Online Scheduling of Car-Sharing Requests Between Two Locations with Many Cars and Flexible Advance Bookings

Kelin Luo[1]
School of Management, Xi'an Jiaotong University, Xi'an, China
luokelin@xjtu.edu.cn
https://orcid.org/0000-0003-2006-0601

Thomas Erlebach
Department of Informatics, University of Leicester, Leicester, United Kingdom
t.erlebach@leicester.ac.uk
https://orcid.org/0000-0002-4470-5868

Yinfeng Xu
School of Management, Xi'an Jiaotong University, Xi'an, China
yfxu@xjtu.edu.cn

Abstract

We study an on-line scheduling problem that is motivated by applications such as car-sharing, in which users submit ride requests, and the scheduler aims to accept requests of maximum total profit using k servers (cars). Each ride request specifies the pick-up time and the pick-up location (among two locations, with the other location being the destination). The scheduler has to decide whether or not to accept a request immediately at the time when the request is submitted (booking time). We consider two variants of the problem with respect to constraints on the booking time: In the fixed booking time variant, a request must be submitted a fixed amount of time before the pick-up time. In the variable booking time variant, a request can be submitted at any time during a certain time interval (called the booking horizon) that precedes the pick-up time. We present lower bounds on the competitive ratio for both variants and propose a balanced greedy algorithm (BGA) that achieves the best possible competitive ratio. We prove that, for the fixed booking time variant, BGA is 1.5-competitive if $k = 3i$ ($i \in \mathbb{N}$) and the fixed booking length is not less than the travel time between the two locations; for the variable booking time variant, BGA is 1.5-competitive if $k = 3i$ ($i \in \mathbb{N}$) and the length of the booking horizon is less than the travel time between the two locations, and BGA is 5/3-competitive if $k = 5i$ ($i \in \mathbb{N}$) and the length of the booking horizon is not less than the travel time between the two locations.

2012 ACM Subject Classification Theory of computation → Online algorithms

Keywords and phrases Car-sharing system, Competitive analysis, On-line scheduling

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2018.64

[1] This work was partially supported by the China Postdoctoral Science Foundation (Grant No. 2016M592811), and the China Scholarship Council (Grant No. 201706280058).

29th International Symposium on Algorithms and Computation (ISAAC 2018).
Editors: Wen-Lian Hsu, Der-Tsai Lee, and Chung-Shou Liao; Article No. 64; pp. 64:1–64:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

In a car-sharing system, a company offers cars to customers for a period of time. Customers can pick up a car in one location, drive it to another location, and return it there. Car booking requests arrive on-line, and the goal is to maximize the profit obtained from satisfied requests. We refer to this problem as the *car-sharing problem*.

In a real setting, customer requests for car bookings arrive over time, and the decision about each request must be made immediately, without knowledge of future requests. This gives rise to an on-line problem that bears some resemblance to interval scheduling, but in which additionally the pick-up and drop-off locations play an important role: The server that serves a request must be at the pick-up location at the start time of the request and will be located at the drop-off location at the end time of the request. We consider a setting where all driving routes go between two fixed locations, but can be in either direction. For example, the two locations could be a residential area and a nearby shopping mall or central business district. Other applications that provide motivation for the problems we study include car rental, taxi dispatching and boat rental for river crossings. A server can serve two consecutive requests only if the drop-off location of the first request is the same as the pick-up location of the second request, or if there is enough time to travel between the two locations otherwise. We allow *empty movements*, i.e., a server can be moved from one location to another while not serving a request. Such empty movements could be implemented by having company staff drive a car from one location to another, or in the future by self-driving cars.

With respect to constraints on the booking time, one can consider the *fixed booking time* variant and the *variable booking time* variant of the car-sharing problem [7]. The fixed booking time variant requires users to submit requests in such a way that the amount of time between the booking time of a request and its start time is a fixed value, independent of the request. This simplifies the scheduling task because the order of the start times of the requests is the same as the order of their release times (booking times). It is, however, less convenient for users because they have to book a request at a specific time. In the *variable booking time* variant, the booking time of a request must lie in a certain time interval (called the *booking horizon*) before the start time of the request. Users can book a request at any time in this interval.

1.1 Related Work

In [7], the authors studied the car-sharing problem for the special case of two locations and a single server, considering both fixed booking times and variable booking times, and presented tight results for the competitive ratio. The optimal competitive ratio was shown to be 2 for fixed booking times and 3 for variable booking times. In [8], the authors dealt with the car-sharing problem with two locations and two servers, considering only the case of fixed booking times, and presented tight results for the competitive ratio. The optimal competitive ratio was shown to be 2. In contrast to the previous work on car-sharing between two locations, in this paper we consider the car-sharing problem for both fixed booking times and variable booking times in the setting with k servers where k can be arbitrarily large. As a larger number of servers provides more flexibility to the algorithm, different lower bound constructions and different techniques for analyzing the competitive ratio of an algorithm are required. It seems natural to expect that a large number of servers can help an algorithm to achieve better competitive ratio, but our results show that, surprisingly, 3 servers (in one case) and 5 servers (in another case) already allow us to get the best competitive ratio, and no improvement is possible with more servers.

Böhmová et al. [3] showed that if all customer requests for car bookings are known in advance, the problem of maximizing the number of accepted requests is solvable in polynomial time. Furthermore, they considered the problem variant with two locations where each customer requests two rides (in opposite directions) and the scheduler must accept either both or neither of the two. They proved that this variant is NP-hard and APX-hard. In contrast to their work, we consider the on-line version of the problem with k servers.

Amongst other related work, the problem that is closest to our setting is the on-line dial-a-ride problem (OLDARP). In OLDARP, transportation requests between locations in a metric space arrive over time, but typically it is assumed that requests want to be served "as soon as possible" rather than at a specific time as in our problem. Versions of OLDARP with the objective of serving all requests while minimizing the makespan [1, 2] or the maximum flow time [6] have been widely studied in the literature. The versions of OLDARP where not all requests need to be served includes competitive algorithms for requests with deadlines where each request must be served before its deadline or rejected [9], and for settings with a given time limit where the goal is to maximize the revenue from requests served before the time limit [5]. In contrast to existing work on OLDARP, in this paper we consider requests that need to be served at a specific time that is specified by the request when it is released. Another related problem is the k-server problem [4, Ch. 10], but in that problem all requests must be served and requests are served at a specific location.

1.2 Problem Description and Preliminaries

We consider a setting with only two locations (denoted by 0 and 1) and k servers (denoted by $s_1, s_2, \ldots, s_k$). The k servers are initially located at location 0. The travel time from 0 to 1 is the same as the travel time from 1 to 0 and is denoted by t.

Let R denote a sequence of requests that are released over time. The i-th request is denoted by $r_i = (\tilde{t}_{r_i}, t_{r_i}, p_{r_i})$ and is specified by the *booking time* or *release time* $\tilde{t}_{r_i}$, the *start time* (or *pick-up time*) t_{r_i}, and the pick-up location $p_{r_i} \in \{0, 1\}$. If r_i is accepted, a server must pick up the customer at p_{r_i} at time t_{r_i} and drop off the customer at location $\dot{p}_{r_i} = 1 - p_{r_i}$, the *drop-off location* of the request, at time $\dot{t}_{r_i} = t_{r_i} + t$, the *end time* (or *drop-off time*) of the request. We assume that for all $r_i \in R$, t_{r_i} is an integer multiple of the travel time between location 0 and location 1, i.e., $t_{r_i} = \nu t$ for some $\nu \in \mathbb{N}$.

Each server can only serve one request at a time. Serving a request yields profit $r > 0$. An empty movement between the two locations takes time t, but has no cost. If two requests are such that they cannot both be served by the same server, we say that the requests are *in conflict*. We denote the set of requests accepted by an algorithm by R', and the i-th request in R', in order of request start times, is denoted by r'_i. We denote the profit of serving the requests in R' by $P_{R'} = r \cdot |R'|$. The goal of the car-sharing problem is to accept a set of requests R' that maximizes the profit $P_{R'}$.

The problem for k servers and two locations for the fixed booking time variant in which $t_{r_i} - \tilde{t}_{r_i} = a$ for all requests r_i, where $a \geq t$ is a constant, is called the *kS2L-F* problem. For the variable booking time variant, the booking time $\tilde{t}_{r_i}$ of any request r_i must satisfy $t_{r_i} - b_u \leq \tilde{t}_{r_i} \leq t_{r_i} - b_l$, where b_l and b_u are constants, with $t \leq b_l < b_u$, that specify the minimum and maximum length, respectively, of the time interval between booking time and start time. The problem for k servers and two locations for the variable booking time variant is called the *kS2L-V* problem. We do not require that the algorithm assigns an accepted request to a server immediately, provided that it ensures that one of the k servers will serve the request. In our setting, however, it is not necessary for an algorithm to use this flexibility.

Table 1 Lower and upper bounds on the competitive ratio for the car sharing problem.

Problem	Booking constraint	Lower bound	Upper bound
kS2L-F	$a \geq t$	1.5	1.5 ($k = 3i$, $i \in \mathbb{N}$)
kS2L-V	$b_l \geq t, b_u - b_l < t$	1.5	1.5 ($k = 3i$, $i \in \mathbb{N}$)
kS2L-V	$b_l \geq t, b_u - b_l \geq t$	5/3	5/3 ($k = 5i$, $i \in \mathbb{N}$)

The performance of an algorithm for kS2L-F or kS2L-V is measured using competitive analysis (see [4]). For any request sequence R, let P_{R^A} denote the objective value produced by an on-line algorithm A, and P_{R^*} that obtained by an optimal scheduler OPT that has full information about the request sequence in advance. The competitive ratio of A is defined as $\rho_A = \sup_R \frac{P_{R^*}}{P_{R^A}}$. We say that A is ρ-competitive if $P_{R^*} \leq \rho \cdot P_{R_A}$ for all request sequences R. Let ON be the set of all on-line algorithms for a problem. We only consider deterministic algorithms. A value β is a *lower bound* on the best possible competitive ratio if $\rho_A \geq \beta$ for all A in ON. We say that an algorithm A is *optimal* if there is a lower bound β with $\rho_A = \beta$.

1.3 Paper Outline

An overview of our results is shown in Table 1. In Section 2, we prove the lower bounds. In Section 3, we propose a balanced greedy algorithm that achieves the best possible competitive ratio. Although variable booking times provide much greater flexibility to customers, we show that our balanced greedy algorithm (only with a different choice of a parameter in the algorithm) is still optimal. When $k \neq 3i$ (resp. $k \neq 5i$), $i \in \mathbb{N}$, the upper bounds for kS2L-V when $b_l \geq t$ and $b_u - b_l < t$ (resp. $b_u - b_l \geq t$) are only slightly worse. The proofs for the latter cases are omitted due to space restrictions.

2 Lower Bounds

In this section, we present lower bounds for kS2L-F and kS2L-V. We use ALG to denote any deterministic on-line algorithm and OPT to denote an optimal scheduler. The set of requests accepted by ALG is referred to as R', and the set of requests accepted by OPT as R^*.

▶ **Theorem 1.** *For $a \geq t$ (resp. $b_l \geq t, b_u - b_l < t$), no deterministic on-line algorithm for kS2L-F (resp. kS2L-V) can achieve competitive ratio smaller than* 1.5.

Proof. Initially, the adversary releases the 1^{st} request sequence $r_1, r_2, \ldots, r_k$ with $r_1 = r_2 = \cdots = r_k = (\nu \cdot t - a, \nu \cdot t, 1)$, where $\nu \in \mathbb{N}$ and $\nu \cdot t - a \geq 0$ (resp. $r_1 = r_2 = \cdots = r_k = (\nu \cdot t - b_l, \nu \cdot t, 1)$ where $\nu \in \mathbb{N}$ and $\nu \cdot t - b_l \geq 0$). Suppose ALG accepts k_1 ($1 \leq k_1 \leq k$) requests in the 1^{st} request sequence. There are two options that the adversary can adopt:

Option 1: The adversary releases the 2^{nd} request sequence $r_{k+1}, r_{k+2}, \ldots, r_{2k}$ with $r_{k+1} = r_{k+2} = \cdots = r_{2k} = (\tilde{t}_{r_1}, t_{r_1}, 0)$, and the 3^{rd} request sequence $r_{2k+1}, r_{2k+2}, \ldots, r_{3k}$ with $r_{2k+1} = r_{2k+2} = \cdots = r_{3k} = (\tilde{t}_{r_1} + t, t_{r_1} + t, 1)$. Note that the requests in the 2^{nd} and the 3^{rd} request sequences must be assigned to different servers from the k_1 servers that have accepted requests of the 1^{st} request sequence as they are in conflict. From this it follows that ALG cannot accept more than $2(k - k_1)$ requests of the 2^{nd} and the 3^{rd} request sequences. OPT accepts all the requests in the 2^{nd} and the 3^{rd} request sequences. We have $P_{R^*} = 2kr$ and $P_{R'} \leq k_1 r + 2(k - k_1)r = (2k - k_1)r$, and hence $\frac{P_{R^*}}{P_{R'}} \geq \frac{2k}{2k-k_1}$.

Option 2: The adversary does not release any more requests. OPT accepts all requests in the 1^{st} request sequence. We have $P_{R^*} = k \cdot r$ and $P_{R'} = k_1 \cdot r$, and hence $\frac{P_{R^*}}{P_{R'}} \geq \frac{k}{k_1}$.

Algorithm 1 Balanced Greedy Algorithm (BGA).

Input: k servers ($2\theta k$ specified servers and $(1-2\theta)k$ unspecified servers), requests arrive over time.
Step: When request r_i arrives, if it is acceptable to a specified server, assign it to that server; otherwise, if r_i is acceptable to an unspecified server, assign r_i to that server; otherwise, reject it.

If $k_1 \geq \frac{2k}{3}$, $\frac{2k}{2k-k_1} \geq 1.5$; if $k_1 \leq \frac{2k}{3}$, $\frac{k}{k_1} \geq 1.5$. As the adversary can choose the option that maximizes $\frac{P_{R^*}}{P_{R'}}$, the claimed lower bound of 1.5 follows. ◀

▶ **Theorem 2.** *For $b_l \geq t$ and $b_u - b_l \geq t$, no deterministic on-line algorithm for kS2L-V can achieve competitive ratio smaller than* $5/3$*.*

Proof. Initially, the adversary releases the 1^{st} request sequence $r_1, r_2, \ldots, r_k$ with $r_1 = r_2 = \cdots = r_k = (\nu \cdot t - b_u, \nu \cdot t, 0)$ (where $\nu \in \mathbb{N}$ with $\nu \cdot t - b_u \geq 0$). Suppose ALG accepts k_1 ($1 \leq k_1 \leq k$) requests in the 1^{st} request sequence. There are now two options that the adversary can adopt.

Option 1: The adversary releases the 2^{nd} request sequence $r_{k+1}, r_{k+2}, \ldots, r_{2k}$ with $r_{k+1} = r_{k+2} = \cdots = r_{2k} = (\tilde{t}_{r_1}, t_{r_1} - t, 0)$ (note that $t_{r_1} - t - \tilde{t}_{r_1} = \nu \cdot t - t - (\nu \cdot t - b_u) = b_u - t \geq b_l$), and the 3^{rd} request sequence $r_{2k+1}, r_{2k+2}, \ldots, r_{3k}$ with $r_{2k+1} = r_{2k+2} = \cdots = r_{3k} = (\tilde{t}_{r_{2k}} + t, t_{r_{2k}} + t, 1)$, and the 4^{th} request sequence $r_{3k+1}, r_{3k+2}, \ldots, r_{4k}$ with $r_{3k+1} = r_{3k+2} = \cdots = r_{4k} = (\tilde{t}_{r_{2k}} + 2t, t_{r_{2k}} + 2t, 0)$.
Note that the requests in the 2^{nd}, the 3^{rd} and the 4^{th} request sequences must be assigned to different servers from the k_1 servers that have accepted requests of the 1^{st} request sequence as they are in conflict. From this it follows that ALG cannot accept more than $3(k-k_1)$ requests in the 2^{nd}, 3^{rd} and 4^{th} request sequences. OPT accepts all the requests in the 2^{nd}, 3^{rd} and 4^{th} request sequences. We have $P_{R^*} = 3kr$ and $P_{R'} \leq k_1 r + 3(k-k_1)r = (3k-2k_1)r$, and hence $\frac{P_{R^*}}{P_{R'}} \geq \frac{3k}{3k-2k_1}$.

Option 2: The adversary does not release any more requests. OPT accepts all requests in the 1^{st} request sequence. We have $P_{R^*} = k \cdot r$ and $P_{R'} = k_1 \cdot r$, and hence $\frac{P_{R^*}}{P_{R'}} \geq \frac{k}{k_1}$.

If $k_1 \geq \frac{3}{5}k$, $\frac{3k}{3k-2k_1} \geq \frac{5}{3}$; if $k_1 \leq \frac{3}{5}k$, $\frac{k}{k_1} \geq \frac{5}{3}$. As the adversary can choose the option that maximizes $\frac{P_{R^*}}{P_{R'}}$, the claimed lower bound of 5/3 follows. ◀

3 Upper Bounds

We propose a Balanced Greedy Algorithm (BGA) for the kS2L-F/V problem, shown in Algorithm 1. The k servers are divided into two groups: a set S_f of $2\theta k$ *specified* servers and a set S_u of $(1-2\theta)k$ *unspecified* servers, where θ is a parameter satisfying $0 \leq \theta \leq \frac{1}{2}$ and chosen in such a way that θk is an integer. The set S_f is further partitioned into sets S_f^o and S_f^e of θk servers each. The θk specified servers in S_f^o serve only requests that start at location 0 at time νt where ν is even and requests that start at location 1 at time νt where ν is odd. The θk specified servers in S_f^e serve the other request types, i.e., requests that start at location 0 (resp. 1) at time νt where ν is odd (resp. even).

When the algorithm receives request r_i, let $R'(r_i)$ denote the set of requests that BGA has already accepted, and let $R'_j(r_i)$ denote the set of requests that BGA has accepted and that are assigned to s_j, for any j. Request r_i is *acceptable* to a specified server if and only if the number of requests in $R'(r_i)$ that start at t_{r_i} and have pick-up location p_{r_i} is less than

θk. Furthermore, r_i is *acceptable* to an unspecified server s_j ($s_j \in S_u$) if and only if r_i is not in conflict with the requests in $R'_j(r_i)$, i.e., for all $r'_q \in R'_j(r_i)$ we have $|t_{r_i} - t_{r'_q}| \geq 2t$ if $p_{r_i} = p_{r'_q}$ and $|t_{r_i} - t_{r'_q}| \geq t$ if $p_{r_i} \neq p_{r'_q}$.

Denote the requests accepted by OPT by $R^* = \{r^*_1, r^*_2, \ldots, r^*_{|R^*|}\}$ and the requests accepted by BGA by $R' = \{r'_1, r'_2, \ldots, r'_{|R'|}\}$ indexed in order of non-decreasing start times. The requests with equal start time are ordered in the order in which they arrive. Let $R^*(d)$ denote the set of requests in R^* which start at time d, and let $R^*(d, e)$ denote the set of requests in R^* which start at time d and have pick-up location e. Observe that for all d, e, we have $|R^*(d)| \leq k$ and $|R^*(d, e)| \leq k$. Let $R'(d)$ denote the set of requests in R' which start at time d, and let $R'(d, e)$ denote the set of requests in R' which start at time d and have pick-up location e. Observe that for all d, e, we have $|R'(d)| \leq k$ and $|R'(d, e)| \leq (1 - \theta)k$.

For simplification of the analysis, we suppose that the specified servers in each of the sets S^o_f and S^e_f are ordered and if a request r_i is acceptable to some specified server, BGA assigns r_i to the available specified server that comes first in that order.

▶ **Observation 3.** If $\theta k > 0$, then $\forall r^*_i \in R^*$: $t_{r'_1} \leq t_{r^*_i} \leq t_{r'_{|R'|}}$.

▶ **Observation 4.** For every $r^*_i \in R^*$, BGA accepts $\min\{|R^*(t_{r^*_i}, p_{r^*_i})|, \theta k\}$ requests that start at $t_{r^*_i}$ and have pick-up location $p_{r^*_i}$ with specified servers, and hence $|R'(t_{r^*_i}, p_{r^*_i})| \geq \min\{|R^*(t_{r^*_i}, p_{r^*_i})|, \theta k\}$.

▶ **Observation 5.** If k_0 servers of OPT, where $0 \leq k_0 \leq k$, each accept y ($y \geq 1$) requests that start during period $[x, x + yt)$ (where $x = \nu t$ for some $\nu \in \mathbb{N}$), then at least $\min\{\theta k, k_0\}$ specified servers of BGA each accept y requests that start during this period.

To illustrate the idea of our analysis of BGA, we first give a simple proof of an upper bound of 2 on the competitive ratio of *BGA*.

▶ **Theorem 6.** *With $\theta = \frac{1}{2}$, BGA is 2-competitive for kS2L-F and kS2L-V if k is even.*

Proof. Since BGA with $\theta = \frac{1}{2}$ accepts a request $r_i \in R$ if the number of requests in $R'(r_i)$ that start at t_{r_i} and have pick-up location p_{r_i} is less than $\frac{k}{2}$, BGA can always accept $\min\{\frac{k}{2}, |R^*(t_{r^*_i}, p_{r^*_i})|\}$ requests that start at the same time and have the same pick-up location. As OPT accepts at most k requests that start at the same time and have the same pick-up location, i.e., $|R^*(t_{r^*_i}, p_{r^*_i})| \leq k$, it follows that $|R'| \geq \frac{1}{2}|R^*|$. ◀

▶ **Definition 7** (Common and uncommon request). For each $r^*_i \in R^*$, if the number of requests in R^* that start at $t_{r^*_i}$ and have pick-up location $p_{r^*_i}$ is no more than the number of requests in R' which start at $t_{r^*_i}$ and have pick-up location $p_{r^*_i}$, i.e., $|R^*(t_{r^*_i}, p_{r^*_i})| \leq |R'(t_{r^*_i}, p_{r^*_i})|$, we say that the requests in $R^*(t_{r^*_i}, p_{r^*_i})$ are *common*; if the number of requests in R^* which start at $t_{r^*_i}$ and have pick-up location $p_{r^*_i}$ is greater than the number of requests in R' which start at $t_{r^*_i}$ and have pick-up location $p_{r^*_i}$, i.e., $|R^*(t_{r^*_i}, p_{r^*_i})| > |R'(t_{r^*_i}, p_{r^*_i})|$, we say that the first $|R'(t_{r^*_i}, p_{r^*_i})|$ requests in $R^*(t_{r^*_i}, p_{r^*_i})$ are common, and the remaining requests in $R^*(t_{r^*_i}, p_{r^*_i})$ are *uncommon*.

▶ **Observation 8.** If $r^*_i \in R^*$ is uncommon, $|R'(t_{r^*_i}, p_{r^*_i})| \geq \theta k$.

▶ **Definition 9** (Sufficient and insufficient interval). We say that an interval $[x, x + t)$ (x is an integer multiple of t) is *sufficient* if $|R'(x)| \geq (1 - \theta)|R^*(x)|$; otherwise it is *insufficient*.

3.1 Upper Bounds for kS2L-F

▶ **Observation 10.** For kS2L-F, if interval $[x, x+t)$ (x is an integer multiple of t) is insufficient, then $x \geq t_{r_1} + t$.

With the following two lemmas, we show that if an interval I is insufficient, the interval I' preceding it must be sufficient and the competitive ratio of BGA with respect to requests starting in I and I' is at most 1.5 (for $\theta = \frac{1}{3}$).

▶ **Lemma 11.** *For $\frac{1}{3} \leq \theta \leq \frac{1}{2}$, if $r_i^* \in R^*$ is uncommon and interval $[t_{r_i^*}, t_{r_i^*} + t)$ is insufficient, then $|R'(t_{r_i^*} - t, p_{r_i^*})| = k - |R'(t_{r_i^*}, p_{r_i^*})|$*

Proof. As r_i^* is uncommon, $|R'(t_{r_i^*}, p_{r_i^*})| \geq \theta k$ (by Observation 8) and every unspecified server has accepted a request that is in conflict with r_i^*, i.e., for all $s_j \in S_u$, there is $r'_q \in R'_j(r_i^*)$ (recall that $R'_j(r_i^*)$ is the set of requests that are accepted and assigned to s_j by BGA at the time when r_i^* is released) such that $t_{r_i^*} = t_{r'_q}$ and $p_{r_i^*} = p_{r'_q}$, or $t_{r_i^*} - t_{r'_q} = t$ and $p_{r_i^*} = p_{r'_q}$, or $t_{r_i^*} = t_{r'_q}$ and $p_{r_i^*} \neq p_{r'_q}$.

Observe that $|R'(t_{r_i^*})| < (1-\theta)k$ because interval $[t_{r_i^*}, t_{r_i^*} + t)$ is insufficient. Since $|R'(t_{r_i^*}, p_{r_i^*})| \geq \theta k$, $|R'(t_{r_i^*}, \dot{p}_{r_i^*})| < (1-\theta)k - \theta k \leq \theta k$ (as $\frac{1}{3} \leq \theta$). This implies that BGA does not use unspecified servers to serve requests in $R'(t_{r_i^*}, \dot{p}_{r_i^*})$ because BGA does not use unspecified servers when specified servers are available. From this it follows that each of the unspecified servers either accepts a request that starts at $t_{r_i^*}$ with pick-up location $p_{r_i^*}$, or accepts a request that starts at $t_{r_i^*} - t$ with pick-up location $p_{r_i^*}$. As $|R'(t_{r_i^*}, p_{r_i^*})| < (1-\theta)k = \theta k + (1-2\theta)k$, at least one unspecified server accepts a request that starts at $t_{r_i^*} - t$ with pick-up location $p_{r_i^*}$. This implies that $|R'(t_{r_i^*} - t, p_{r_i^*})| \geq \theta k$. Since $|R'(t_{r_i^*}, p_{r_i^*})| \geq \theta k$, each of the specified servers either accepts a request that starts at $t_{r_i^*}$ at $p_{r_i^*}$ or a request that starts at $t_{r_i^*} - t$ at $p_{r_i^*}$. Therefore $|R'(t_{r_i^*} - t, p_{r_i^*})| = k - |R'(t_{r_i^*}, p_{r_i^*})|$. ◀

▶ **Lemma 12.** *For $\theta = \frac{1}{3}$, if $r_i^* \in R^*$ is uncommon and interval $[t_{r_i^*}, t_{r_i^*} + t)$ is insufficient, then $|R'(t_{r_i^*} - t)| + |R'(t_{r_i^*})| \geq \frac{2}{3}(|R^*(t_{r_i^*} - t)| + |R^*(t_{r_i^*})|)$ and also $|R'(t_{r_i^*} - t)| > \frac{2}{3}|R^*(t_{r_i^*} - t)|$.*

Proof. According to Lemma 11, $|R'(t_{r_i^*} - t, p_{r_i^*})| = k - |R'(t_{r_i^*}, p_{r_i^*})|$. From this it follows that each server of BGA accepts at least one request that starts during period $[t_{r_i^*} - t, t_{r_i^*}]$, i.e., $|R'(t_{r_i^*} - t)| + |R'(t_{r_i^*})| \geq k$. Suppose k_0 servers of *OPT* each accept two requests that start during period $[t_{r_i^*} - t, t_{r_i^*}]$. We distinguish two cases.

Case 1: $k_0 \geq \frac{k}{3}$. By Observation 5 (with $y = 2$), at least θk servers of BGA accept two requests that start during period $[t_{r_i^*} - t, t_{r_i^*}]$. Since each server accepts at least one request that starts during period $[t_{r_i^*} - t, t_{r_i^*}]$, $|R'(t_{r_i^*} - t)| + |R'(t_{r_i^*})| \geq 2\theta k + (1-\theta)k = \frac{4}{3}k$. Since $|R^*(t_{r_i^*} - t)| + |R^*(t_{r_i^*})| \leq 2k$ (each server of *OPT* accepts at most two requests that start during period $[t_{r_i^*} - t, t_{r_i^*}]$), we have $\frac{|R^*(t_{r_i^*} - t)| + |R^*(t_{r_i^*})|}{|R'(t_{r_i^*} - t)| + |R'(t_{r_i^*})|} \leq \frac{2k}{\frac{4}{3}k} = \frac{3}{2}$.

Case 2: $k_0 < \frac{k}{3}$. Note that each server of *OPT* accepts at most two requests that start during period $[t_{r_i^*} - t, t_{r_i^*}]$, so $|R^*(t_{r_i^*} - t)| + |R^*(t_{r_i^*})| < \frac{2k}{3} + (k - \frac{k}{3}) = \frac{4}{3}k$. Since $|R'(t_{r_i^*} - t)| + |R'(t_{r_i^*})| \geq k$, we have $\frac{|R^*(t_{r_i^*} - t)| + |R^*(t_{r_i^*})|}{|R'(t_{r_i^*} - t)| + |R'(t_{r_i^*})|} \leq \frac{\frac{4}{3}k}{k} < \frac{3}{2}$.

Because $|R'(t_{r_i^*})| < \frac{2}{3}|R^*(t_{r_i^*})|$ and $|R'(t_{r_i^*} - t)| + |R'(t_{r_i^*})| \geq \frac{2}{3}(|R^*(t_{r_i^*} - t)| + |R^*(t_{r_i^*})|)$, we have $|R'(t_{r_i^*} - t)| > \frac{2}{3}|R^*(t_{r_i^*} - t)|$. ◀

▶ **Corollary 13.** *If interval $[x, x+t)$ (x is an integer multiple of t) is insufficient, then interval $[x - t, x)$ and interval $[x + t, x + 2t)$ are sufficient.*

Algorithm 2 Partition Rule (for kS2L-F).

Initialization: $\gamma = \frac{t_{r'_{|R'|}} - t_{r'_1}}{t}$, $j = 2$, $l_j = 0$, $i = 0$.
while $i \leq \gamma$ do
  if interval i and $i+1$ are sufficient then
    $j = j+1$, $i = i+1$, $l_j = i$;
  else if interval i is sufficient and interval $i+1$ is insufficient then
    $j = j+1$, $i = i+2$, $l_j = i$;
$\gamma' = j$.

▶ **Theorem 14.** *With $\theta = \frac{1}{3}$, BGA is $\frac{3}{2}$-competitive for kS2L-F if $k = 3\nu$ ($\nu \in \mathbb{N}$).*

Proof. We partition the time horizon $[0, \infty)$ into γ' ($\gamma' \leq \gamma + 3$, where $\gamma = \frac{t_{r'_{|R'|}} - t_{r'_1}}{t}$) periods that can be analyzed independently. Let interval i ($0 \leq i \leq \gamma$) denote interval $[t_{r'_1} + it, t_{r'_1} + (i+1)t)$. We partition the time horizon based on the Partition rule (Algorithm 2) and let period j ($1 < j < \gamma'$) denote $[t_{r'_1} + l_j \cdot t, t_{r'_1} + l_{j+1} \cdot t)$, in such a way that BGA and OPT do not accept any requests in the first period $[0, t_{r'_1})$ and the last period $[t_{r'_1} + l_{\gamma'} \cdot t, \infty)$, and the length of each period j ($1 < j < \gamma'$), i.e., $(l_{j+1} - l_j)t$, is either t or $2t$. By Corollary 13 and the Partition rule (Algorithm 2), we have the following properties: if the length of period j is t, i.e., $l_{j+1} - l_j = 1$, period j is sufficient; if the length of period j is $2t$, i.e., $l_{j+1} - l_j = 2$, the first half of period j, i.e., $[t_{r'_1} + l_j \cdot t, t_{r'_1} + (l_j + 1) \cdot t)$, is sufficient and the second half of period j, i.e., $[t_{r'_1} + (l_j + 1) \cdot t, t_{r'_1} + (l_j + 2) \cdot t)$, is insufficient. Recall that interval 0 is always sufficient by Observation 10.

Let $R^*_{(j)}$ denote the set of requests accepted by OPT that start in period j, for $1 \leq j \leq \gamma'$. Let $R'_{(j)}$ denote the set of requests accepted by BGA that start in period j, for $1 \leq j \leq \gamma'$. We bound the competitive ratio of BGA by analyzing each period independently. As $R' = \bigcup_j R'_{(j)}$ and $R^* = \bigcup_j R^*_{(j)}$, it is clear for any $\alpha \geq 1$ that $P_{R^*}/P_{R'} \leq \alpha$ follows if we can show that $P_{R^*_{(j)}}/P_{R'_{(j)}} \leq \alpha$ for all j, $1 \leq j \leq \gamma'$.

According to Lemma 12, when $l_{j+1} - l_j = 2$, $P_{R^*_{(j)}}/P_{R'_{(j)}} \leq \frac{3}{2}$. Based on the partition rule, when $l_{j+1} - l_j = 1$, period j is sufficient, i.e., $|R'_{(j)}| \geq (1-\theta)|R^*_{(j)}|$ and hence $P_{R^*_{(j)}}/P_{R'_{(j)}} \leq \frac{3}{2}$. Since $P_{R^*_{(j)}} = P_{R'_{(j)}} = 0$ for $j = 1$ and $j = \gamma'$ (recall that by Observation 3, all requests accepted by BGA and OPT do not start earlier than $t_{r'_1}$ and do not start later than $t_{r'_{|R'|}}$), the theorem follows. ◀

3.2 Upper Bounds for kS2L-V

If $b_l \geq t$ and $b_u - b_l < t$ for the kS2L-V problem, let $\theta = \frac{1}{3}$. Since each request starts at time νt for some $\nu \in \mathbb{N}$, all requests start in order of their release times, and therefore the upper bound for the kS2L-V problem is equal to the upper bound for the kS2L-F problem (with $a \geq t$). From now on consider the kS2L-V problem with $b_l \geq t$ and $b_u - b_l \geq t$, and let $\theta = \frac{2}{5}$.

▶ **Lemma 15.** *For $\theta = \frac{2}{5}$, if $r^*_i \in R^*$ is uncommon and interval $[t_{r^*_i}, t_{r^*_i} + t)$ is insufficient, then one of the following holds:*

(i) $|R'(t_{r^*_i} - t, p_{r^*_i})| = k - |R'(t_{r^*_i}, p_{r^*_i})|$ $(> \frac{2}{5}k)$ *and* $|R'(t_{r^*_i} + t, p_{r^*_i})| \leq \theta k$, *or*

(ii) $|R'(t_{r^*_i} + t, p_{r^*_i})| = k - |R'(t_{r^*_i}, p_{r^*_i})|$ $(> \frac{2}{5}k)$ *and* $|R'(t_{r^*_i} - t, p_{r^*_i})| \leq \theta k$, *or*

(iii) $|R'(t_{r^*_i} - t, p_{r^*_i})| - \theta k + |R'(t_{r^*_i} + t, p_{r^*_i})| - \theta k + |R'(t_{r^*_i}, p_{r^*_i})| - \theta k \geq (1 - 2\theta)k$ *and* $|R'(t_{r^*_i} - t, p_{r^*_i})| > \theta k$ *and* $|R'(t_{r^*_i} + t, p_{r^*_i})| > \theta k$.

Proof. As r_i^* is uncommon, $|R'(t_{r_i^*}, p_{r_i^*})| \geq \theta k$ (by Observation 8) and every unspecified server has accepted a request that is in conflict with r_i^*, i.e., for every $s_j \in S_u$ there exists $r'_q \in R'_j(r_i^*)$ (recall that $R'_j(r_i^*)$ is the set of requests that are accepted and assigned to s_j by BGA at the time when r_i^* arrives) such that $t_{r'_q} = t_{r_i^*}$ and $p_{r_i^*} = p_{r'_q}$, or $t_{r'_q} = t_{r_i^*} - t$ and $p_{r_i^*} = p_{r'_q}$, or $t_{r'_q} = t_{r_i^*} + t$ and $p_{r_i^*} = p_{r'_q}$, or $t_{r'_q} = t_{r_i^*}$ and $p_{r_i^*} \neq p_{r'_q}$.

Observe that $|R'(t_{r_i^*})| < (1-\theta)k$ because interval $[t_{r_i^*}, t_{r_i^*} + t)$ is insufficient. Since $|R'(t_{r_i^*}, p_{r_i^*})| \geq \theta k$, $|R'(t_{r_i^*}, \dot{p}_{r_i^*})| < (1-\theta)k - \theta k \leq \theta k$ (as $\theta = \frac{2}{5}$). This implies that BGA does not use unspecified servers to serve requests in $R'(t_{r_i^*}, \dot{p}_{r_i^*})$ because BGA does not use unspecified servers when specified servers are available. From this it follows that each of the unspecified servers either accepts a request that starts at $t_{r_i^*}$ with pick-up location $p_{r_i^*}$, or accepts a request that starts at $t_{r_i^*} - t$ with pick-up location $p_{r_i^*}$, or accepts a request that starts at $t_{r_i^*} + t$ with pick-up location $p_{r_i^*}$. As $|R'(t_{r_i^*}, p_{r_i^*})| < (1-\theta)k = \theta k + (1-2\theta)k$, at least one unspecified server accepts a request that starts at $t_{r_i^*} - t$ at $p_{r_i^*}$, or a request that starts at $t_{r_i^*} + t$ at $p_{r_i^*}$. We distinguish three cases.

Case 1: No unspecified server accepts a request that starts at $t_{r_i^*} + t$ at $p_{r_i^*}$. Then at least one unspecified server accepts a request that starts at $t_{r_i^*} - t$ at $p_{r_i^*}$. This implies that $|R'(t_{r_i^*} - t, p_{r_i^*})| \geq \theta k$. Since $|R'(t_{r_i^*}, p_{r_i^*})| \geq \theta k$, each of the specified servers either accepts a request that starts at $t_{r_i^*}$ at $p_{r_i^*}$ or a request that starts at $t_{r_i^*} - t$ at $p_{r_i^*}$. Each of the unspecified servers either accepts a request that starts at $t_{r_i^*}$ at $p_{r_i^*}$, or a request that starts at $t_{r_i^*} - t$ at $p_{r_i^*}$. Therefore, $|R'(t_{r_i^*} - t, p_{r_i^*})| = k - |R'(t_{r_i^*}, p_{r_i^*})|$, and (i) holds.

Case 2: No unspecified server accepts a request that starts at $t_{r_i^*} - t$ at $p_{r_i^*}$. By symmetric arguments to Case 1, we get $|R'(t_{r_i^*} + t, p_{r_i^*})| = k - |R'(t_{r_i^*}, p_{r_i^*})|$, and (ii) holds.

Case 3: At least one unspecified server accepts a request that starts at $t_{r_i^*} + t$ at $p_{r_i^*}$, and at least one unspecified server accepts a request that starts at $t_{r_i^*} - t$ at $p_{r_i^*}$. This implies that $|R'(t_{r_i^*} - t, p_{r_i^*})| \geq \theta k$ and $|R'(t_{r_i^*} + t, p_{r_i^*})| \geq \theta k$. Since each of the unspecified servers either accepts a request that starts at $t_{r_i^*}$ at $p_{r_i^*}$, or a request that starts at $t_{r_i^*} - t$ at $p_{r_i^*}$, or a request that starts at $t_{r_i^*} + t$ at $p_{r_i^*}$, we have that $|R'(t_{r_i^*} - t, p_{r_i^*})| - \theta k + |R'(t_{r_i^*} + t, p_{r_i^*})| - \theta k + |R'(t_{r_i^*}, p_{r_i^*})| - \theta k \geq (1-2\theta)k$, and (iii) holds. ◀

▶ **Definition 16** (l-full and r-full, l-large and r-large, l-small and r-small)**.** If r_i^* is an uncommon request such that the interval $I = [t_{r_i^*}, t_{r_i^*} + t)$ is insufficient, we say that the interval $[t_{r_i^*} - t, t_{r_i^*})$ (resp. $[t_{r_i^*} + t, t_{r_i^*} + 2t)$) is *l-full* (resp. *r-full*) with respect to I if $|R'(t_{r_i^*} - t, p_{r_i^*})| = k - |R'(t_{r_i^*}, p_{r_i^*})|$ (resp. if $|R'(t_{r_i^*} + t, p_{r_i^*})| = k - |R'(t_{r_i^*}, p_{r_i^*})|$); we say that the interval $[t_{r_i^*} - t, t_{r_i^*})$ (resp. $[t_{r_i^*} + t, t_{r_i^*} + 2t)$) is *l-large* (resp. *r-large*) with respect to I if $\frac{2}{5}k < |R'(t_{r_i^*} - t, p_{r_i^*})| < k - |R'(t_{r_i^*}, p_{r_i^*})|$ (resp. $\frac{2}{5}k < |R'(t_{r_i^*} + t, p_{r_i^*})| < k - |R'(t_{r_i^*}, p_{r_i^*})|$); and we say that the interval $[t_{r_i^*} - t, t_{r_i^*})$ (resp. $[t_{r_i^*} + t, t_{r_i^*} + 2t)$) is *l-small* (resp. *r-small*) with respect to I if $|R'(t_{r_i^*} - t, p_{r_i^*})| \leq \frac{2}{5}k$ (resp. $|R'(t_{r_i^*} + t, p_{r_i^*})| \leq \frac{2}{5}k$).

Note that the properties l-full, l-large and l-small refer to the interval directly to the *left* of an insufficient interval, and the properties r-full, r-large and r-small to the interval directly to the *right* of an insufficient interval.

▶ **Observation 17** (Uniqueness)**.** If r_i^* is uncommon and interval $[t_{r_i^*}, t_{r_i^*} + t)$ is insufficient, then interval $[t_{r_i^*} - t, t_{r_i^*})$ is either l-full, l-large, or l-small, and interval $[t_{r_i^*} + t, t_{r_i^*} + 2t)$ is either r-full, r-large, or r-small.

By Lemma 15, we obtain:

▶ **Observation 18.** If r_i^* is uncommon, interval $[t_{r_i^*}, t_{r_i^*} + t)$ is insufficient, and interval $[t_{r_i^*} + t, t_{r_i^*} + 2t)$ is r-small, then interval $[t_{r_i^*} - t, t_{r_i^*})$ is l-full. Similarly, if r_i^* is uncommon, interval $[t_{r_i^*}, t_{r_i^*} + t)$ is insufficient, and interval $[t_{r_i^*} - t, t_{r_i^*})$ is l-small, then interval $[t_{r_i^*} + t, t_{r_i^*})$ is r-full.

▶ **Lemma 19.** *For $\theta = \frac{2}{5}k$, if $r_i^* \in R^*$ is uncommon, interval $[t_{r_i^*}, t_{r_i^*} + t)$ is insufficient and $|R'(t_{r_i^*} - t, p_{r_i^*})| > \frac{2}{5}k$ (i.e., interval $[t_{r_i^*} - t, t_{r_i^*})$ is l-large or l-full), then $|R'(t_{r_i^*} - t)| + |R'(t_{r_i^*})| \geq \frac{3}{5}(|R^*(t_{r_i^*} - t)| + |R^*(t_{r_i^*})|)$ and interval $[t_{r_i^*} - t, t_{r_i^*})$ is sufficient. Similarly, if r_i^* is uncommon, interval $[t_{r_i^*}, t_{r_i^*} + t)$ is insufficient and $|R'(t_{r_i^*} + t, p_{r_i^*})| > \frac{2}{5}k$ (i.e., interval $[t_{r_i^*} + t, t_{r_i^*} + 2t)$ is r-large or r-full), then $|R'(t_{r_i^*} + t)| + |R'(t_{r_i^*})| \geq \frac{3}{5}(|R^*(t_{r_i^*} + t)| + |R^*(t_{r_i^*})|$ and interval $[t_{r_i^*} + t, t_{r_i^*} + 2t)$ is sufficient.*

Proof. Observe that $|R'(t_{r_i^*}, p_{r_i^*})| \geq \frac{2}{5}k$ because r_i^* is uncommon. Since $|R'(t_{r_i^*} - t, p_{r_i^*})| \geq \frac{2}{5}k$ (resp. $|R'(t_{r_i^*} + t, p_{r_i^*})| \geq \frac{2}{5}k$), each specified server of BGA accepts at least one request that starts during period $[t_{r_i^*} - t, t_{r_i^*}]$ (resp. $[t_{r_i^*}, t_{r_i^*} + t]$). Suppose k_0 servers of OPT each accept two requests that start during period $[t_{r_i^*} - t, t_{r_i^*}]$ (resp. $[t_{r_i^*}, t_{r_i^*} + t]$). We distinguish two cases.

Case 1: $k_0 \geq \frac{2}{5}k$. By Observation 5 (with $y = 2$), at least $\frac{2}{5}k$ servers of BGA each accept two requests that start during period $[t_{r_i^*} - t, t_{r_i^*}]$ (resp. $[t_{r_i^*}, t_{r_i^*} + t]$). Since each specified server of BGA accepts at least one request that starts during period $[t_{r_i^*} - t, t_{r_i^*}]$ (resp. $[t_{r_i^*}, t_{r_i^*} + t]$), $|R'(t_{r_i^*} - t)| + |R'(t_{r_i^*})| \geq 2 \cdot \frac{2}{5}k + \frac{2}{5}k \geq \frac{6}{5}k$ (resp. $|R'(t_{r_i^*} + t)| + |R'(t_{r_i^*})| \geq \frac{6}{5}k$). Since $|R^*(t_{r_i^*} - t)| + |R^*(t_{r_i^*})| \leq 2k$ and $|R^*(t_{r_i^*} + t)| + |R^*(t_{r_i^*})| \leq 2k$ (each server of OPT accepts at most two requests that start during period $[t_{r_i^*} - t, t_{r_i^*}]$ or period $[t_{r_i^*} + t, t_{r_i^*}]$), we have $\frac{|R^*(t_{r_i^*} - t)| + |R^*(t_{r_i^*})|}{|R'(t_{r_i^*} - t)| + |R'(t_{r_i^*})|} \leq \frac{2k}{\frac{6}{5}k} = \frac{5}{3}$ (resp. $\frac{|R^*(t_{r_i^*} + t)| + |R^*(t_{r_i^*})|}{|R'(t_{r_i^*} + t)| + |R'(t_{r_i^*})|} \leq \frac{5}{3}$).

Case 2: $k_0 < \frac{2}{5}k$. According to the use of specified servers by BGA, at least k_0 servers of BGA each accept two requests that start during period $[t_{r_i^*} - t, t_{r_i^*}]$ (resp. $[t_{r_i^*}, t_{r_i^*} + t]$). Since each specified server of BGA accepts at least one request that starts during period $[t_{r_i^*} - t, t_{r_i^*}]$ (resp. $[t_{r_i^*}, t_{r_i^*} + t]$), $|R'(t_{r_i^*} - t)| + |R'(t_{r_i^*})| \geq 2k_0 + \frac{4}{5}k - k_0 \geq \frac{4}{5}k + k_0$ (resp. $|R'(t_{r_i^*} + t)| + |R'(t_{r_i^*})| \geq \frac{4}{5}k + k_0$). Since $|R^*(t_{r_i^*} - t)| + |R^*(t_{r_i^*})| \leq 2k_0 + k - k_0 = k + k_0$ and $|R^*(t_{r_i^*} + t)| + |R^*(t_{r_i^*})| \leq k + k_0$, we have $\frac{|R^*(t_{r_i^*} - t)| + |R^*(t_{r_i^*})|}{|R'(t_{r_i^*} - t)| + |R'(t_{r_i^*})|} \leq \frac{k + k_0}{\frac{4}{5}k + k_0} \leq \frac{5}{4} < \frac{5}{3}$ (resp. $\frac{|R^*(t_{r_i^*} + t)| + |R^*(t_{r_i^*})|}{|R'(t_{r_i^*} + t)| + |R'(t_{r_i^*})|} < \frac{5}{3}$).

Observe that $|R'(t_{r_i^*})| < \frac{3}{5}|R^*(t_{r_i^*})|$ because interval $[t_{r_i^*}, t_{r_i^*} + t)$ is insufficient. If $|R'(t_{r_i^*} - t, p_{r_i^*})| > \frac{2}{5}k$, then $|R'(t_{r_i^*} - t)| + |R'(t_{r_i^*})| \geq \frac{3}{5}(|R^*(t_{r_i^*} - t)| + |R^*(t_{r_i^*})|)$ implies $|R'(t_{r_i^*} - t)| \geq \frac{3}{5}|R^*(t_{r_i^*} - t)|$. Similarly, if $|R'(t_{r_i^*} + t, p_{r_i^*})| > \frac{2}{5}k$, then $|R'(t_{r_i^*} + t)| \geq \frac{3}{5}|R^*(t_{r_i^*} + t)|$. ◀

▶ **Lemma 20.** *For $\theta = \frac{2}{5}k$, consider any $r_i^*, r_j^* \in R^*$ where $t_{r_j^*} = t_{r_i^*} + 2t$, r_i^* and r_j^* are uncommon, intervals $[t_{r_i^*}, t_{r_i^*} + t)$ and $[t_{r_j^*}, t_{r_j^*} + t)$ are insufficient, and interval $[t_{r_i^*} + t, t_{r_j^*})$ is r-full ($|R'(t_{r_i^*} + t, p_{r_i^*})| = k - |R'(t_{r_i^*}, p_{r_i^*})|$). Then $|R'(t_{r_i^*})| + |R'(t_{r_i^*} + t)| + |R'(t_{r_i^*} + 2t)| \geq \frac{3}{5}(|R^*(t_{r_i^*})| + |R^*(t_{r_i^*} + t)| + |R^*(t_{r_i^*} + 2t)|)$.*

Proof. Observe that $|R'(t_{r_i^*}, p_{r_i^*})| \geq \frac{2}{5}k$ (as r_i^* is uncommon), $|R'(t_{r_i^*} + t, p_{r_i^*})| \geq \frac{2}{5}k$ (as interval $[t_{r_i^*} + t, t_{r_i^*})$ is r-full) and $|R'(t_{r_j^*}, p_{r_j^*})| \geq \frac{2}{5}k$ (as r_j^* is uncommon). From this it follows that at least $\frac{2}{5}k$ specified servers of BGA each at least accept two requests that start during period $[t_{r_i^*}, t_{r_j^*}]$. Since $|R'(t_{r_i^*} + t, p_{r_i^*})| = 1 - |R'(t_{r_i^*}, p_{r_i^*})|$, each server of BGA at least accepts one request that starts during period $[t_{r_i^*}, t_{r_j^*}]$. Suppose k_0 servers of OPT each accept three requests that start during period $[t_{r_i^*}, t_{r_j^*}]$. We distinguish two cases.

Case 1: $k_0 \geq \frac{2}{5}k$. By Observation 5 (with $y = 3$), at least $\frac{2}{5}k$ servers of BGA each accept three requests that start during period $[t_{r_i^*}, t_{r_j^*}]$. Since each server of BGA accepts at least one request that starts during period $[t_{r_i^*}, t_{r_j^*}]$, $|R'(t_{r_i^*})| + |R'(t_{r_i^*} + t)| + |R'(t_{r_i^*} + 2t)| \geq 3 \cdot \frac{2}{5}k + \frac{3}{5}k \geq \frac{9}{5}k$. Since $|R^*(t_{r_i^*})| + |R^*(t_{r_i^*} + t)| + |R^*(t_{r_i^*} + 2t)| \leq 3k$, we have $|R'(t_{r_i^*})| + |R'(t_{r_i^*} + t)| + |R'(t_{r_i^*} + 2t)| \geq \frac{3}{5}(|R^*(t_{r_i^*})| + |R^*(t_{r_i^*} + t)| + |R^*(t_{r_i^*} + 2t)|)$.

Algorithm 3 Partition Rule (for kS2L-V).

Initialization: $\gamma = \frac{t_{r'_{|R'|}} - t_{r'_1}}{t}$, $j = 2$, $l_j = 0$, $i = 0$.
while $i \leq \gamma$ do
  if interval i and interval $i+1$ are Suf, then
    $j = j+1$, $i = i+1$, $l_j = i$;
  else if interval i is Suf and l-small, and interval $i+1$ is InSuf, then
    $j = j+1$, $i = i+1$, $l_j = i$;
  else if interval i is Suf and not l-small, and interval $i+1$ is InSuf, then
    $j = j+1$, $i = i+2$, $l_j = i$;
  else if interval i is InSuf, interval $i+1$ is r-full and interval $i+2$ is InSuf, then
    $j = j+1$, $i = i+3$, $l_j = i$;
  else if interval i is InSuf, interval $i+1$ is r-full and interval $i+2$ is Suf, then
    $j = j+1$, $i = i+2$, $l_j = i$;
$\gamma' = j$.

Case 2: $k_0 < \frac{2}{5}k$. By Observation 5 (with $y = 3$), at least k_0 servers of BGA each accept three requests that start during period $[t_{r_i^*}, t_{r_j^*}]$. Since each server of BGA accepts at least one request that starts during period $[t_{r_i^*}, t_{r_j^*}]$ and at least $\frac{2}{5}k$ specified servers of BGA each accept at least two requests that start during period $[t_{r_i^*}, t_{r_j^*}]$, $|R'(t_{r_i^*})| + |R'(t_{r_i^*} + t)| + |R'(t_{r_i^*} + 2t)| \geq 3k_0 + 2 \cdot (\frac{2}{5}k - k_0) + \frac{3}{5}k \geq \frac{7}{5}k + k_0$. Since $|R^*(t_{r_i^*})| + |R^*(t_{r_i^*} + t)| + |R^*(t_{r_i^*} + 2t)| \leq 3k_0 + 2(k - k_0) = 2k + k_0$, we have $\frac{|R^*(t_{r_i^*})|+|R^*(t_{r_i^*}+t)|+|R^*(t_{r_i^*}+2t)|}{|R'(t_{r_i^*})|+|R'(t_{r_i^*}+t)|+|R'(t_{r_i^*}+2t)|} \leq \frac{2k+k_0}{\frac{7}{5}k+k_0} \leq \frac{10}{7} < \frac{5}{3}$. ◀

▶ **Theorem 21.** *With $\theta = \frac{2}{5}$, BGA is $\frac{5}{3}$-competitive for kS2L-V if $k = 5\nu$ ($\nu \in \mathbb{N}$).*

Proof. We partition the time horizon $[0, \infty)$ into γ' ($\gamma' \leq \gamma + 3, \gamma = \frac{t_{r'_{|R'|}} - t_{r'_1}}{t}$) periods that can be analyzed independently. Let interval i ($0 \leq i \leq \gamma$) denote the interval $[t_{r'_1} + it, t_{r'_1} + (i+1)t)$. We partition the time horizon using the partition rule shown in Algorithm 3, where we use InSuf as an abbreviation for insufficient and Suf as an abbreviation for sufficient. We let period j ($1 < j < \gamma'$) denote $[t_{r'_1} + l_j \cdot t, t_{r'_1} + l_{j+1} \cdot t)$.

Observe that BGA and OPT do not accept any requests in the first period $[0, t_{r'_1})$ and in the last period $[t_{r'_1} + l_{\gamma'}t, \infty)$, and that the length of each period j ($1 < j < \gamma'$), i.e., $(l_{j+1} - l_j)t$, is either t, $2t$ or $3t$. By the partition rule (Algorithm 3), we have the following properties: if the length of period j is t, i.e., $l_{j+1} - l_j = 1$, period j is sufficient; if the length of period j is $2t$, i.e., $l_{j+1} - l_j = 2$, either interval i is l-large or l-full and interval $i+1$ is insufficient, or interval i is insufficient and interval $i+1$ is r-full; if the length of period j is $3t$, i.e., $l_{j+1} - l_j = 3$, interval i and interval $i+2$ are insufficient and interval $i+1$ is r-full.

Let $R^*_{(j)}$ denote the set of requests accepted by OPT that start in period j, for $1 \leq j \leq \gamma'$. Let $R'_{(j)}$ denote the set of requests accepted by BGA that start in period j, for $1 \leq j \leq \gamma'$. By Observation 18, if interval i is insufficient and interval $i-1$ is l-small, then interval $i+1$ is r-full. By Lemma 15 and Lemma 19, if interval i is insufficient and interval $i+1$ is insufficient, then interval $i-1$ is l-full and interval $i+2$ is r-full. From this it follows that an invariant of Algorithm 3 is that at the start of each iteration of the while-loop, either interval i is sufficient, or interval i is insufficient and interval $i+1$ is r-full. Hence, the partition rule (Algorithm 3) is complete, i.e., in each iteration of the while-loop one of the if-cases applies.

We bound the competitive ratio of BGA by analyzing each period independently. As $R' = \bigcup_j R'_{(j)}$ and $R^* = \bigcup_j R^*_{(j)}$, it is clear that for any $\alpha \geq 1$, $P_{R^*}/P_{R'} \leq \alpha$ follows if we can show that $P_{R^*_{(j)}}/P_{R'_{(j)}} \leq \alpha$ for all j, $1 \leq j \leq \gamma'$.

According to Lemma 19, when $l_{j+1} - l_j = 2$, $P_{R^*_{(j)}}/P_{R'_{(j)}} \leq \frac{5}{3}$. According to Lemma 20, when $l_{j+1} - l_j = 3$, $P_{R^*_{(j)}}/P_{R'_{(j)}} \leq \frac{5}{3}$. By the partition rule, if $l_{j+1} - l_j = 1$, then period j is sufficient, i.e., $|R'_{(j)}| \geq \frac{3}{5}|R^*_{(j)}|$, and hence $P_{R^*_{(j)}}/P_{R'_{(j)}} \leq \frac{5}{3}$. Since $P_{R^*_{(j)}} = P_{R'_{(j)}} = 0$ when $j = 1$ and $j = \gamma'$ (recall that by Observation 3, all requests accepted by BGA and *OPT* do not start earlier than $t_{r'_1}$ and do not start later than $t_{r'_{|R'|}}$), the theorem follows. ◀

4 Conclusion

We have studied an on-line problem with k servers and two locations that is motivated by applications such as car sharing and taxi dispatching. In particular, we have analyzed the effects that different constraints on the booking time of requests have on the competitive ratio that can be achieved. For all variants of booking time constraints we have given matching lower and upper bounds on the competitive ratio. The upper bounds are all achieved by the same balanced greedy algorithm (BGA) with different choices for the number of specified servers ($2\theta k$). Interestingly, $k = 3$ servers suffice to achieve competitive ratio 1.5 (in the case of kS2L-F with $a \geq t$ and kS2L-V with $b_l \geq t$ and $b_u - b_l < t$), and $k = 5$ servers suffice to achieve competitive ratio $\frac{5}{3}$ (in the case of kS2L-V with $b_l \geq t$ and $b_u - b_l \geq t$), and a larger number of servers does not lead to better competitive ratios.

In future work, it would be interesting to determine how the number of servers, the number of locations, and the constraints on the booking time affect the competitive ratio for the general car-sharing problem with k servers and m locations.

References

1 Norbert Ascheuer, Sven Oliver Krumke, and Jörg Rambau. Online Dial-a-Ride Problems: Minimizing the Completion Time. In *Proc. of the 17th Annual Symposium on Theoretical Aspects of Computer Science (STACS '00)*, volume 1770 of *LNCS*, pages 639–650. Springer, 2000. doi:10.1007/3-540-46541-3_53.

2 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 *Proc. of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '17)*, pages 994–1005. SIAM, 2017. doi:10.1137/1.9781611974782.63.

3 Katerina Böhmová, Yann Disser, Matús Mihalák, and Rastislav Srámek. Scheduling Transfers of Resources over Time: Towards Car-Sharing with Flexible Drop-Offs. In *Proc. of the 12th Latin American Symposium on Theoretical Informatics (LATIN'16)*, volume 9644 of *LNCS*, pages 220–234. Springer, 2016. doi:10.1007/978-3-662-49529-2_17.

4 Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis.* Cambridge University Press, 1998.

5 Ananya Christman, William Forcier, and Aayam Poudel. From theory to practice: maximizing revenues for on-line dial-a-ride. *J. Comb. Optim.*, 35(2):512–529, 2018. doi:10.1007/s10878-017-0188-z.

6 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 *Proc. of the 3rd International Workshop on Approximation and Online Algorithms (WAOA 2005), Revised Papers*, volume 3879 of *LNCS*, pages 258–269. Springer, 2006. doi:10.1007/11671411_20.

7 Kelin Luo, Thomas Erlebach, and Yinfeng Xu. Car-Sharing between Two Locations: Online Scheduling with Flexible Advance Bookings. In *Proc. of the 24th International Computing and Combinatorics Conference (COCOON '18)*, 2018. To appear.

8 Kelin Luo, Thomas Erlebach, and Yinfeng Xu. Car-Sharing between Two Locations: Online Scheduling with two Servers. In *Proc. of 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS '18)*, 117. LIPIcs, 2018. To appear. URL: `http://www.cs.le.ac.uk/~te17/papers/mfcs2018.pdf`.

9 Fanglei Yi and Lei Tian. On the Online Dial-A-Ride Problem with Time-Windows. In *Proc. of the 1st International Conference on Algorithmic Applications in Management (AAIM '05)*, volume 3521 of *LNCS*, pages 85–94. Springer, 2005. `doi:10.1007/11496199_11`.