approx 2.6288.$$

For every other $\Theta > 1$ *with* $\Theta \neq \Theta_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}}$ *the competitive ratio of* SMARTERSTART *is strictly larger than* $\rho_{\mathrm{S+}}^{\mathrm{T},\mathbb{R}}$.

6.4 Upper Bounds in the General Setting

It remains to examine SMARTERSTART for general continuous metric spaces. Since the real line is a special case of a continuous metric space, every lower bound established in the previous sections also holds for general continuous metric spaces. This is not necessarily true for the upper bounds we have presented. However, the upper bound g_1^{up} for open online DIAL-A-RIDE presented in Proposition 6.3 and the upper bound $g_{1,\mathrm{TSP}}^{\mathrm{up}}$ for open online TSP presented in Proposition 6.22 are also valid for general continuous metric spaces. The upper bound for the non-postponing case presented in Proposition 6.4 relies on Proposition 4.10 which uses line-specific features. Therefore, we need to compute a new upper bound for the non-postponing case.

Proposition 6.35. *If* SMARTERSTART *for open online* DIAL-A-RIDE *does not postpone executing* S_N, *we have*

$$\frac{\textsc{SmarterStart}(\sigma)}{\textsc{Opt}(\sigma)} \leq \Theta + 1 =: g_{2,X}^{\mathrm{up}}(\Theta).$$

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

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

$$\textsc{Opt}(\sigma) \geq r_n. \tag{6.26}$$

In total we have

$$\begin{aligned} \textsc{SmarterStart}(\sigma) &\overset{(4.1)}{=} t_N + L(t_N, p_N, \sigma_N) \\ &\overset{(6.4)}{=} r_n + L(t_N, p_N, \sigma_N) \\ &\overset{(6.26)}{\leq} \textsc{Opt}(\sigma) + L(t_N, p_N, \sigma_N) \end{aligned}$$

$$\stackrel{\text{Lem 6.2}}{\leq} \left(2 + \frac{\Theta - 1}{\Theta + 1}\right)\textsc{Opt}(\sigma)$$
$$\stackrel{\Theta \geq 1}{<} (\Theta + 1)\textsc{Opt}(\sigma).$$

Now, consider the case that the final schedule is started immediately after the second to final schedule. Let $s_N^{\textsc{Opt}}$ be the first request of σ_N that is served by Opt and let $a_N^{\textsc{Opt}}$ be its starting position and $r_N^{\textsc{Opt}}$ be its release time. We have

$$\begin{aligned} \textsc{SmarterStart}(\sigma) &\stackrel{(4.1)}{=} t_N + L(t_N, p_N, \sigma_N) \\ &\stackrel{(6.3)}{=} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + L(t_N, p_N, \sigma_N) \\ &\stackrel{t_N \geq r_N^{\textsc{Opt}}}{\leq} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + L(r_N^{\textsc{Opt}}, p_N, \sigma_N). \end{aligned} \tag{6.27}$$

Since Opt serves all requests of σ_N after time $r_N^{\textsc{Opt}}$, starting with a request with starting position $a_N^{\textsc{Opt}}$, we have

$$\textsc{Opt}(\sigma) \geq r_N^{\textsc{Opt}} + L(r_N^{\textsc{Opt}}, a_N^{\textsc{Opt}}, \sigma_N). \tag{6.28}$$

Furthermore, we have

$$r_N^{\textsc{Opt}} > t_{N-1} \tag{6.29}$$

since otherwise $s_N^{\textsc{Opt}} \in \sigma_{N-1}$ would hold. This gives us

$$\begin{aligned} \textsc{SmarterStart}(\sigma) &\stackrel{(6.27)}{\leq} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + L(r_N^{\textsc{Opt}}, p_N, \sigma_N) \\ &\stackrel{(4.3)}{\leq} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + |a_N^{\textsc{Opt}} - p_N| \\ &\quad + L(r_N^{\textsc{Opt}}, a_N^{\textsc{Opt}}, \sigma_N) \\ &\stackrel{(6.28)}{\leq} t_{N-1} + L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + |a_N^{\textsc{Opt}} - p_N| + \textsc{Opt}(\sigma) - r_N^{\textsc{Opt}} \\ &\stackrel{(6.29)}{<} L(t_{N-1}, p_{N-1}, \sigma_{N-1}) + |a_N^{\textsc{Opt}} - p_N| + \textsc{Opt}(\sigma) \\ &\stackrel{(4.3)}{\leq} |p_{N-1}| + L(t_{N-1}, 0, \sigma_{N-1}) + |a_N^{\textsc{Opt}} - p_N| + \textsc{Opt}(\sigma) \\ &\stackrel{\text{Lem 6.1}}{\leq} (\Theta - 1)t_{N-2} + L(t_{N-1}, 0, \sigma_{N-1}) + |a_N^{\textsc{Opt}} - p_N| + \textsc{Opt}(\sigma). \end{aligned} \tag{6.30}$$

We have

$$\textsc{Opt}(\sigma) \geq t_{N-2} + |a_N^{\textsc{Opt}} - p_N|, \tag{6.31}$$

because $\textsc{Opt}$ has to visit both $a_N^{\textsc{Opt}}$ and p_N after time t_{N-2}: It has to visit $a_N^{\textsc{Opt}}$ to collect $s_{S_N}^{\textsc{Opt}}$ and it has to visit p_N to deliver some request of σ_{N-1}. In the case $\Theta \geq 2$, we have

$$\begin{aligned}
\textsc{SmarterStart}(\sigma) &\overset{(6.30)}{<} (\Theta-1)t_{N-2} + L(t_{N-1}, 0, \sigma_{N-1}) + |a_N^{\textsc{Opt}} - p_N| + \textsc{Opt}(\sigma) \\
&\overset{(6.31)}{\leq} 2\textsc{Opt}(\sigma) + L(t_{N-1}, 0, \sigma_{N-1}) + (\Theta-2)t_{N-2} \\
&\overset{(6.31),(4.4)}{\leq} (\Theta+1)\textsc{Opt}(\sigma).
\end{aligned}$$

Thus, we may assume $\Theta < 2$. Similarly as in inequality (6.13), we get

$$\begin{aligned}
\textsc{SmarterStart}(\sigma) &\overset{(6.30)}{<} (\Theta-1)t_{N-2} + L(t_{N-1}, 0, \sigma_{N-1}) + |a_N^{\textsc{Opt}} - p_N| + \textsc{Opt}(\sigma) \\
&\overset{(6.31)}{\leq} \Theta\textsc{Opt}(\sigma) + L(t_{N-1}, 0, \sigma_{N-1}) + (2-\Theta)|a_N^{\textsc{Opt}} - p_N| \\
&\overset{(6.1)}{\leq} \Theta\textsc{Opt}(\sigma) + (\Theta-1)t_{N-1} + (2-\Theta)|a_N^{\textsc{Opt}} - p_N| \\
&\leq (2\Theta-1)\textsc{Opt}(\sigma) + (2-\Theta)|a_N^{\textsc{Opt}} - p_N|,
\end{aligned}$$

where the last inequality follows, because of $\textsc{Opt}(\sigma) \geq t_{N-1}$. □

The upper bound $g_{2,X}^{\text{up}}(\Theta)$ is slightly weaker than the upper bound $g_2^{\text{up}}(\Theta)$. We use Proposition 6.35 to compute a general upper bound for the competitive ratio of SMARTERSTART for open online DIAL-A-RIDE on general continuous metric spaces.

Theorem 6.36. *The function* $\max\{g_1^{\text{up}}, g_{2,X}^{\text{up}}\}$ *gives an upper bound for the competitive ratio of* SMARTERSTART *for open online* DIAL-A-RIDE *in the general setting for all* $\Theta > 1$. *Let* $\Theta_{\text{S+}}^{\text{D},X} \approx 1.6956$ *be the unique solution of* $g_1^{\text{up}}(\Theta) = g_{2,X}^{\text{up}}(\Theta)$, *i.e., of*

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

in the interval $(1, \infty)$. *Then,* $\Theta_{\text{S+}}^{\text{D},X}$ *is the unique minimum of the function* $\max\{g_1^{\text{up}}, g_{2,X}^{\text{up}}\}$ *and* SMARTERSTART *with scaling parameter* $\Theta_{\text{S+}}^{\text{D},X}$ *is* $\rho_{\text{S+}}^{\text{D},X}$*-competitive with*

$$\rho_{\text{S+}}^{\text{D},X} := g_1^{\text{up}}(\Theta_{\text{S+}}^{\text{D},X}) = g_{2,X}^{\text{up}}(\Theta_{\text{S+}}^{\text{D},\mathbb{R}}) \approx 2.6956.$$

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

$$\frac{\textsc{SmarterStart}(\sigma)}{\textsc{Opt}(\sigma)} \leq \frac{2\Theta^2 - \Theta + 1}{\Theta^2 - 1} = g_1^{\text{up}}(\Theta)$$

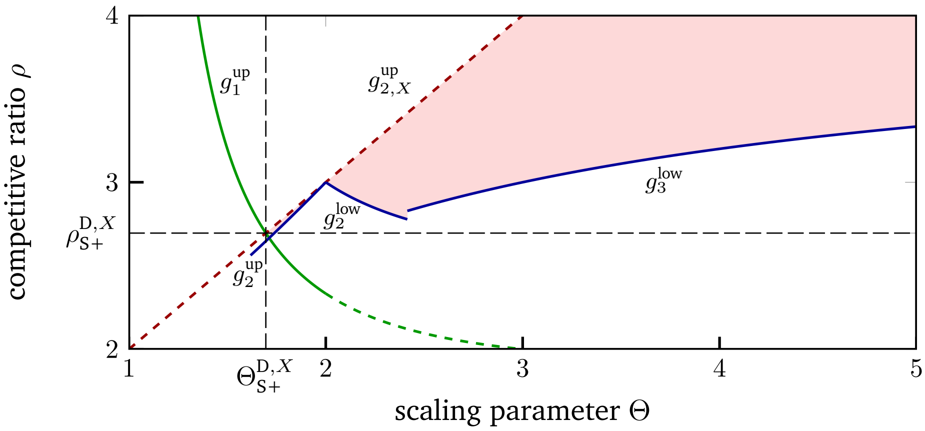

Figure 6.14: Functions g_1^{up} (green) / $g_{2,X}^{\text{up}}$ (red): upper bounds for competitive ratio for postponing / non-postponing case, drawn solid if tight. Functions: g_2^{low} and g_3^{low} (blue): lower bounds for competitive ratio. Red area: possible values for the competitive ratio, bounded by g_2^{up} and $g_{2,X}^{\text{up}}$ as well as g_2^{low} and g_3^{low}.

in Proposition 6.3, and for the case where SmarterStart starts the final schedule immediately after the second to final schedule we have established the upper bound

$$\frac{\textsc{SmarterStart}(\sigma)}{\textsc{Opt}(\sigma)} \leq \Theta + 1 = g_{2,X}^{\text{up}}(\Theta)$$

in Proposition 6.35. Thus, the maximum of both bounds is a general upper bound for the competitive ratio of SmarterStart for open online Dial-a-Ride on the line that is independent of SmarterStart's behavior.

Function g_1^{up} is strictly decreasing for $\Theta > 1$ and function $g_{2,X}^{\text{up}}$ is strictly increasing for $\Theta > 1$. Therefore, the minimum of $\max\{g_1^{\text{up}}, g_{2,X}^{\text{up}}\}$ in the interval $(1, \infty)$ lies in the intersection point of g_1^{up} and $g_{2,X}^{\text{up}}$, i.e., in $\Theta_{\text{S+}}^{\text{D},X} \approx 1.6956$. The resulting upper bound for the competitive ratio is

$$\rho_{\text{S+}}^{\text{D},X} = g_1^{\text{up}}(\Theta_{\text{S+}}^{\text{D},X}) = g_{2,X}^{\text{up}}(\rho_{\text{S+}}^{\text{D},X}) \approx 2.6956. \qquad \square$$

See Figure 6.14 for a visualization of the general upper bound for the competitive ratio of SmarterStart for open online Dial-a-Ride presented in Theorem 6.36 together with the general lower bound presented in Theorem 6.19.

We use Proposition 6.35 to compute a general upper bound for the competitive ratio of SmarterStart for open online TSP on general contiuous metric spaces.

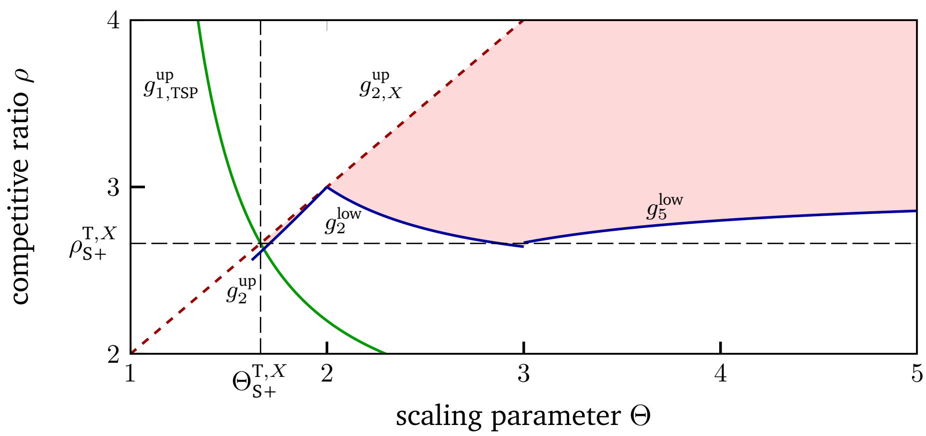

Figure 6.15: Functions $g_{1,\mathrm{TSP}}^{\mathrm{up}}$ (green) / $g_{2,X}^{\mathrm{up}}$ (red): upper bounds for competitive ratio for postponing / non-postponing case, drawn solid if tight. Functions: g_2^{up}, g_2^{low}, g_5^{low} (blue): lower bounds for competitive ratio. Red area: possible values for the competitive ratio, bounded by $g_{2,X}^{\mathrm{up}}$ and g_2^{up} as well as g_2^{low} and g_5^{low}.

Theorem 6.37. *The function* $\max\{g_{1,\mathrm{TSP}}^{\mathrm{up}}, g_{2,X}^{\mathrm{up}}\}$ *gives an upper bound for the competitive ratio of* SMARTERSTART *for open online* TSP *in the general setting for all* $\Theta > 1$*. Let* $\Theta_{S+}^{T,X} \approx 1.6625$ *be the unique solution of* $g_{1,\mathrm{TSP}}^{\mathrm{up}}(\Theta) = g_{2,X}^{\mathrm{up}}(\Theta)$*, i.e., of*

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

in the interval $(1, \infty)$*. Then,* $\Theta_{S+}^{T,X}$ *is the unique minimum of the function* $\max\{g_{1,\mathrm{TSP}}^{\mathrm{up}}, g_{2,X}^{\mathrm{up}}\}$ *and* SMARTERSTART *with scaling parameter* $\Theta_{S+}^{T,X}$ *is* $\rho_{S+}^{T,X}$*-competitive with*

$$\rho_{S+}^{T,X} := g_{1,\mathrm{TSP}}^{\mathrm{up}}(\Theta_{S+}^{T,X}) = g_{2,X}^{\mathrm{up}}(\Theta_{S+}^{T,\mathbb{R}}) \approx 2.6625.$$

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

$$\frac{\textsc{SmarterStart}(\sigma)}{\textsc{Opt}(\sigma)} \leq \frac{3\Theta^2 - \Theta + 1}{2\Theta^2 - \Theta - 1} = g_{1,\mathrm{TSP}}^{\mathrm{up}}(\Theta)$$

in Proposition 6.22, and for the case where SMARTERSTART starts the final schedule immediately after the second to final schedule we have established the upper bound

$$\frac{\textsc{SmarterStart}(\sigma)}{\textsc{Opt}(\sigma)} \leq \Theta + 1 = g_{2,X}^{\mathrm{up}}(\Theta)$$

in Proposition 6.35. Thus, the maximum of both bounds is a general upper bound for the competitive ratio of SMARTERSTART for open online DIAL-A-RIDE on the line that is independent of SMARTERSTART's behavior.

Function $g_{1,\text{TSP}}^{\text{up}}$ is strictly decreasing for $\Theta > 1$ and function $g_{2,X}^{\text{up}}$ is strictly increasing for $\Theta > 1$. Therefore, the minimum of $\max\{g_{1,\text{TSP}}^{\text{up}}, g_{2,X}^{\text{up}}\}$ in the interval $(1, \infty)$ lies in the intersection point of $g_{1,\text{TSP}}^{\text{up}}$ and $g_{2,X}^{\text{up}}$, i.e., in $\Theta_{\text{S+}}^{\text{T},X} \approx 1.6625$. The resulting upper bound for the competitive ratio is

$$\rho_{\text{S+}}^{\text{T},X} = g_{1,\text{TSP}}^{\text{up}}(\Theta_{\text{S+}}^{\text{T},X}) = g_{2,X}^{\text{up}}(\rho_{\text{S+}}^{\text{T},X}) \approx 2.6625. \qquad \square$$

See Figure 6.15 for a visualization of the general upper bound for the competitive ratio of SMARTERSTART for open online TSP presented in Theorem 6.37 together with the general lower bound presented in Theorem 6.33.

6.5 Closed Version of Online DIAL-A-RIDE and TSP

SMARTERSTART for closed online DIAL-A-RIDE behaves very similar to SMARTSTART for closed online DIAL-A-RIDE. The reason for this is that in the closed version every schedule starts in the origin. Thus, the only difference between SMARTERSTART's and SMARTSTART's waiting routine is that SMARTERSTART uses all known requests to compute its waiting time, while SMARTSTART only uses the unserved requests. We will see that this has no impact on the competitive ratio, i.e., the closed version of SMARTERSTART has the same competitive ratio as SMARTSTART of exactly 2 for DIAL-A-RIDE and TSP on the line as well as on general continuous metric spaces.

As in the open version of SMARTERSTART, we distinguish between two cases depending on whether or not the final schedule is postponed. We start with the case that the final schedule is postponed.

Proposition 6.38. *In case* SMARTERSTART *for closed online* DIAL-A-RIDE *or closed online* TSP *postpones executing S_N, we have*

$$\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{\Theta}{\Theta - 1} =: h_1^{\text{up}}(\Theta).$$

Proof. Assume SMARTERSTART postpones its final schedule, then we have

$$\begin{aligned}\text{SMARTERSTART}(\sigma) &\overset{(4.1)}{=} t_N + L(t_N, 0, \sigma_N)\\ &\overset{(6.2)}{=} \frac{L(t_N, 0, \sigma_{\leq t_N})}{\Theta - 1} + L(t_N, 0, \sigma_N)\end{aligned}$$

$$\overset{\text{Lem 4.4}}{\leq} \frac{\Theta}{\Theta - 1}\textsc{Opt}(\sigma). \qquad \square$$

Next, we examine the case that the final schedule is not postponed by the waiting routine and is instead started directly after the second to final schedule is finished.

Proposition 6.39. *If* SMARTERSTART *for closed online* DIAL-A-RIDE *or closed online* TSP *does not postpone the final schedule, we have*

$$\frac{\textsc{SmarterStart}(\sigma)}{\textsc{Opt}(\sigma)} \leq \max\left\{\frac{\Theta + 2}{2}, \Theta\right\} =: h_2^{\text{up}}(\Theta).$$

Proof. Assume algorithm SMARTERSTART does not postpone the final schedule, i.e., SMARTERSTART starts the final schedule S_N either immediately after finishing S_{N-1} or immediately after the last requests are released. Let the latter be the case, then the final schedule is started at the release time r_n of the last request. Since OPT also has to serve the last request, we have $\textsc{Opt}(\sigma) \geq r_n$ and since the execution of the final schedule is not postponed, we have $r_n > \frac{1}{\Theta-1}L(t_N, 0, \sigma)$, i.e.,

$$L(t_N, 0, \sigma_N) \leq L(t_N, 0, \sigma) < (\Theta - 1)\textsc{Opt}(\sigma). \tag{6.32}$$

In total we have

$$\begin{aligned}
\textsc{Smartstart}(\sigma) &\overset{(4.1)}{=} t_N + L(t_N, 0, \sigma_N)\\
&\overset{(6.4)}{=} r_n + L(t_N, 0, \sigma_N)\\
&\overset{(6.32)}{<} \Theta\textsc{Opt}(\sigma)\\
&\leq \max\left\{\frac{\Theta + 2}{2}, \Theta\right\}\textsc{Opt}(\sigma).
\end{aligned}$$

Now let the final schedule be started immediately after the second to final schdedule. We have

$$\begin{aligned}
\textsc{SmarterStart}(\sigma) &\overset{(4.1)}{=} t_N + L(t_N, 0, \sigma_N)\\
&\overset{(6.3)}{=} t_{N-1} + L(t_{N-1}, 0, \sigma_{N-1}) + L(t_N, 0, \sigma_N)\\
&\overset{(4.4)}{\leq} t_{N-1} + L(t_{N-1}, 0, \sigma_{\leq t_N}) + L(t_N, 0, \sigma_N)\\
&\overset{(6.2)}{\leq} \Theta t_{N-1} + L(t_N, 0, \sigma_N).
\end{aligned} \tag{6.33}$$

Let $s_N^{\textsc{Opt}}$ be the first request of σ_N that is served by Opt and let $a_N^{\textsc{Opt}}$ be its starting position and $r_N^{\textsc{Opt}}$ be its release time. We have

$$\textsc{Opt}(\sigma) \geq r_N^{\textsc{Opt}} + L(r_N^{\textsc{Opt}}, a_N^{\textsc{Opt}}, \sigma_N) \geq t_{N-1} + L(t_N, a_N^{\textsc{Opt}}, \sigma_N). \tag{6.34}$$

Since Opt has to return to the origin after serving $s_N^{\textsc{Opt}}$, we have

$$\textsc{Opt}(\sigma) \geq r_N^{\textsc{Opt}} + d(0, a_N^{\textsc{Opt}}) \geq t_{N-1} + d(0, a_N^{\textsc{Opt}}) \tag{6.35}$$

and

$$d(0, a_N^{\textsc{Opt}}) \leq \frac{1}{2}\textsc{Opt}(\sigma). \tag{6.36}$$

To sum it up, we have

$$\begin{aligned}
\textsc{SmarterStart}(\sigma) &\overset{(6.33)}{\leq} \Theta t_{N-1} + L(t_N, 0, \sigma_N) \\
&\overset{(4.3)}{\leq} \Theta t_{N-1} + d(0, a_N^{\textsc{Opt}}) + L(t_N, a_N^{\textsc{Opt}}, \sigma_N) \\
&\overset{(6.35)}{\leq} (\Theta - 1)t_{N-1} + d(0, a_N^{\textsc{Opt}}) + \textsc{Opt}(\sigma) \\
&\overset{(6.35)}{\leq} \max\{0, \Theta - 2\}t_{N-1} + \max\{2 - \Theta, 0\}d(0, a_N^{\textsc{Opt}}) \\
&\qquad + \max\{\Theta, 2\}\textsc{Opt}(\sigma) \\
&\overset{(6.36)}{\leq} \max\{0, \Theta - 2\}t_{N-1} + \max\left\{\frac{\Theta + 2}{2}, 2\right\}\textsc{Opt}(\sigma) \\
&\overset{\textsc{Opt}(\sigma) \geq t_{N-1}}{<} \max\left\{\frac{\Theta + 2}{2}, \Theta\right\}\textsc{Opt}(\sigma).
\end{aligned}$$

□

We summarize the upper bounds for the competitive ratio of SmarterStart for closed online Dial-a-Ride and closed online TSP provided by the proposition above into one general upper bound.

Theorem 6.40. *The function* $\max\{h_1^{\text{up}}, h_2^{\text{up}}\}$ *gives an upper bound for the competitive ratio of* SmarterStart *for closed online* Dial-a-Ride *and closed online* TSP *for all* $\Theta > 1$. *Let* $\Theta_{\text{S+}}^{\text{closed}} = 2$ *be the unique solution of* $h_1^{\text{up}}(\Theta) = h_2^{\text{up}}(\Theta)$, *i.e., of*

$$\frac{\Theta}{\Theta - 1} = \max\left\{\frac{\Theta + 2}{2}, \Theta\right\},$$

in the interval $(1, \infty)$. *Then,* $\Theta_{\text{S+}}^{\text{closed}}$ *is the unique minimum of the function* $\max\{h_1^{\text{up}}, h_2^{\text{up}}\}$ *and* SmarterStart *with scaling parameter* $\Theta_{\text{S+}}^{\text{closed}}$ *is* $\rho_{\text{S+}}^{\text{closed}}$*-competitive with*

$$\rho_{\text{S+}}^{\text{closed}} := h_1^{\text{up}}(\Theta_{\text{S+}}^{\text{closed}}) = h_2^{\text{up}}(\Theta_{\text{S+}}^{\text{closed}}) = 2.$$

Proof. For the case where SMARTERSTART postpones the final schedule we have established the upper bound

$$\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{\Theta}{\Theta - 1} = h_1^{\text{up}}(\Theta)$$

in Proposition 6.38, and for the case where SMARTERSTART does not postpone the final schedule we have established the upper bound

$$\frac{\text{SMARTERSTART}(\sigma)}{\text{OPT}(\sigma)} \leq \max\left\{\frac{\Theta + 2}{2}, \Theta\right\} = h_2^{\text{up}}(\Theta)$$

in Proposition 6.39. Thus, the maximum of both bounds is a general upper bound for the competitive ratio of SMARTERSTART for closed online DIAL-A-RIDE and closed online TSP that is independent of SMARTERSTART's behavior before the final schedule.

The function h_1^{up} is strictly decreasing for $\Theta > 1$ and the function h_2^{up} is strictly increasing for $\Theta > 1$. Therefore, the minimum of $\max\{h_1^{\text{up}}, h_2^{\text{up}}\}$ in the interval $(1, \infty)$ lies in the intersection point of h_1^{up} and h_2^{up}, i.e., in $\Theta_{\text{S+}}^{\text{closed}} = 2$. The resulting upper bound for the competitive ratio is

$$\rho_{\text{S+}}^{\text{closed}} = h_1^{\text{up}}(\Theta_{\text{S+}}^{\text{closed}}) = h_2^{\text{up}}(\rho_{\text{S+}}^{\text{closed}}) = 2. \qquad \square$$

The main theorem of this section follows by combining Theorem 6.40 and Theorem 4.3.

Theorem 6.41. *The competitive ratio of* SMARTERSTART *for closed online* DIAL-A-RIDE *and closed online* TSP *with scaling parameter* $\Theta_{\text{S+}}^{\text{closed}} = 2$ *is exactly*

$$\rho_{\text{S+}}^{\text{closed}} = h_1^{\text{up}}(\Theta_{\text{S+}}^{\text{closed}}) = h_2^{\text{up}}(\Theta_{\text{S+}}^{\text{closed}}) = 2.$$

There is no scaling parameter $\Theta > 1$ *with* $\Theta \neq \Theta_{\text{S+}}^{\text{D},\mathbb{R}}$ *that yields a better competitive ratio than* $\rho_{\text{S+}}^{\text{closed}}$.

Conclusion and Outlook

We provided a conclusive analysis for SMARTERSTART in this chapter. We computed tight bounds for the competitive ratio for open online DIAL-A-RIDE and open online TSP on the line and provided upper bounds for the open online DIAL-A-RIDE and open online TSP in the general setting. For the closed version, we provided tight bounds for the competive ratio of online DIAL-A-RIDE and online TSP for both the real line and the general setting. For the open version of online DIAL-A-RIDE on the line we have shown a tight competitive ratio of 2.6662: The upper bound was proven in Theorem 6.5 and the lower bound was proven in Theorem 6.19. For the open version of online TSP on the line we have shown a tight competitive ratio of 2.6288: The upper bound was proven in Theorem 6.23 and the lower bound was proven in Theorem 6.33. While we have tight results on the line for open online TSP, it remains unclear if SMARTERSTART performs worse in the general setting: We provided an upper bound of 2.6956 for the competitive ratio of SMARTERSTART for open online DIAL-A-RIDE and an upper bound of 2.6625 for the competitive ratio of SMARTERSTART for open online TSP. The lower bounds obtained on the real line carry over to the general setting. See Table 2.8 for a summary of the results.

If we compare SMARTERSTART with SMARTSTART, we see that SMARTERSTART has a better competitive ratio for the open version of the problems, while achieving the same competitive ratio as SMARTSTART for the closed version. Consequently, SMARTERSTART further improves the upper bound for the competitive ratio of online DIAL-A-RIDE on the real line as well as on general continuous metric spaces, improving the best known bound from SMARTSTART's 2.9377 to 2.6662 on the real line and from 3 to 2.6956 in the general setting. Moreover, for open online DIAL-A-RIDE on the line, the gap between the upper bound for the competitive ratio of SMARTERSTART of 2.6662 and the best known lower bound for the competitive ratio of schedule-based algorithms of 2.5 is rather small. Nonetheless, it remains unclear, if there is a better schedule-based algorithm for open online DIAL-A-RIDE on the line. For open online TSP on the line the gap is a bit larger: While SMARTERSTART is roughly 2.6288-competitive, our lower bound for schedule-based algorithms is $\frac{7}{3}$. Again, it remains unclear, if there is a better schedule-based algorithm for open online TSP on the line.

7 Algorithm REPLAN

After having analyzed schedule-based algorithms thoroughly, we now examine an online algorithm for online DIAL-A-RIDE and online TSP that has a completely different design philosophy. Like schedule-based algorithms, the algorithm REPLAN (see Algorithm 5) executes optimum offline walks. However, unlike schedule-based algorithms, these walks are recomputed every time a new request is released.

Algorithm 5 REPLAN

repeat
  if *new request appears* **then**
    Start optimal walk serving unserved requests R_t starting from current position

The algorithm REPLAN was first examined in [5]. Ascheuer et al. showed that the algorithm is $\frac{5}{2}$-competitive for closed online DIAL-A-RIDE with capacity $c = 1$. Krumke [32], one of the authors of [5], examined the algorithm more thoroughly in his PhD thesis. He showed that the algorithm is $\frac{7}{2}$-competitive for closed online DIAL-A-RIDE with capacity $c > 1$ and 3-competitive for open online DIAL-A-RIDE with capacity $c = 1$ as well as $\frac{9}{2}$-competitive for open online DIAL-A-RIDE with capacity $c > 1$. For open online TSP, Ausiello et al. showed that the algorithm has a tight competitive ratio of $\frac{5}{2}$ [8]. REPLAN was the best known online algorithm for open online TSP until Bjelde et al. published a 2.4142-competitive algorithm in [13]. See Table 2.4 for a summary of known results.

In this thesis, we present a lower bound of 2 for the competitive ratio of REPLAN for closed online TSP on the line (Thm 7.4). We complement this lower bound with a matching upper bound for closed online TSP on the line (Thm 7.5). The upper bound for closed online TSP in the general setting remains $\frac{5}{2}$. For closed online DIAL-A-RIDE, we provide an upper bound of 3 for capacity $c > 1$ and on the line (Thm 7.6) and an upper bound of $\frac{5}{2}$ for capacity $c = \infty$ in the general setting (Thm 7.7). For the open version of online DIAL-A-RIDE, we improve Krumke's upper bound of $\frac{9}{2}$ for capacity $c > 1$ to a bound of 4 for capacity $1 < c < \infty$ (Thm 7.8) and to a bound of 3 for capacity $c = \infty$ (Thm 7.9). We begin our analysis with the lower bound construction for closed online DIAL-A-RIDE and closed online TSP and analyze open online TSP and online DIAL-A-RIDE in the second section of this chapter.

7.1 Bounds for Closed Online DIAL-A-RIDE and online TSP

We start by proving that the competitive ratio of the closed version of REPLAN is larger or equal to 2. Essentially, the idea of the lower bound construction is to force the server to stay in ε range to the origin until the last request is released. This way, after the last request is released, REPLAN still has to move almost the complete distance OPT moves. Consequently, REPLAN's total completion time is almost twice as large as OPT's. To be more precise: For every sufficiently small $\varepsilon > 0$, we provide a request sequence $\sigma^{\text{cl}}_{\text{RP},m}$ such that

$$\text{REPLAN}(\sigma^{\text{cl}}_{\text{RP},m}) = (2-\varepsilon)\text{OPT}(\sigma^{\text{cl}}_{\text{RP},m}).$$

We start by defining the request sequence $\sigma^{\text{cl}}_{\text{RP},m}$.

Definition 7.1. Let $m \in \mathbb{N}$ with $m \geq 2$. We define

$$\sigma^{\text{cl}}_{\text{RP},m} := (s^L_0, s^R, s^L_1, \dots, s^L_{2m-2})$$

with

$$\begin{aligned} s^L_0 &= \left(-\frac{1}{m}; 0\right), \\ s^R &= \left(1-\frac{1}{m}; \frac{1}{m}\right), \\ s^L_i &= \left(-\frac{1}{m}; \frac{i}{m} + \frac{1}{2m}\right) \qquad \text{for } i \in \{1, \dots, 2m-2\}. \end{aligned}$$

We begin our analysis of $\sigma^{\text{cl}}_{\text{RP},m}$ with the computation of $\text{OPT}(\sigma^{\text{cl}}_{\text{RP},m})$.

Lemma 7.2. *We have*

$$\text{OPT}(\sigma^{\text{cl}}_{\text{RP},m}) = 2.$$

Proof. OPT performs the walk

$$0 \to 1 - \frac{1}{m} \to -\frac{1}{m} \to 0.$$

An illustration of OPT's walk is presented in blue in Figure 7.1. We show that all requests are served this way: OPT collects s^R at time $1 - \frac{1}{m}$ which is after time $r^R = \frac{1}{m}$ since

$m \geq 2$. The last request on the other side of the origin at position $-\frac{1}{m}$ is released at time $r^L_{2m-2} = 2 - \frac{3}{2m}$, which is before OPT reaches position $a^L_i = -\frac{1}{m}$ at time $2 - \frac{1}{m}$. Therefore, we have

$$\text{OPT}(\sigma^{\text{cl}}_{\text{RP},m}) = D\left(0 \to 1 - \frac{1}{m} \to -\frac{1}{m} \to 0\right) = 2. \qquad \square$$

Next, we compute REPLAN's completion time.

Lemma 7.3. *We have*

$$\text{REPLAN}(\sigma^{\text{cl}}_{\text{RP},m}) = 4 - \frac{2}{m}.$$

Proof. REPLAN's walk is presented in green in Figure 7.1. We show that at the release times of s^L_i for $i \geq 1$, REPLAN is always at position $-\frac{1}{2m}$. Since no requests except s^L_0 are released before time $\frac{1}{m}$, REPLAN serves request s^L_0 at time $\frac{1}{m}$, i.e., REPLAN is at position $-\frac{1}{m}$ at time $\frac{1}{m}$ and then moves towards the origin. Thus, at time $r^L_1 = \frac{3}{2m}$, when s^L_1 is released, REPLAN is at position $-\frac{1}{2m}$ as claimed. Now assume, REPLAN is at position $-\frac{1}{2m}$ at time $r^L_i = \frac{2i+1}{2m}$, when s^L_i is released. The shortest walk serving s^R before s^L_i has length

$$D\left(-\frac{1}{2m} \to 1 - \frac{1}{m} \to -\frac{1}{m} \to 0\right) = 2 + \frac{1}{2m}.$$

On the other hand, the shortest walk serving s^R after s^L_i has length

$$D\left(-\frac{1}{2m} \to -\frac{1}{m} \to 1 - \frac{1}{m} \to 0\right) = 2 - \frac{1}{2m}.$$

Thus, REPLAN proceeds to serve s^L_i first and then walks towards a^R. Therefore, REPLAN is again at position $\frac{1}{2m}$ at time $r^L_{i+1} = \frac{2(i+1)+1}{2m}$. In particular, REPLAN is at position $\frac{1}{2m}$ at time $r^L_{2m-2} = 2 - \frac{3}{2m}$. In total, we have

$$\text{REPLAN}(\sigma^{\text{cl}}_{\text{RP},m}) = 2 - \frac{3}{2m} + D\left(-\frac{1}{2m} \to -\frac{1}{m} \to 1 - \frac{1}{m} \to 0\right) = 4 - \frac{2}{m}. \qquad \square$$

Equipped with Lemmas 7.2 and 7.3, we can compute a lower bound for the competitive ratio of REPLAN for closed online TSP on the line.

Theorem 7.4. *For every sufficiently small $\varepsilon > 0$, we have*

$$\frac{\text{REPLAN}(\sigma^{\text{cl}}_{\text{RP},m})}{\text{OPT}(\sigma^{\text{cl}}_{\text{RP},m})} \geq 2 - \varepsilon.$$

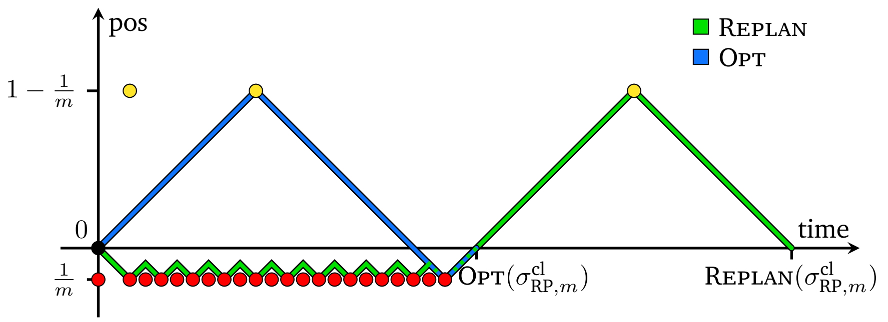

Figure 7.1: REPLAN's and OPT's walk serving $\sigma^{\text{cl}}_{\text{RP},m}$ with $m = 6$. Request s^R is yellow ● and requests s_i^L are red ●.

Proof. Let $\varepsilon \leq \frac{1}{m}$. By Lemma 7.3, we have

$$\text{REPLAN}(\sigma^{\text{cl}}_{\text{RP},m}) = 4 - \frac{2}{m}.$$

Lemma 7.2 implies

$$\text{OPT}(\sigma^{\text{cl}}_{\text{RP},m}) = 2.$$

Since we have $\varepsilon \leq \frac{1}{m}$, we obtain

$$\frac{\text{REPLAN}(\sigma^{\text{cl}}_{\text{RP},m})}{\text{OPT}(\sigma^{\text{cl}}_{\text{RP},m})} = 2 - \frac{1}{m} \geq 2 - \varepsilon. \qquad \square$$

Next, we examine REPLAN for closed online DIAL-A-RIDE and online TSP on the line, presenting a tight upper bound using line-specific features. Recall that we denote by

$$x^{\min} := \min\{0, a_1, \ldots, a_n, b_1, \ldots, b_n\}$$

the leftmost and by

$$x^{\max} := \max\{0, a_1, \ldots, a_n, b_1, \ldots, b_n\}$$

the rightmost position that needs to be visited by the server to serve σ. We have

$$\text{OPT}(\sigma) \geq 2x^{\max} + 2|x^{\min}|, \tag{7.1}$$

since OPT has to visit both extreme points and has to return to the origin.

Theorem 7.5. REPLAN *for closed online* TSP *on the line is* 2*-competitive.*

Proof. Let r_n be the time when the last request is released. REPLAN's position $\text{pos}(r_n)$ at time r_n is in the interval $[x^{\text{min}}, x^{\text{max}}]$. All requests are already released at time r_n, i.e., REPLAN can serve all remaining requests by visiting both extreme points and returning to the origin. If we have $\text{pos}(r_n) \geq 0$, we obtain

$$\begin{aligned}
\textsc{Replan}(\sigma) &\leq r_n + D\Big(\text{pos}(r_n) \to x^{\text{max}} \to x^{\text{min}} \to 0\Big) \\
&\leq r_n + 2x^{\text{max}} + 2|x^{\text{min}}| \\
&\overset{(7.1)}{\leq} r_n + \textsc{Opt}(\sigma) \\
&\overset{\textsc{Opt}(\sigma) \geq r_n}{\leq} 2\textsc{Opt}(\sigma).
\end{aligned}$$

Analogously, if we have $\text{pos}(r_n) < 0$, we obtain

$$\begin{aligned}
\textsc{Replan}(\sigma) &\leq r_n + D\Big(\text{pos}(r_n) \to x^{\text{min}} \to x^{\text{max}} \to 0\Big) \\
&< r_n + 2x^{\text{max}} + 2|x^{\text{min}}| \\
&\overset{(7.1)}{\leq} r_n + \textsc{Opt}(\sigma) \\
&\overset{\textsc{Opt}(\sigma) \geq r_n}{\leq} 2\textsc{Opt}(\sigma).
\end{aligned}$$

□

Next, we examine closed online DIAL-A-RIDE on the line for capacity $c > 1$ and provide an improved upper bound of 3 for REPLAN's competitive ratio.

Theorem 7.6. REPLAN *for closed online* DIAL-A-RIDE *on the line with capacity* $c > 1$ *is* 3*-competitive.*

Proof. Let r_n be the time when the last request is released. We consider two cases depending on whether or not the REPLAN server has loaded requests at time r_n. Assume the server is at position $\text{pos}(r_n)$ at time r_n and is empty. Then REPLAN can serve all remaining requests by returning to the origin and starting an optimum offline walk from there. This gives us

$$\begin{aligned}
\textsc{Replan}(\sigma) &\leq r_n + d(\text{pos}(r_n), 0) + \textsc{Opt}(\sigma) \\
&\overset{\textsc{Opt}(\sigma) \geq r_n}{\leq} d(\text{pos}(r_n), 0) + 2\textsc{Opt}(\sigma) \\
&\overset{\textsc{Opt}(\sigma) \geq 2d(\text{pos}(r_n), 0)}{\leq} \frac{5}{2}\textsc{Opt}(\sigma).
\end{aligned}$$

Note that we have $\textsc{Opt}(\sigma) \geq 2d(\text{pos}(r_n), 0)$ since $\textsc{Opt}$ has to visit $\text{pos}(r_n)$ and has to return to the origin. Now assume the server has loaded $k \leq c$ requests. $\textsc{Replan}$'s position $\text{pos}(r_n)$ at time r_n is in the interval $[x^{\min}, x^{\max}]$. $\textsc{Replan}$ can deliver all loaded requests by visiting both extreme points. Afterwards, $\textsc{Replan}$ can serve all remaining requests by returning to the origin and starting an optimum offline walk from there. If we have $\text{pos}(r_n) \geq 0$, we obtain

$$\begin{aligned}
\textsc{Replan}(\sigma) &\leq r_n + D\Big(\text{pos}(r_n) \to x^{\max} \to x^{\min} \to 0\Big) + \textsc{Opt}(\sigma) \\
&\leq r_n + 2x^{\max} + 2|x^{\min}| + \textsc{Opt}(\sigma) \\
&\overset{(7.1)}{\leq} r_n + 2\textsc{Opt}(\sigma) \\
&\overset{\textsc{Opt}(\sigma) \geq r_n}{\leq} 3\textsc{Opt}(\sigma).
\end{aligned}$$

Analogously, if we have $\text{pos}(r_n) < 0$, we obtain

$$\begin{aligned}
\textsc{Replan}(\sigma) &\leq r_n + D\Big(\text{pos}(r_n) \to x^{\min} \to x^{\max} \to 0\Big) + \textsc{Opt}(\sigma) \\
&< r_n + 2x^{\max} + 2|x^{\min}| + \textsc{Opt}(\sigma) \\
&\overset{(7.1)}{\leq} r_n + 2\textsc{Opt}(\sigma) \\
&\overset{\textsc{Opt}(\sigma) \geq r_n}{\leq} 3\textsc{Opt}(\sigma).
\end{aligned}$$

□

Finally, we examine the closed version of $\textsc{Replan}$ in the general setting for capacity $c = \infty$.

Theorem 7.7. $\textsc{Replan}$ *for closed online* $\textsc{Dial-a-Ride}$ *with* $c = \infty$ *is* $\frac{5}{2}$*-competitive.*

Proof. Let r_n be the time when the last request is released and let $\text{pos}(r_n)$ be the position of the $\textsc{Replan}$ server at time r_n. Since the server has an infinite capacity, $\textsc{Replan}$ can just return to the origin and start an optimum offline walk from there serving all remaining requests. This gives us

$$\begin{aligned}
\textsc{Replan}(\sigma) &\leq r_n + d(\text{pos}(r_n), 0) + \textsc{Opt}(\sigma) \\
&\overset{\textsc{Opt}(\sigma) \geq r_n}{\leq} d(\text{pos}(r_n), 0) + 2\textsc{Opt}(\sigma) \\
&\overset{\textsc{Opt}(\sigma) \geq 2d(\text{pos}(r_n), 0)}{\leq} \frac{5}{2}\textsc{Opt}(\sigma).
\end{aligned}$$

Note that we have $\textsc{Opt}(\sigma) \geq 2d(\text{pos}(r_n), 0)$ since $\textsc{Opt}$ has to visit $\text{pos}(r_n)$ and has to return to the origin. □

7.2 Bounds for Open Online DIAL-A-RIDE and Online TSP

In this subsection we provide improved bounds for the open version of REPLAN. We start with the case that the server has a capacity larger than 1.

Theorem 7.8. REPLAN *for open online* DIAL-A-RIDE *on the line with capacity* $c > 1$ *is* 4*-competitive.*

Proof. Let r_n be the time when the last request is released. We consider two cases depending on whether or not the REPLAN server has loaded requests at time r_n. Assume the server is at position $\text{pos}(r_n)$ at time r_n and is empty. Then REPLAN can serve all remaining requests by returning to the origin and starting an optimum offline walk from there. This gives us

$$\begin{aligned}
\text{REPLAN}(\sigma) &\leq r_n + d(\text{pos}(r_n), 0) + \text{OPT}(\sigma) \\
&\overset{\text{OPT}(\sigma) \geq r_n}{\leq} d(\text{pos}(r_n), 0) + 2\text{OPT}(\sigma) \\
&\overset{\text{OPT}(\sigma) \geq d(\text{pos}(r_n), 0)}{\leq} 3\text{OPT}(\sigma).
\end{aligned}$$

Now assume the server has loaded $k \leq c$ requests. REPLAN serves all remaining requests by delivering all loaded requests, returning to the origin and starting an optimum offline walk from there. By construction, REPLAN's position $\text{pos}(r_n)$ is on the shortest way between two positions $x, y \in X$ with

$$x, y \in \{0, a_1, \ldots, a_n, b_1, \ldots, b_n\}.$$

Let W be an optimum open offline walk with length $c(W)$ starting from the origin, visiting all destinations of the requests currently loaded by REPLAN as well as x and y, and ending in some position $z \in X$. Since OPT also has to visit all destinations of the requests currently loaded by REPLAN as well as x and y, we have $\text{OPT}(\sigma) \geq c(W)$. Furthermore, by triangle inequality, we have

$$\begin{aligned}
\text{OPT}(\sigma) &\geq c(W) \\
&\geq D(0 \to x \to y \to z) \\
&= D(0 \to x \to \text{pos}(r_n) \to y \to z) \\
&\geq D(0 \to \text{pos}(r_n) \to z) \\
&\geq d(\text{pos}(r_n), z).
\end{aligned} \tag{7.2}$$

Note that we used the fact that $\text{pos}(r_n)$ lies on a shortest way between x and y in the inequality above. Replan can deliver all loaded requests and return to the origin by moving from $\text{pos}(r_n)$ to z and then walking the walk W backwards. To sum it up, we have

$$\begin{aligned}
\text{Replan}(\sigma) &\leq r_n + d(\text{pos}(r_n), z) + c(W) + \text{Opt}(\sigma) \\
&\overset{\text{Opt}(\sigma) \geq c(W)}{\leq} r_n + d(\text{pos}(r_n), z) + 2\text{Opt}(\sigma) \\
&\overset{(7.2)}{\leq} r_n + 3\text{Opt}(\sigma) \\
&\overset{\text{Opt}(\sigma) \geq r_n}{\leq} 4\text{Opt}(\sigma).
\end{aligned}$$

□

For capacity $c = \infty$ we prove a stronger bound of 3.

Theorem 7.9. Replan *for open online* Dial-a-Ride *with capacity $c = \infty$ is 3-competitive.*

Proof. Let r_n be the time when the last request is released and let $\text{pos}(r_n)$ be the position of the Replan server at time r_n. Since the server has an infinite capacity, Replan can just return to the origin and start an optimum offline walk from there serving all remaining requests. This gives us

$$\begin{aligned}
\text{Replan}(\sigma) &\leq r_n + d(\text{pos}(r_n), 0) + \text{Opt}(\sigma) \\
&\overset{\text{Opt}(\sigma) \geq r_n}{\leq} d(\text{pos}(r_n), 0) + 2\text{Opt}(\sigma) \\
&\overset{\text{Opt}(\sigma) \geq d(\text{pos}(r_n), 0)}{\leq} 3\text{Opt}(\sigma).
\end{aligned}$$

□

Conclusion and Outlook

In this chapter, we improved several bounds for the competitive ratio of algorithm Replan. While we were able to prove tight bounds for closed online TSP on the line, a gap remains for most other versions of the algorithm. It is not clear, if Replan has different compatitive ratios for online Dial-a-Ride and online TSP. All known lower bounds for its competitive ratios only utilize TSP requests. Furthermore, it is not clear if Replan's competitiveness on the line is different to its competitiveness in the general setting since all known lower bounds are constructed on the line. Significant is that the best known upper bounds not only differ between finite and infinite capacities but also between unit capacity and larger capacities. Again, it is not clear if Replan's competitiveness is dependent on the capacity of the server or if just the necessary tools to conduct tighter analyses for larger capacities are missing. For an overview of all known bounds for Replan's competitive ratio including the bounds shown in this thesis, see Table 2.9.

Conclusion and Outlook

In this thesis, we analyzed the online optimization problem online Dial-a-Ride and its special case online TSP. In Chapter 3, we provided new lower bounds for the competitive ratio of open and closed online Dial-a-Ride on the line with finite capacity. Both bounds are inspired by the lower bound construction for open online TSP on the line from [13, Thm 4]. However, while the original construction relies on an iterative first stage, the bounds from this thesis only use a single iteration as first stage. It remains unclear if an iterative approach can also be applied to our lower bounds, potentially leading to improvements.

Concerning upper bounds, we analyze several online algorithms. The algorithms Ignore, Smartstart and SmarterStart have a similar design and belong to the class of schedule based algorithms. Algorithm SmarterStart attains the best competitive ratios of the three algorithms and is the best known online algorithm for open online Dial-a-Ride with finite capacity. However, there still remains a gap between SmarterStart's competitive ratio of roughly 2.6662 and the lower bound for the competitive ratio of open schedule-based algorithms of 2.5. It remains unclear, if SmarterStart's waiting routine can be improved or if the lower bound can be lifted. While SmarterStart uses the information about all released requests for the computation of its waiting time, it does not use the current position of the server for its computation. Using this information in a smart way could further improve SmarterStart. However, even if SmarterStart can be further improved to achieve a competitive ratio of 2.5 matching the lower bound for open schedule-based algorithms, its competitiveness still would be weaker than the competitive ratio of 2.4142 of the currently best known online algorithm for open online Dial-a-Ride with infinite capacity. This indicates that the schedule-based design, albeit achieving good results, is not optimal for open online Dial-a-Ride.

For the closed version of online Dial-a-Ride and online TSP, this is different. SmarterStart is the best possible schedule-based algorithm for closed online Dial-a-Ride and closed online TSP on the line as well as in the general setting with a competitive ratio of 2. Therefore, schedule-based algorithms for closed online Dial-a-Ride and online TSP are fully understood. Moreover, for the general setting, SmarterStart is a best-possible algorithm matching the lower bound of [8, Thm 3.2]. This shows that, at least for the general setting, the schedule-based design is best-possible for the closed version of online

DIAL-A-RIDE and online TSP. On the line, on the other hand, the general lower bound is roughly 1.7636, which raises the question if there is an online algorithm for closed online DIAL-A-RIDE that attains a competitive ratio strictly below 2. It is clear that such an algorithm cannot be schedule-based.

A good candidate seemed to be REPLAN. However, we showed that REPLAN for closed online DIAL-A-RIDE and TSP is at best 2-competitive. Additionally, since REPLAN for open online DIAL-A-RIDE and online TSP is at best 2.5-competitive, REPLAN's competitiveness is not better than the competitiveness of schedule-based algorithms. A hybrid of both designs could lead to an improvement, but for now this remains an open question.

Bibliography

[1] Susanne Albers. "Improved Randomized On-Line Algorithms for the List Update Problem". In: *SIAM Journal on Computing* 27.3 (1998), pp. 682–693. DOI: 10.1137/S0097539794277858.

[2] Susanne Albers, Arindam Khan, and Leon Ladewig. "Improved Online Algorithms for Knapsack and GAP in the Random Order Model". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2019, September 20-22, 2019, Massachusetts Institute of Technology, Cambridge, MA, USA*. Ed. by Dimitris Achlioptas and László A. Végh. Vol. 145. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, 22:1–22:23. DOI: 10.4230/LIPIcs.APPROX-RANDOM.2019.22.

[3] Susanne Albers, Bernhard von Stengel, and Ralph Werchner. "A Combined BIT and TIMESTAMP Algorithm for the List Update Problem". In: *Inf. Process. Lett.* 56.3 (1995), pp. 135–139. DOI: 10.1016/0020-0190(95)00142-Y.

[4] Yuichi Asahiro, Eiji Miyano, Shuichi Miyazaki, and Takuro Yoshimuta. "Weighted nearest neighbor algorithms for the graph exploration problem on cycles". In: *Inf. Process. Lett.* 110.3 (2010), pp. 93–98. DOI: 10.1016/j.ipl.2009.10.013.

[5] Norbert Ascheuer, Sven Oliver Krumke, and Jörg Rambau. "Online Dial-a-Ride Problems: Minimizing the Completion Time". In: *STACS 2000, 17th Annual Symposium on Theoretical Aspects of Computer Science, Lille, France, February 2000, Proceedings*. Ed. by Horst Reichel and Sophie Tison. Vol. 1770. Lecture Notes in Computer Science. Springer, 2000, pp. 639–650. DOI: 10.1007/3-540-46541-3_53.

[6] Mikhail J. Atallah and S. Rao Kosaraju. "Efficient Solutions to Some Transportation Problems with Applications to Minimizing Robot Arm Travel". In: *SIAM J. Comput.* 17.5 (1988), pp. 849–869. DOI: 10.1137/0217053.

[7] Giorgio Ausiello, Marc Demange, Luigi Laura, and Vangelis Th. Paschos. "Algorithms for the On-Line Quota Traveling Salesman Problem". In: *Inf. Process. Lett.* 92.2 (2004), pp. 89–94. DOI: 10.1016/j.ipl.2004.06.013.

[8] Giorgio Ausiello, Esteban Feuerstein, Stefano Leonardi, Leen Stougie, and Maurizio Talamo. "Algorithms for the On-Line Travelling Salesman". In: *Algorithmica* 29.4 (2001), pp. 560–581. DOI: 10.1007/s004530010071.

[9] Shai Ben-David, Allan Borodin, Richard M. Karp, Gábor Tardos, and Avi Wigderson. "On the Power of Randomization in Online Algorithms (Extended Abstract)". In: *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*. Ed. by Harriet Ortiz. ACM, 1990, pp. 379–386. DOI: 10.1145/100216.100268.

[10] Alexander Birx and Yann Disser. "Tight Analysis of the Smartstart Algorithm for Online Dial-a-Ride on the Line". In: *SIAM J. Discret. Math.* 34.2 (2020), pp. 1409–1443. DOI: 10.1137/19M1268513.

[11] Alexander Birx, Yann Disser, Alexander V. Hopp, and Christina Karousatou. "Improved Lower Bound for Competitive Graph Exploration". In: *CoRR* abs/2002.10958 (2020). arXiv: 2002.10958.

[12] Alexander Birx, Yann Disser, and Kevin Schewior. "Improved Bounds for Open Online Dial-a-Ride on the Line". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2019, September 20-22, 2019, Massachusetts Institute of Technology, Cambridge, MA, USA*. Vol. 145. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, 21:1–21:22. DOI: 10.4230/LIPIcs.APPROX-RANDOM.2019.21.

[13] Antje Bjelde, Yann Disser, Jan Hackfeld, Christoph Hansknecht, Maarten Lipmann, Julie Meißner, Kevin Schewior, Miriam Schlöter, and Leen Stougie. "Tight Bounds for Online TSP on the Line". In: *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*. Ed. by Philip N. Klein. SIAM, 2017, pp. 994–1005. DOI: 10.1137/1.9781611974782.63.

[14] Michiel Blom, Sven Oliver Krumke, Willem de Paepe, and Leen Stougie. "The Online TSP Against Fair Adversaries". In: *INFORMS J. Comput.* 13.2 (2001), pp. 138–148. DOI: 10.1287/ijoc.13.2.138.10517.

[15] Sebastian Brandt, Klaus-Tycho Foerster, Jonathan Maurer, and Roger Wattenhofer. "Online Graph Exploration on a Restricted Graph Class: Optimal Solutions for Tadpole Graphs". In: *CoRR* abs/1903.00581 (2019). arXiv: 1903.00581. URL: http://arxiv.org/abs/1903.00581.

[16] Moses Charikar and Balaji Raghavachari. "The Finite Capacity Dial-A-Ride Problem". In: *39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA*. IEEE Computer Society, 1998, pp. 458–467. DOI: 10.1109/SFCS.1998.743496.

[17] Pei-Chuan Chen, Erik D. Demaine, Chung-Shou Liao, and Hao-Ting Wei. "Waiting is not easy but worth it: the online TSP on the line revisited". In: *CoRR* abs/1907.00317 (2019). arXiv: 1907.00317.

[18] Marek Chrobak, Howard J. Karloff, T. H. Payne, and Sundar Vishwanathan. "New Results on Server Problems". In: *SIAM J. Discret. Math.* 4.2 (1991), pp. 172–181. DOI: 10.1137/0404017.

[19] Marek Chrobak and Lawrence L. Larmore. "An Optimal On-Line Algorithm for k-Servers on Trees". In: *SIAM J. Comput.* 20.1 (1991), pp. 144–148. DOI: 10.1137/0220008.

[20] Marek Chrobak and Lawrence L. Larmore. "On Fast Algorithms for Two Servers". In: *J. Algorithms* 12.4 (1991), pp. 607–614. DOI: 10.1016/0196-6774(91)90035-W.

[21] Xiaotie Deng and Christos H. Papadimitriou. "Exploring an Unknown Graph". In: *Proceedings of the 31st Annual Symposium on Foundations of Computer Science (FOCS)*. 1990, 355–361 vol. 1. ISBN: 0-8186-2082-X. DOI: 10.1109/FSCS.1990.89554.

[22] Xiaotie Deng and Christos H. Papadimitriou. "Exploring an unknown graph". In: *Journal of Graph Theory* 32.3 (1999), pp. 265–297. DOI: 10.1002/(SICI)1097-0118(199911)32:3<265::AID-JGT6>3.0.CO;2-8.

[23] Dariusz Dereniowski, Yann Disser, Adrian Kosowski, Dominik Pajak, and Przemyslaw Uznanski. "Fast collaborative graph exploration". In: *Inf. Comput.* 243 (2015), pp. 37–49. DOI: 10.1016/j.ic.2014.12.005.

[24] Stefan Dobrev, Rastislav Královic, and Euripides Markou. "Online Graph Exploration with Advice". In: *Structural Information and Communication Complexity - 19th International Colloquium, SIROCCO 2012, Reykjavik, Iceland, June 30-July 2, 2012, Revised Selected Papers*. Ed. by Guy Even and Magnús M. Halldórsson. Vol. 7355. Lecture Notes in Computer Science. Springer, 2012, pp. 267–278. DOI: 10.1007/978-3-642-31104-8_23.

[25] Esteban Feuerstein and Leen Stougie. "On-line single-server dial-a-ride problems". In: *Theor. Comput. Sci.* 268.1 (2001), pp. 91–105. DOI: 10.1016/S0304-3975(00)00261-9.

[26] Klaus-Tycho Foerster and Roger Wattenhofer. "Lower and upper competitive bounds for online directed graph exploration". In: *Theor. Comput. Sci.* 655 (2016), pp. 15–29. DOI: 10.1016/j.tcs.2015.11.017.

[27] Paul C. Gilmore and Ralph E. Gomory. "Sequencing a One State-Variable Machine: A Solvable Case of the Traveling Salesman Problem". In: *Operations Research* 12.5 (1964), pp. 655–679. DOI: 10.1287/opre.12.5.655.

[28] Dih Jiun Guan. "Routing a Vehicle of Capacity Greater than one". In: *Discret. Appl. Math.* 81.1-3 (1998), pp. 41–57. DOI: 10.1016/S0166-218X(97)00074-7.

[29] Dietrich Hauptmeier, Sven Oliver Krumke, and Jörg Rambau. "The Online Dial-a-Ride Problem under Reasonable Load". In: *Algorithms and Complexity, 4th Italian Conference, CIAC 2000, Rome, Italy, March 2000, Proceedings*. Ed. by Gian Carlo Bongiovanni, Giorgio Gambosi, and Rossella Petreschi. Vol. 1767. Lecture Notes in Computer Science. Springer, 2000, pp. 125–136. DOI: 10.1007/3-540-46521-9_11.

[30] Bala Kalyanasundaram and Kirk Pruhs. "Constructing Competitive Tours from Local Information". In: *Theor. Comput. Sci.* 130.1 (1994), pp. 125–138. DOI: 10.1016/0304-3975(94)90155-4.

[31] Anna R. Karlin, Mark S. Manasse, Larry Rudolph, and Daniel Dominic Sleator. "Competitive Snoopy Caching". In: *Algorithmica* 3 (1988), pp. 77–119. DOI: 10.1007/BF01762111.

[32] Sven O. Krumke. "Online Optimization Competitive Analysis and Beyond". PhD thesis. Technical University Berlin, 2001.

[33] Sven Oliver Krumke, Luigi Laura, Maarten Lipmann, Alberto Marchetti-Spaccamela, Willem de Paepe, Diana Poensgen, and Leen Stougie. "Non-abusiveness Helps: An O(1)-Competitive Algorithm for Minimizing the Maximum Flow Time in the Online Traveling Salesman Problem". In: *Approximation Algorithms for Combinatorial Optimization, 5th International Workshop, APPROX 2002, Rome, Italy, September 17-21, 2002, Proceedings*. Ed. by Klaus Jansen, Stefano Leonardi, and Vijay V. Vazirani. Vol. 2462. Lecture Notes in Computer Science. Springer, 2002, pp. 200–214. DOI: 10.1007/3-540-45753-4_18.

[34] Sven Oliver Krumke, Willem de Paepe, Diana Poensgen, Maarten Lipmann, Alberto Marchetti-Spaccamela, and Leen Stougie. "On Minimizing the Maximum Flow Time in the Online Dial-a-Ride Problem". In: *Approximation and Online Algorithms, Third International Workshop, WAOA 2005, Palma de Mallorca, Spain, October 6-7, 2005, Revised Papers*. Ed. by Thomas Erlebach and Giuseppe Persiano. Vol. 3879.

Lecture Notes in Computer Science. Springer, 2005, pp. 258–269. DOI: 10.1007/11671411_20.

[35] Maarten Lipmann. "On-Line Routing". PhD thesis. Technical University Eindhoven, 2003.

[36] Maarten Lipmann, Xiwen Lu, Willem de Paepe, René Sitters, and Leen Stougie. "On-Line Dial-a-Ride Problems Under a Restricted Information Model". In: *Algorithmica* 40.4 (2004), pp. 319–329. DOI: 10.1007/s00453-004-1116-z.

[37] Mark S. Manasse, Lyle A. McGeoch, and Daniel Dominic Sleator. "Competitive Algorithms for Server Problems". In: *J. Algorithms* 11.2 (1990), pp. 208–230. DOI: 10.1016/0196-6774(90)90003-W.

[38] Alberto Marchetti-Spaccamela and Carlo Vercellis. "Stochastic on-line knapsack problems". In: *Math. Program.* 68 (1995), pp. 73–104. DOI: 10.1007/BF01585758.

[39] Nicole Megow, Kurt Mehlhorn, and Pascal Schweitzer. "Online graph exploration: New results on old and new algorithms". In: *Theor. Comput. Sci.* 463 (2012), pp. 62–72. DOI: 10.1016/j.tcs.2012.06.034.

[40] Shuichi Miyazaki, Naoyuki Morimoto, and Yasuo Okabe. "The Online Graph Exploration Problem on Restricted Graphs". In: *IEICE Trans. Inf. Syst.* 92-D.9 (2009), pp. 1620–1627. DOI: 10.1587/transinf.E92.D.1620.

[41] Willem de Paepe, Jan Karel Lenstra, Jiri Sgall, René A. Sitters, and Leen Stougie. "Computer-Aided Complexity Classification of Dial-a-Ride Problems". In: *INFORMS J. Comput.* 16.2 (2004), pp. 120–132. DOI: 10.1287/ijoc.1030.0052.

[42] Daniel Rosenkrantz, Richard Edwin Stearns, and Philip M. Lewis II. "An Analysis of Several Heuristics for the Traveling Salesman Problem". In: *SIAM Journal on Computing* 6 (1977), pp. 563–581. DOI: 10.1137/0206041.

[43] David B. Shmoys, Joel Wein, and David P. Williamson. "Scheduling Parallel Machines On-Line". In: *SIAM J. Comput.* 24.6 (1995), pp. 1313–1331. DOI: 10.1137/S0097539793248317.

[44] Daniel Dominic Sleator and Robert Endre Tarjan. "Amortized Efficiency of List Update and Paging Rules". In: vol. 28. 2. 1985, pp. 202–208. DOI: 10.1145/2786.2793.

[45] Fanglei Yi and Lei Tian. "On the Online Dial-A-Ride Problem with Time-Windows". In: *Algorithmic Applications in Management, First International Conference, AAIM 2005, Xian, China, June 22-25, 2005, Proceedings*. Ed. by Nimrod Megiddo, Yinfeng Xu, and Binhai Zhu. Vol. 3521. Lecture Notes in Computer Science. Springer, 2005, pp. 85–94. DOI: 10.1007/11496199_11.

[46] Fanglei Yi, Yinfeng Xu, and Chunlin Xin. “Online Dial-A-Ride Problem with Time-Windows Under a Restricted Information Model”. In: *Algorithmic Aspects in Information and Management, Second International Conference, AAIM 2006, Hong Kong, China, June 20-22, 2006, Proceedings*. Ed. by Siu-Wing Cheng and Chung Keung Poon. Vol. 4041. Lecture Notes in Computer Science. Springer, 2006, pp. 22–31. DOI: 10.1007/11775096_4.

Curriculum Vitæ

10/2011 – 09/2014	**Bachelor's studies**, *Technische Universität Darmstadt*, Darmstadt, Germany, Bachelor of Science Mathematics, Bachelor's thesis: *Das Theorem von Riemann-Roch*
10/2014 – 08/2017	**Master's studies**, *Technische Universität Darmstadt*, Darmstadt, Germany, Master of Science Mathematics, Master's thesis: *Graphical Models with Total Cardinality Constraint*
09/2017 – 08/2020	**Doctoral scholarship** of the Graduate School of Computational Engineering
08/07/2020	**Submission of the doctor's thesis (Dissertation)**, *Competitive analysis of the online dial-a-ride problem* at Technische Universität Darmstadt, Darmstadt, Germany
09/10/2020	**Defense of the doctor's thesis**, overall assessment: *magna cum laude / sehr gut bestanden*